UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

AISLAN GOMIDE FOINA

SISTEMA DE SELEÇÃO AUTOMÁTICA DE CONTEÚDO TELEVISIVO
ESCALÁVEL BASEADO EM REDE DE SENSORES

São Paulo
2012

AISLAN GOMIDE FOINA

SISTEMA DE SELEÇÃO AUTOMÁTICA DE CONTEÚDO TELEVISIVO
ESCALÁVEL BASEADO EM REDE DE SENSORES

Tese apresentada à Escola Politécnica
da Universidade de São Paulo para
obtenção do título de Doutor em
Ciências

São Paulo
2012

AISLAN GOMIDE FOINA

SISTEMA DE SELEÇÃO AUTOMÁTICA DE CONTEÚDO TELEVISIVO
ESCALÁVEL BASEADO EM REDE DE SENSORES

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Ciências

Área de concentração:

Engenharia Elétrica Sistemas Eletrônicos

Orientador:

Prof. Francisco Javier Ramirez-Fernandez

São Paulo
2012

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 30 de janeiro de 2012.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Foina, Aislan Gomide

**Sistema de seleção automática de conteúdo televisivo escalável baseado em rede de sensores / A.G. Foina. -- ed.rev. – São Paulo, 2012.
125 p.**

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1. Mineração de dados 2. Sistemas distribuídos 3. Arquitetura e organização de computadores 4. Tecnologia bluetooth 5. Computação pervasiva I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II. t.

A meus pais, Paulo e Luciana, que sempre foram meus professores do dia a dia; a meu irmão, Ariel, cujas divergências sempre nos rendiam horas de discussão; e a meu avô, Rubens, o culpado pela minha paixão pela Engenharia.

AGRADECIMENTOS

Após quase cinco longos anos de dias e noites de trabalho imensurável, dificuldades para conseguir equipamentos, tempo para conciliar todas as tarefas pessoais e profissionais e, às vezes, falta de energia para continuar acordado para mais uma hora de trabalho, muitas foram as pessoas que colaboraram com este trabalho direta ou indiretamente. Tantos palpites, comentários, sugestões, críticas e brigas fizeram com que esta tese atingisse o seu formato final.

Antes de tudo, agradeço à natureza, por sua beleza nas coisas simples do dia a dia, com um pôr do sol no final da tarde, um arco-íris depois de uma chuva forte, e todo o ecossistema que tentamos destruir a cada dia, porém que ainda resiste bravamente à nossa intervenção.

Das pessoas, o primeiro que merece agradecimento é meu orientador e amigo, Prof. Dr. Francisco Javier Ramirez-Fernandez, cuja experiência e conhecimento colaboraram de forma única com este trabalho, sem falar das longas e duradouras discussões sobre temas de engenharia e pesquisa. Outro orientador que tive foi a Prof.^a Rosa Maria Badia, do Barcelona Supercomputing Center (BSC), que por sua fé na minha palavra, aceitou me orientar no programa sanduíche em Barcelona sem nunca ter me conhecido pessoalmente e nem conversado por telefone. A oportunidade que me foi dada mudou completamente a minha vida.

Aos meus pais, que também tiveram uma colaboração de extrema importância pelo apoio emocional, conselhos e sugestões. Agradeço pela revisão técnica realizada pelo meu pai, Paulo Rogério Foina, e pela revisão gramatical feita pela minha mãe, Luciana de Mello Gomide.

Ao meu irmão Ariel, que sempre me rendeu longas conversas sobre assuntos aleatórios; à minha avó Anita que me sempre me deu abrigo e carinho quando precisei; ao meu falecido avô Rubens, por todos os ensinamentos sobre engenharia que ele me passou; à minha falecida avó Wanda, que descansa em paz, e o meu ainda vivo avô Herval que não via há mais de 10 anos e o reví poucos meses antes de terminar esta tese. Enfim, a toda família, que apesar do tempo e da distância que nos separa, sempre serão pessoas importantes na minha vida.

À minha amada Ana, pela paciência e compreensão, e às vezes até incompreensão, por não estar presente ou disponível para momentos de lazer durante esses anos de pesquisa.

À Jú, amiga e “irmã”, sempre ao meu lado tentando enriquecer o meu vocabulário e me tornar uma pessoa mais culta. Graças a seu conhecimento em inglês, sempre fez ótimas traduções e correções. Ao Mateus, que é meu outro “irmão”, que devido à nossa competitividade com tudo, até na hora de doar sangue, sempre me força a ser o melhor em tudo que faço.

Aos meus grandes amigos de toda vida, especialmente Emille, Henrique, Cris, Alvinho, Rafael Máquina, Mari, Neto, Drudi, Coruja, George, Baixinho, Dri&Dan+Guido, Tiago Januzzi e Marisa, que tiveram uma enorme importância na minha vida, me apoiaram enquanto este texto ainda estava no início e comemoraram comigo a cada etapa pessoal e profissional concluída.

Aos amigos que fiz durante minha estadia em Barcelona, os casais brasileiros Fabio&Dri, Tábata&Thi e Gi&Gere, às também brasileiras Gisele, Ju, Daianna. Aos mexicanos, entre eles Luis Pedro, Aida, Enrique e Paula e Adriana, que me receberam de braços abertos. À minha grande amiga Nati e as *Charlie Angels* Marta, Merche e Sonia com seus respectivos irmãos, que me apresentaram o mundo espanhol e catalão. Aos outros latinos de Barcelona Ricardo, Rócio e Daniel, e minha arqui-inimiga argentina Valentina, parceiros de festa e programas culturais. A todos os outros com quem dividi apartamento e conheci durante todos esses meses, que suportaram as minhas manias.

À FAPESP, CNPq e todas as outras instituições de fomento à pesquisa, pelos seus apoios em minha vida acadêmica, tentando diminuir o grande obstáculo criado pelo Brasil para a pesquisa e desenvolvimento.

À USP, por sua grandiosidade e pela formação profissional e pessoal; e ao LME, inclusive às pessoas que lá trabalham, por terem me acolhido quando eu era um simples estudante de 2º ano e que esteve me apoiando durante toda a minha vida acadêmica.

Ao Barcelona Supercomputing Center que me acolheu por meses em um país onde não conhecia ninguém e que me abriu as portas para a Europa. Os equipamentos que me foram disponibilizados nesta instituição, como o Marenostrum, eram o que tinham de mais moderno na época, e que nunca pensei que um dia teria acesso.

A todas as empresas que colaboraram com equipamentos, informações, apoio e contatos, entre elas a GTP e a Activewave.

“Lembrar que estarei morto em breve é a ferramenta mais importante que já encontrei para me ajudar nas grandes escolhas da vida”

(Steve Jobs)

RESUMO

Com o uso da tecnologia de Identificação por Radiofrequência (RFID), arquiteturas heterogêneas de processadores e as novas tendências da TV Digital e televisão via rede IP (IPTV) foi desenvolvido um sistema para montar, em tempo real, em forma automática, uma programação televisiva personalizada, baseada no perfil do grupo de usuários de um determinado televisor. Aplicações de vídeo sob demanda (VoD), IPTV e TV Digital permitem que cada telespectador possa assistir aos programas a qualquer momento, e assim construir sua grade de programação personalizada. Com um sistema de RFID é possível identificar as pessoas que se encontram próximas ao televisor. Com essas tecnologias unidas a um subsistema de análise de perfil, junto com os dados fornecidos pelos telespectadores no momento da contratação do serviço, e uma interface (*middleware*) para gerenciar os dados, é possível configurar um sistema que escolhe automaticamente quais programas e quais comerciais serão apresentados no aparelho de TV. Essa escolha é baseada no perfil dos telespectadores presentes naquele momento à frente da televisão e dos programas disponíveis naquele instante. As etiquetas (*tags*) de RFID usadas para o levantamento da audiência foram aparelhos celulares equipados com tecnologia *Bluetooth*, que possibilitam a identificação simultânea dos telespectadores via rádio. O algoritmo de recomendação é híbrido, possuindo componentes baseados em conteúdo e componentes colaborativos. O uso dos novos processadores heterogêneos exigiu o desenvolvimento de algoritmos paralelos que utilizam instruções do tipo SIMD, aceleradores e GPUs. Os sistemas que existem no momento (2011) nesta área, se limitam à identificação dos usuários mediante a digitação usando o controle remoto da TV e só identificam uma pessoa de cada vez. O uso de tecnologia por rádio, proposto nesta pesquisa, permite a identificação de várias pessoas simultaneamente, exigindo o desenvolvimento de padrões de um sistema completo baseado em grupos de perfis diferentes. A arquitetura do sistema elaborado está baseada no processador Cell BE e nas arquiteturas CPU+GPU, de forma que o tempo de execução do algoritmo fosse minimizado.

Palavras-chave: Bluetooth. Computação intensiva. Processadores heterogêneos. RFID. Sistemas de recomendação. Televisão.

ABSTRACT

Merging together Radiofrequency identification (RFID), heterogeneous architectures of processors and new tendencies of the Digital TV (DTV) and television through IP network (IPTV), a system to create, automatically and in real-time, a personalized TV program schedule, based on the group of people profile next to a TV. Video-on-Demand (VoD) applications, IPTV and DTV allow each person to watch a chosen program at any moment and to its personalized programming guide. The RFID system allows the identification of the people next to the TV. This technology used with a profile analysis subsystem accessing a database of people preferences, and a middleware to manage the data, it is possible to set a system that automatically chooses with TV shows and with TV ads will be presented in the TV. This selection is based on the profile of the people next to the TV in that instant and on the available programs. The RFID tags used to detect the audience were the mobile phones equipped with Bluetooth, which allows the identification of its owner wirelessly. The recommendation algorithm is hybrid, containing collaborative and content-based components. The new heterogeneous processors demanded the development of parallel algorithms that use SIMD instruction, accelerators and GPUs. The systems that were available in the moment of this research (2011) were limited to the identification through login using remote control, one person by time. The use of RFID technology, proposed in this research, enables the simultaneous identification of many people at a time, demanding the development standards for group profiles recommendation. The systems architectures will be based on Cell BE processor and the conjunct CPU+GPU, focusing in the reduction of the algorithm execution time.

Keywords: Bluetooth. Intensive Computing. Heterogeneous Processors. RFID. Recommendation systems. Television.

LISTA DE FIGURAS

Figura 1 - Sistema de RFID	25
Figura 2 - Piconet e Scatternet	27
Figura 3 - Arquitetura Ginga.....	31
Figura 4 – As diversas APIs do Ginga-J	32
Figura 5 - Comunicação dos serviços do 2S ACT.....	51
Figura 6 - Diagrama dos módulos do 2S ACT embarcados em um STB.....	52
Figura 7 – Gráfico do perfil de gêneros do telespectador 885013.	58
Figura 8 - Diagrama de fluxo do STBM.....	65
Figura 9 - Logs de entrada e saída dos dispositivos da área de cobertura	66
Figura 10 - Gráfico dos testes de demora na detecção dos celulares.....	67
Figura 11 - Gráfico dos alcances obtidos	67
Figura 12 - Diagrama de fluxo da implementação do G-means. CV é o valor crítico calculado usando o nível de confiabilidade α , k é o número de clusters, C é o conjunto de centros dos clusters, X é o conjunto de dados e X_j é o conjunto de pontos que pertencem ao centro c_j	71
Figura 13 - Gráfico comparando quatro diferentes execuções do G-Means. A linha de baixo (quadrado) representa o ganho do código contendo somente as melhoras do k-Means. A segunda linha (triângulo) apresenta o ganho do G-Means com as melhoras na função de projeção, enquanto a seguinte linha (círculo) mostra o ganho do código contendo o a melhora do cálculo do Anderson&Darling. A última linha (estrela) representa a execução do G-Means com o código final após a afinação dos parâmetros de execução e de tamanho de <i>chunk</i>	80
Figura 14 - Gráfico comparativo entre diversos ambientes computacionais de um processador, mostrando o ganho do algoritmo usando como referência o tempo de execução em um Intel P4.....	81
Figura 15 - Gráfico comparativo entre diversos ambientes computacionais multi-processado mostrando o ganho do algoritmo usando como referência o tempo de execução em um Intel P4 da figura anterior.	82
Figura 16 - Representação gráfica de um arquivo de depuração da aplicação GAPGEOFEM com os 13 clusters encontrados representado por símbolos e cores diferentes.	83
Figura 17 – Comparação na mesma escala de tempo da execução do G-Means usando os 2 CPUs e usando dois CPUs com dois GPUs. As duas primeiras linhas representam o uso do CPU e as duas últimas representam o uso do GPU. As cores, com exceção do azul claro, representam as diferentes tarefas em execução. O azul claro marca o estado de ociosidade do processador.	85
Figura 18 - (Acima) <i>Timeline</i> da execução usando GPUs; (Abaixo) Zoom nas tarefas, sendo as vermelhas tarefas de CPU; as rosas são as tarefas do GPU; e as transferências entre a memória RAM e a memória dos GPUs são ilustradas nas cores azul e branca.....	85
Figura 19. Gráfico mostrado a variação do número de clusters encontrados em relação ao valor crítico.	87
Figura 20 – Perfil médio gerado pela média dos perfis existentes na base de dados.....	88
Figura 21 – Cluster número um onde o gênero forte é terror, seguido de suspense e ficção científica, e os gêneros fracos são comédia, drama e romance.	88
Figura 22 – Cluster número seis, com drama e romance sendo gêneros fortes, e ação, terror e ficção científica com gêneros fracos.	89
Figura 23 – Cluster número oito, mostrando uma situação provável de família com filho pequeno, onde os gêneros fortes são família, animação e fantasia. Os gêneros fracos são drama, comédia, ação, suspense e romance.	89
Figura 24- Diagrama de funcionamento do PMS junto com o ReS.....	92
Figura 25 - Gráfico de previsões por segundo pelo número de processadores.	96

LISTA DE TABELAS

Tabela 1 - Gêneros do IMDb	35
Tabela 2 – Tabela de Gêneros do telespectador de ID 885013 da base da Netflix.....	58
Tabela 3 – Tempo de execução do G-Means para cada parte do algoritmo.....	77
Tabela 4 – Tempo de execução do G-Means para tamanhos de <i>chunk</i> diferentes usando o conjunto de dados 600k60d.....	78
Tabela 5 – Tempo de execução do G-Means para cada parte do algoritmo melhorado.	79

LISTA DE ALGORITMOS

Algoritmo 1	G-Means(X, α)	69
Algoritmo 2	Anderson-Darling(X_j, α)	70
Algoritmo 3	Parte do código da tarefa mostrando a implementação do algoritmo de cálculo de distância usando instruções SIMD.....	73
Algoritmo 4	Código da tarefa de cálculo da projeção do SPU.....	74
Algoritmo 5	Tarefas de cálculo do teste de Anderson&Darling	75

LISTA DE SIGLAS

2S ACT	Sistema de Seleção Automática de Conteúdo Televisivo
A&D	Anderson & Darling
API	<i>(Application Programming Interface)</i> Interface de programação de aplicativos
ATSC	<i>(Advanced Television System Committee)</i> Comitê de sistema televisivo avançado
BML	<i>(Broadcast Markup Language)</i> Linguagem de marcação para radiodifusão
BtID	<i>(Bluetooth Identification)</i> Identificação via Bluetooth
BtTag	<i>(Bluetooth Tag)</i> Etiqueta de Identificação via Bluetooth
Cell BE	<i>(Cell Broadband Engine)</i> Motor de banda larga Cell
CellSs	Cell Superscalar
CDC	<i>(Connected Device Configuration)</i> Configuração de dispositivos conectados
CLDC	<i>(Connected Limited Device Configuration)</i> Configuração limitada de dispositivos conectados
CPU	<i>(Central Processing Unit)</i> Unidade central de processamento
CSS	<i>(Cascading Style Sheets)</i> Folha de estilo em cascata
CUDA	<i>(Compute Unified Device Architecture)</i> Arquitetura de dispositivos de computação unificada
DMA	<i>(Direct Memory Access)</i> Acesso direto a memória
DTV	<i>(Digital TV)</i> TV digital
DVB	<i>(Digital Video Broadcast)</i> Radiodifusão de vídeo digital
DVD	<i>(Digital Video Disk)</i> Vídeo disco digital
EPC	<i>(Electronic Product Code)</i> Código eletrônico do produto
EPG	<i>(Electronic Product Guide)</i> Guia eletrônico de produto
GCF	<i>(Generic Connection Framework)</i> Plataforma genérica de conexão
GEM	<i>(Globally Executable MHP)</i> MHP de execução global
GPGPU	<i>(General Purpose Graphic Processing Unit)</i> Unidade de processamento

gráfico de uso geral

GPU	<i>(Graphical Processing Unit)</i> Unidade de processamento gráfico
GPUSs	GPU Superscalar
HD	<i>(High Definition)</i> Alta Definição
HF	<i>(High Frequency)</i> Alta Frequência
HTML	<i>(HyperText Markup Language)</i> Linguagem de marcação de hipertexto
ID	Identificação
IMDb	<i>(Internet Movie Database)</i> Base de dados de filmes da Internet
IPTV	<i>(IP Television)</i> Televisão via rede IP
ISDB	<i>(Integrated Services Digital Broadcasting)</i> Serviço integrado de transmissão digital
ISDB-T	<i>(Integrated Services Digital Broadcasting Terrestrial)</i> Serviço integrado de transmissão digital terrestre
ISM	<i>(Industrial, Scientific, Medical)</i> Industrial, científica e médica
iTV	TV interativa
LED	<i>(Light-emitting Diode)</i> Diodo emissor de luz
LF	<i>(Low Frequency)</i> Baixa frequência
MHP	<i>(Multimedia Home Platform)</i> Plataforma multimídia residencial
NestedSs	Nested Superscalar
NFC	<i>(Near-Field Communication)</i> Comunicação de campo próximo
OBEX	<i>(Object Exchange)</i> Troca de objetos
OCAP	<i>(OpenCable Application Platform)</i> Plataforma de aplicação da OpenCable
OPP	<i>(Object Push Profile)</i> Perfil de envio de objetos
PC	<i>(Personal Computer)</i> Computador pessoal
PDA	<i>(Personal Digital Assistants)</i> Assistente pessoal digital
PMS	<i>(Profile Manager Subsystem)</i> Subsistema de gerência de perfis
PPU	<i>(Power Processor Unit)</i> Unidade de processamento Power
PTV	<i>(Personalized Television Listings)</i> Listagem de televisão personalizada
RC	<i>(Remote Control)</i> Controle Remoto
ReS	<i>(Recommender Subsystem)</i> Subsistema de recomendação

RF	Radio Frequência
RFID	<i>(Radio Frequency Identification)</i> Identificação por radiofrequência.
RMSE	<i>(Root Mean-Square Error)</i> Raiz do erro médio quadrático
RSSI	<i>(Received Signal Strength Indication)</i> Indicação de intensidade de sinal recebido
RTLS	<i>(Real Time Location System)</i> Sistema de localização em tempo real
SBTVD	Sistema brasileiro de TV Digital
SIMD	<i>(Single Instruction Multiple Data)</i> Instrução única múltiplos dados
SISD	<i>(Single Instruction Single Data)</i> Instrução única dado único
SMP	<i>(Symmetric Multiprocessing)</i> Multiprocessamento simétrico
SMPs	SMP Superscalar
SPUs	<i>(Synergistic Processor Unit)</i> Unidade de processamento sinérgico
StarSs	Star Superscalar
STB	<i>(Set-top Box)</i> Conversor de TV
STBM	<i>(Set-top Box Middleware)</i> Middleware do conversor de TV
TDOA	<i>(Time Difference of Arrival)</i> Diferença de tempo de chegada
TV	Televisão
UBA	<i>(User Behavior Analyzes subsystem)</i> Subsistema de análise comportamental do usuário
UHF	<i>(Ultrahigh Frequency)</i> Ultra alta frequência
vCard	Cartão de visita eletrônico
VM	<i>(Virtual Machine)</i> Máquina Virtual
VoD	<i>(Video on-demand)</i> Vídeo sob demanda
WUSB	<i>(Wireless Universal Serial Bus)</i> USB sem fio
XML	<i>(Extensible Markup Language)</i> Linguagem de marcação extensível

SUMÁRIO

1. INTRODUÇÃO	17
1.1. 2S ACT.....	18
1.2. Objetivos	20
1.3. Justificativa.....	21
2. FUNDAMENTOS	24
2.1. RFID.....	24
2.2. Bluetooth	26
2.3. Bluetooth como Tecnologia RFID	28
2.4. Redes Ad Hoc	29
2.5. Padrões e Especificações para TV Digital e IPTV	29
2.6. Sistemas de Recomendação	32
2.7. Processadores Multi-core Heterogêneos.....	35
2.8. Star Superscalar.....	38
2.9. Conclusão	41
3. REVISÃO BIBLIOGRÁFICA	42
3.1. Bluetooth	42
3.2. Sistemas de Recomendação para TV	43
3.3. Sistemas de Recomendação com Recurso de Identificação	45
3.4. Sistemas de Recomendação para Múltiplos Usuários	46
3.5. Sistemas de Recomendação Escaláveis	48
3.6. Conclusão	49
4. METODOLOGIA	51
4.1. Arquitetura.....	51
4.2. STBM.....	52
4.3. PMS	54
4.4. UBA.....	57
4.5. ReS	59
4.6. Netflix, Movielens, IMDb e Allmovie.....	60
4.7. Forma de Análise dos Resultados	61
4.8. Conclusão	63
5. IMPLEMENTAÇÃO	64
5.1. STBM	64
5.2. UBA.....	68
5.3. PMS e ReS.....	90
5.4. Conclusão	97
6. CONCLUSÃO E PERSPECTIVAS DE TRABALHOS FUTUROS.....	98
REFERÊNCIAS	103
APÊNDICE A - SCRIPT DE CORRELAÇÃO ENTRE A BASE IMDB E O NETFLIX.....	114
APÊNDICE B - CÁLCULO DO RMSE USANDO O PROCEDIMENTO DO BELLKOR ..	115
ANEXO A - SCRIPT DE CONVERSÃO DE MYSQL PARA SQLITE	118

1. INTRODUÇÃO

Desde o princípio da sociedade moderna, a humanidade vem substituindo o legado de suas tecnologias por novas descobertas feitas por meio de pesquisa e desenvolvimento. Antigos motores a vapor foram trocados por motores a combustão interna, que por sua vez estão sendo trocados por motores elétricos. A tocha pré-histórica se tornou uma lâmpada de óleo, que agora perdeu espaço para as lanternas elétricas e iluminação a LED. Os antigos telégrafos foram trocados pelos sistemas de telefonia fixa, cujo uso está sendo reduzido por causa dos sistemas de telefonia de voz sobre rede IP (VoIP) e pelos telefones celulares. Este mesmo fenômeno que acontece com todas as tecnologias existentes também acontece com a televisão.

O modelo atual de TV foi desenvolvido nas décadas de 20 (ABRAMSON, 1995; STASHOWER, 2002), e desde então ele mantém a mesma estrutura, com pequenas alterações nos tipos de tela e na sua qualidade. No final dos anos 90, algumas transmissões televisivas se tornaram digitais, e alguns países começaram suas transmissões digitais em alta definição (HD), iniciando assim a primeira troca dos televisores para TVs digitais (DTV) (WALLACE, 1997). Na mesma década de 90, mais precisamente em 1995, a expressão IPTV foi mencionada pela 1ª vez (SATTLER, 1995; DISZ; OLSON; STEVENS, 1997). Ao final da década, algumas empresas começaram a transmitir via *web* programas de TV ao vivo, e após o ano 2000 outras empresas começaram a explorar o sistema de IPTV e de vídeo sob demanda (VOD) junto com telefonia fixa e internet banda-larga, criando o conceito do *triple-play* (internet, telefonia e televisão). Nestes casos, o conteúdo televisivo é transmitido ao usuário final utilizando a infraestrutura de telefonia fixa ou pelo sistema de TV a cabo (CHANG; COGGINS; PITT, 1994).

Outra tecnologia que não mudou significativamente seu uso no contexto televisivo foi o controle remoto (RC). Este dispositivo foi usado na década de 50 para controlar uma televisão usando um longo fio ligado o controle remoto e a TV. Alguns anos depois, o primeiro controle remoto sem fio para controlar televisores foi inventado usando uma luz piscante para enviar comandos de alteração de volume e de canais (JAGER, 1959; BUCHSBAUM, 1957). Desde aquela época, a única alteração significativa no seu funcionamento foi a invenção do controle remoto universal com recursos de aprendizado.

Essa nova funcionalidade permite um só controle substituir pelo menos quatro ou mais controles usados para comandar um tocador de DVD, um televisor, um receptor de TV a cabo, ou *Set-top Box* (STB), um *home-theater*, entre outros.

Desde a invenção da televisão e do controle remoto, outras tecnologias com potencial de inovar o mercado televisivo foram desenvolvidas. Algumas delas são o telefone celular, os processadores embarcados, as telas de alta-definição e recentemente as televisões 3D. Essas tecnologias, e como elas podem inovar a forma de assistir TV, foram analisadas de forma a melhorar a experiência do telespectador e gerar novas fontes de receita para as empresas de televisão. O resultado dessa pesquisa deu origem ao Sistema de Seleção Automática de Conteúdo Televisivo, ou simplesmente 2S ACT apresentado nesta Tese.

1.1. 2S ACT

As operadoras de TV a cabo estão disponibilizando cada vez mais novos canais na sua grade de programação. Hoje, esse número já ultrapassa os 100 canais oferecidos (NET, 2007). A navegação desses canais de forma a encontrar um programa de interesse é muito demorada, pois o telespectador tem que assistir cada canal para escolher qual programa o agrada mais. Recursos como grade de programação apresentada na tela e sinopse dos programas são disponibilizados para facilitar a escolha. Porém, mesmo com esses recursos, essa escolha ainda é um processo demorado. Se o aparelho de TV pudesse saber em tempo real o perfil das pessoas que o estão assistindo, o mesmo poderia recomendar os programas que se enquadram melhor no gosto do telespectador no início da escolha de canais.

Usando uma tecnologia de Identificação por Radiofrequência (RFID), os serviços de escolha da programação e da transmissão dos comerciais podem ser melhorados. Essa tecnologia permite a identificação de pessoas e objetos a alguns metros de distância por meio de ondas de rádio. Com a incorporação dessa tecnologia no receptor de sinal de TV, chamado de *Set-Top Box* (STB), e o uso de transponders (denominados também de *tags*) pelas pessoas que assistem TV, o STB saberá quem está próximo ao televisor, podendo escolher canais automaticamente. Um celular com Bluetooth pode ser usado como um *tag* de RFID para identificar unicamente seus donos. Assim, aproveitando a capacidade computacional do celular, além de identificar os telespectadores, o aparelho pode ser usado como controle remoto, permitindo a todos telespectadores o controle sobre o televisor e ao

mesmo tempo permitindo ao STB saber quem escolheu cada canal. O mesmo telefone celular também pode ser usado como interface dos programas interativos, onde cada telespectador pode interagir da forma que desejar com o programa de televisão.

No Brasil existem aproximadamente 100 milhões de pessoas que assistem TV, sendo que 9,8 milhões são assinantes de televisão por assinatura (ABTA, 2011). Todos esses telespectadores assistem às mesmas programações, ainda que haja escolha de conteúdo entre os canais de TV aberta e fechada. Os anúncios comerciais são inseridos nos intervalos de acordo com os dados de público-alvo e audiência disponibilizados pelas emissoras e institutos de pesquisa, sendo que não há garantia sobre quem de fato está sendo exposto à propaganda.

Com a chegada da TV digital (DTV), Televisão IP (IPTV) e Vídeo sob demanda (VoD), surge a possibilidade de o telespectador montar manualmente sua própria programação, sem ficar preso ao horário das atuais grades de programas. Esse serviço seria similar ao serviço de escolha de filmes por *pay-per-view*, oferecido pelas operadoras de TV a cabo, que o usuário pode assistir a um dado filme a hora que ele quiser. Além de escolher o horário de início do filme ou programa, o telespectador tem recursos de avançar, pausar e voltar, como se estivesse assistindo a um DVD. Essas novas abordagens de transmissão televisiva também possibilitam a interação nos programas de TV, pelo fato da existência de um canal de retorno, permitindo uma comunicação bidirecional entre o canal de TV e o televisor. Com esse canal de dados estabelecido é possível a compra de produtos, filmes e programas pelo próprio controle remoto do aparelho.

Analisando os canais que um telespectador assiste, e gravando as informações de quais programas e filmes foram assistidos em uma base de dados, pode-se definir um perfil de gosto da pessoa. Sabendo o perfil de cada uma das pessoas que estão assistindo a TV, o STB ou a empresa provedora de conteúdo televisivo pode escolher automaticamente apenas os programas de interesse do grupo. Quanto aos comerciais, somente seriam inseridos aqueles que têm o perfil dos telespectadores como parte do público alvo. Então, juntando a tecnologia de RFID para saber quem está assistindo a TV no momento, um sistema de IPTV e VoD para recepção de programação personalizada, uma base de dados com informações dos hábitos de consumo de cada pessoa, e finalmente um sistema de análise de perfil para consultar essa base de dados, foi construído o 2S ACT. Este sistema monta uma grade dirigida em tempo real, que contém recomendações de programas e comerciais. A seguir,

envia essa grade para as pessoas que estiverem assistindo TV naquele momento e que possuam um aparelho celular com recurso de Bluetooth ativado, que serve como ferramenta para detecção das pessoas presentes.

O problema é que algoritmos de análise de perfil utilizam técnicas de mineração de dados, e essas técnicas fazem parte do grupo dos algoritmos de computação intensiva. Executar um algoritmo de análise de perfil de uma pessoa já necessita uma quantidade grande de cálculos estatísticos, então a análise de um grupo de pessoas multiplica esse tempo de computação pelo número de pessoas do grupo. Desta forma, os algoritmos precisam ser otimizados para utilizar todos os recursos disponíveis do processador para realizar esses cálculos no menor tempo possível. Algumas arquiteturas de processadores podem ser utilizadas para acelerar a execução desses algoritmos, como os processadores heterogêneos, gerando os resultados em um tempo aceitável.

1.2. Objetivos

O objetivo principal desta pesquisa é estudar meios que viabilizem um sistema escalável de seleção de programas de TV baseada no perfil dos telespectadores de maneira automática e em tempo real, sem a intervenção humana. Para isso, esse projeto tem alguns objetivos secundários.

Um desses objetivos é o uso de arquiteturas heterogêneas *multi-core* e/ou que suportem instruções de cálculo vetorial (SIMD) para executar os algoritmos de recomendação, reduzindo o tempo de processamento para grandes quantidades de dado. Neste caso, mais especificamente, foram escolhidos ambientes heterogêneos que possuam tanto aceleradores SIMD, placas gráficas ou o processador *Cell Broadband Engine* (Cell BE), como ambientes que possuem vários núcleos simétricos (SMP).

Outro objetivo secundário é a integração de hardware e software de um leitor de RFID com um STB de TV. Essa integração é fundamental para o sistema de seleção automática, pois a parte de identificação dos telespectadores é feita pelo leitor de RFID. A tecnologia de RFID usará padrão Bluetooth de comunicação, e um aparelho celular como *tag*.

E o último dos objetivos secundários seria o estudo e implementação de métodos de classificação do perfil de grupos de telespectadores baseado, por exemplo, nos canais assistidos, no tempo gasto em cada um, gênero, atores e diretores dos programas assistidos,

se o programa foi assistido até o final, entre outros dados a serem analisados para melhor definir um perfil. Esses dados serão cruzados com o histórico de grupos formados pelos telespectadores entre si, a fim de definir um perfil do grupo de telespectadores que está assistindo ao televisor naquele instante. Essa classificação servirá como fonte de informação para o *middleware* escolher os programas que serão enviados ao STB.

1.3. Justificativa

As principais justificativas desta pesquisa são as vantagens que seus resultados podem proporcionar frente à tecnologia existente de IPTV e TV digital, seja para pesquisadores, telespectadores ou empresas relacionadas com o mercado de televisão, e do grande leque de novas aplicações e linhas de pesquisa que podem fazer uso deste tipo de sistema.

Os primeiros interessados neste tipo de sistema são pessoas que possuem algum tipo de deficiência ou incapacidade, e linhas de pesquisa nessa área. LEE et al. (2007) apresentam uma estrutura de casa inteligente para idosos com uso de tecnologia para facilitar a rotina diária de seus moradores. Seu foco é em interfaces homem-máquina mais inteligentes que facilitam o controle sobre os utilitários domésticos. A união deste sistema com uma interface direta cerebral poderá permitir deficientes físicos a terem controle total sobre seu televisor (JACKSON, 2006). Crianças com deficiência mental também podem utilizar este sistema para ajudar no aprendizado e servir como parte do seu tratamento (MEDINA; CASTELLANI, 2003).

Além da linha de pesquisa na área de saúde, os resultados obtidos aqui podem ser muito úteis na linha de pesquisa de computação pervasiva. O iDorm, por exemplo, é um projeto de quarto inteligente que utiliza diversos equipamentos e sensores ligados em rede para automatizar hábitos diários dos moradores (DUMAN; HAGRAS; CALLAGHAN, 2007).

Outros beneficiados pelos resultados desta pesquisa são os próprios telespectadores, que não precisarão despender muito tempo para escolher o programa que lhes interessa. Simplesmente quando o televisor for ligado, será escolhido automaticamente o programa ou filme que tem mais chance de agradar à preferência de cada usuário. Caso exista mais de uma opção, uma delas é escolhida e as outras serão apresentadas na tela e no controle remoto. Da mesma forma que se durante um programa começar a ser transmitido outro programa de igual interesse em outro canal, o telespectador pode ser avisado por uma pequena mensagem no visor da TV, ou no celular caso tenha mais de uma pessoa assistindo.

Outro benefício para os telespectadores vem do fato de serem usados telefones celulares com Bluetooth para identificação. Por causa disso, as funcionalidades de controle remoto podem ser colocadas no celular, como em INOVAÇÃO TECNOLÓGICA (2011) que aproveita as características do *middleware* do sistema de TV digital brasileiro para conectar um *smartphone* ao STB. Outra vantagem de se utilizar um dispositivo inteligente como controle remoto é poder ao mesmo tempo mostrar as mensagens do sistema de recomendação de canal. O aparelho pode servir também de interface para os programas interativos, apresentando placar de jogos, opções de votação de enquetes e até para servir como interface de compra de produtos que forem apresentados na TV. Todas as funcionalidades do celular podem ser herdadas, permitindo, por exemplo, ao telespectador receber um alarme via SMS em seu aparelho quando um dado programa começar, ativar o alarme de vibratório a cada programa recomendado enviado pelo sistema, ou até criar um contato na agenda do celular com o número telefônico de um anunciante de um comercial que o agradou.

Quanto aos horários de intervalo, estes também têm vantagens em relação ao sistema convencional. Os telespectadores podem ser submetidos somente às propagandas que se enquadrem no seu perfil. Assim, as pessoas se interessarão mais pelos comerciais, gerando uma possível redução da dispersão de audiência durante os intervalos.

Em relação às empresas do mercado televisivo, os resultados obtidos aqui podem gerar diversos diferenciais quando comparado com os sistemas convencionais. Entre essas empresas estão as anunciantes, as geradoras de conteúdo, as transmissoras de TV, e até mesmo empresas de marketing e pesquisa de mercado. Um dos diferenciais propostos permite à empresa de televisão saber o tipo de público que assiste a cada programa em cada horário. Hoje, para o levantamento dessa informação é necessário contratar um instituto de pesquisa para realizar um estudo sobre a audiência, como o IBOPE por exemplo. Outra vantagem é que os canais de televisão podem vender o mesmo horário de comercial para várias empresas anunciantes com diferentes públicos-alvo. Duas ou mais empresas podem comprar o mesmo horário comercial e suas propagandas serem transmitidas somente para as TVs que estiverem sendo assistidas pelo seu público-alvo. Desta forma, diferentes pacotes de horários podem ser oferecidos para as empresas anunciantes a um custo menor. Do outro lado, a empresa de televisão terá um aumento na receita, pois estará vendendo

diversas vezes o mesmo horário. Com a redução de custo do horário comercial, mais empresas poderão anunciar nos canais de televisão.

Programas de vendas também podem se beneficiar desses resultados, apresentando produtos de interesse das pessoas que estão assistindo ao programa. A compra de produtos pode ser feita sem precisar inserir dados de cartão de crédito e informações pessoais, pois esses dados pessoais já estão na base de dados de perfil do telespectador ou armazenadas no próprio telefone celular. Diferentes produtos podem ser ofertados ao mesmo tempo para diferentes pessoas assistindo à mesma TV pelo aparelho celular. Por exemplo, ao ver uma cena de ação de um filme interativo, no aparelho de um pai de família pode estar sendo oferecido um produto *A* da cena, como a roupa de um ator, e no da sua esposa pode estar sendo oferecido um produto *B*, como o vestido da atriz da cena, e ambas as pessoas podem comprar ao mesmo tempo, cada uma com suas informações pessoais de compra.

O texto está dividido em seis capítulos. Exposta esta introdução, o capítulo seguinte, de número dois, apresenta as tecnologias estudadas e utilizadas para desenvolver a pesquisa. A seguir, o Capítulo 3 apresenta a contextualização do 2S ACT em relação ao que já foi e ao que está sendo pesquisando e publicado pela academia. O Capítulo 4 apresenta a metodologia utilizada e a forma de medir os resultados. Já no Capítulo 5, o desenvolvimento e o resultado de cada módulo do sistema são apresentados. Finalmente, no Capítulo 6 está escrita a conclusão, seguida pela bibliografia. Complementando o texto existem dois apêndices e um anexo que descrevem algumas das rotinas e códigos fonte utilizados nesta pesquisa.

2. FUNDAMENTOS

Em muitos cenários a tecnologia de identificação por radiofrequência (RFID) está presente de forma pervasiva. Alarmes de carros e residenciais, onde um controle remoto se identifica e desativa o sistema, e cartões de pagamento automático de pedágio sem a necessidade de estacionar o carro são alguns dos casos de uso da tecnologia de RFID. Cartões de controle de acesso, bilhetes de transporte público e passaportes inteligentes também são exemplos de aplicação desta tecnologia. Porém, todas essas aplicações de RFID exigem que o usuário porte um *tag* de RFID, sendo ele um controle remoto de carro ou um cartão de acesso.

Apesar de muitas pessoas carregarem consigo mais de um tipo de *tag* de RFID, seria muito desejável que os diversos *tags* fossem substituídos por somente um único exemplar que estivesse sempre junto do seu dono. Um equipamento que pode substituir todos os *tags* é o telefone celular, através de tecnologias como *Near-Field Communication* (NFC) e Bluetooth.

Assim, neste capítulo são apresentados conceitos de RFID, Bluetooth e seu uso na identificação de pessoas. Outras tecnologias envolvidas nesta pesquisa também são descritas, como os processadores heterogêneos Cell BE e Tegra, e a ferramenta de desenvolvimento StarSs, que faz a divisão da execução do sistema em tarefas para ser executada entre os seus diversos núcleos disponíveis. Outras tecnologias apresentadas são os padrões de TV digital e seus recursos, assim como os tipos de sistemas de recomendação existentes.

2.1. RFID

Identificação por Radiofrequência, ou RFID, é uma tecnologia sem fio que, tal qual o código de barras, faz parte do grupo de tecnologias de identificação e captura de dados de forma automática. Seu surgimento remonta há várias décadas, mas o crescimento massivo de seu uso vem se percebendo nos últimos anos, em especial pela redução do custo de seus componentes e à padronização de protocolos.

Um sistema RFID convencional é composto por: um transceptor, ou leitor, que gera as ondas de radiofrequência; uma antena que transmite essa onda; e um transponder, mais

conhecido por *tag*, que recebe a onda de RF e responde com algum dado, normalmente um código de identificação. Ao leitor é conectado um sistema computacional que gerencia as informações do sistema RFID e interage com o usuário ou com outras aplicações. Todos os componentes podem ser vistos na Figura 1.

Figura 1 - Sistema de RFID

Existe uma grande diversidade de padrões de frequência, protocolos de comunicação e tipos de *tags* para as mais diversas aplicações. Os *tags* se dividem em três grupos de acordo com a sua eletrônica: os ativos, que possuem bateria interna e a utilizam para alimentar o circuito, enviar e receber mensagens; os passivos, que não possuem bateria e se alimentam das ondas de RF enviadas pelo leitor; e os semi-passivos, que também se alimentam das ondas enviadas pelo leitor, porém possuem uma bateria para amplificar o sinal de resposta transmitido. Cada um desses grupos de *tags* é subdividido por padrão de comunicação utilizado.

As tecnologias RFID passiva e semi-passiva utilizam principalmente frequências entre 125 kHz e 2,4 GHz, e são distinguidos por três faixas: baixa - LF, alta – HF e ultra alta – UHF (WU et al., 2006). O padrão de comunicação mais atual, que está sendo o responsável pela popularização do RFID passivo, é o chamado *Electronic Product Code* (EPC), que pretende utilizar o RFID para substituir o código de barras na cadeia logística. Outros padrões muito comuns na frequência de HF são o ISO 15639 e o ISO 14444 que são usados por cartões de controle de acesso e para comunicação NFC entre celulares.

Sobre os *tags* ativos, existem diversos padrões de comunicação proprietários e alguns padrões que utilizam protocolos conhecidos, como o IEEE 802.11 – Wi-Fi. O RFID ativo possui recursos como medição de distância com precisão por intensidade de sinal (RSSI) ou pelo atraso de tempo entre a transmissão e a recepção (TDOA) que junto com seu longo alcance

de comunicação permitem o seu uso para aplicações de localização em tempo real (RTLS). Outra característica dos *tags* ativos é a sua capacidade de transmitir informação para o leitor sem ter sido requisitado, uma funcionalidade que o passivo não possui por necessitar de um sinal do leitor para se alimentar.

2.2. Bluetooth

Bluetooth é um protocolo padrão de comunicação primariamente projetado para pouco consumo de energia com curto alcance, baseado em *microchips* transmissores de baixo custo em cada dispositivo. O Bluetooth possibilita a comunicação desses dispositivos uns com os outros quando estão dentro do seu raio de alcance. Os dispositivos usam um sistema de comunicação via rádio, por isso não necessitam estar na linha de visão um do outro, e podem estar até em outros ambientes. O único requisito para a comunicação é que a razão sinal/ruído da transmissão recebida seja suficientemente alta para ser interpretada e que uma sequência de autenticação seja feita. As especificações do Bluetooth foram desenvolvidas e licenciadas pelo *Bluetooth Special Interest Group* que conta hoje com mais de 15 mil empresas filiadas.

Um dispositivo mestre Bluetooth pode se comunicar com no máximo outros sete dispositivos. Esse grupo de rede de até oito dispositivos é chamado *Piconet*. Uma *piconet* é uma rede *ad-hoc*, usando protocolos de tecnologia Bluetooth para permitir um dispositivo mestre de se interconectar com até sete dispositivos escravos ativos. Até 255 dispositivos podem estar inativos, "estacionados", com os quais o dispositivo mestre pode se comunicar e alterá-los para o status de ativo quando necessário.

Em qualquer momento, dados podem ser transferidos entre o mestre e outro dispositivo, porém os dispositivos podem trocar de posição e o escravo pode se tornar o mestre a qualquer momento. A condição de mestre troca rapidamente de um dispositivo para outro, e transmissões simultâneas do mestre para vários outros dispositivos são possíveis, porém não muito utilizadas. As especificações do Bluetooth permitem conexão de duas ou mais *piconets* juntas para formar uma *scatternet*, com alguns dispositivos agindo como roteador para simultaneamente trabalhar como o mestre e o escravo em cada uma das *piconets*. Um exemplo dessa topologia de rede pode ser visto na Figura 2.

Figura 2 - Piconet e Scatternet

Os dispositivos Bluetooth operam na faixa *Industrial, Scientific, Medical* (ISM) centrada em 2,45 GHz que era formalmente reservada para alguns grupos de usuários profissionais. Os dispositivos são classificados de acordo com a potência e alcance, em três níveis: classe 1 – 100 metros de alcance; classe 2 – 10 metros; e classe 3 – um metro. A banda é dividida em 79 portadoras espaçadas de 1 MHz, portanto cada dispositivo pode transmitir em 79 frequências diferentes. O dispositivo mestre pode mudar as frequências de transmissão de seus escravos por até 1600 vezes por segundo depois de sincronizado para minimizar as interferências. Teoricamente sua velocidade pode chegar a 721 Kbps e possui três canais de voz.

Um dispositivo Bluetooth, sempre que demandado, irá transmitir alguns pacotes de informações básicas. Essas informações são: nome do dispositivo; classe do dispositivo; lista de serviços disponíveis; e informações técnicas, como por exemplo, características, fabricante, especificação Bluetooth e configuração de *clock*.

Qualquer dispositivo pode realizar uma varredura para encontrar outros dispositivos disponíveis para conexão, e qualquer dispositivo pode ser configurado para responder ou não a essas requisições. Porém, se o dispositivo que estiver tentando conectar já souber previamente o endereço do outro dispositivo, o mesmo vai sempre responder a requisições de conexões diretas, sem a varredura prévia, e transmitir as informações básicas se requisitado. O uso dos dispositivos, porém, requer aceitação do proprietário ou pareamento. Assim, a conexão ficará ativa e aguardando autorização até que seja finalizada ou até que saia do alcance.

Parear dispositivos é o ato de estabelecer uma comunicação segura "aprendendo" uma senha secreta (*passkey*) entre dois dispositivos por meio de entrada de dados do usuário. O

dispositivo que deseja se comunicar com outro deve informar uma senha que também deve ser comum aos dois aparelhos. Assim, depois de emparelhar, os dispositivos lembram os nomes amigáveis dos outros e conectam-se de forma transparente todas as vezes, assim como reconhecemos nossos amigos. Como o endereço Bluetooth é permanente, o pareamento é preservado, mesmo se o nome de algum dos dispositivos for trocado. Os pareamentos armazenados podem ser apagados a qualquer momento, e assim ter as autorizações de conexão removidas.

2.3. Bluetooth como Tecnologia RFID

Vistas as características básicas de um sistema de RFID e algumas informações sobre o protocolo de comunicação Bluetooth, agora será explorado como esta tecnologia pode ser usada como um sistema de RFID.

Cada dispositivo Bluetooth é dotado de um número único de 48 bits que serve de identificação, no formato 00:00:00:00:00:00. Esse número é denominado "Endereço de Bluetooth" (*Bluetooth Address*) e são únicos e exclusivos para cada dispositivo fabricado, assim como o endereço MAC das placas de rede.

Durante uma comunicação via Bluetooth entre dispositivos, esses endereços geralmente não são mostrados para o usuário, e no seu lugar aparecerá o nome amigável do dispositivo, um campo de 82 caracteres que pode ser configurado pelo proprietário. Esse nome é o que aparecerá na lista de dispositivos disponíveis de qualquer aparelho que efetuar uma varredura.

Assim, um dispositivo Bluetooth pode ser configurado para realizar buscas de outro dispositivo Bluetooth periodicamente, fazendo o papel de um leitor de RFID. O dispositivo que faz esse papel será chamado aqui de BtReader. Ao mesmo tempo, outro dispositivo Bluetooth faria o papel do *tag*, transmitido o *Bluetooth Address* e/ou seu nome amigável como código de identificação sempre que for requisitado em uma busca. O dispositivo Bluetooth configurado dessa segunda forma será chamado de BtTag.

Tendo os dois componentes básicos da tecnologia de RFID, o leitor e o *tag*, o Bluetooth pode ser usado para aplicações de RFID onde os alcances necessários atinjam até 100 metros. O sistema de RFID usando tecnologia Bluetooth foi batizado nesta pesquisa de

Bluetooth Identification (BtID), e consiste em BtReaders e BtTags, onde a identificação que o *tag* transmite para o BtReader é o seu *Bluetooth Address* ou o seu nome amigável.

2.4. Redes Ad Hoc

Ad hoc é uma expressão em Latim que significa “para um propósito determinado” e este termo é usado para definir um método de conexão de rede temporário entre dispositivos sem fio. De acordo com a especificação do Bluetooth, uma rede *Ad hoc* é uma rede criada de maneira espontânea, onde não é necessária nenhuma infraestrutura formal, e a sua existência é limitada no tempo e no espaço (BLUETOOTH, 2007). Baseada nessa definição, a conexão entre o telefone celular e o set-top box por exemplo, é considerada uma conexão de rede *Ad hoc*.

Observando a rede criada entre os telefones celulares e o set-top box do ponto de vista da topologia de rede, o STB pode ser usado como um roteador e um servidor. Ele estará conectado na rede de banda larga do IPTV ou ao canal de retorno da TV digital e pode prover conexão transparente entre os celulares e a internet. O STB pode ser usado também como um servidor, executando serviços de rede como *proxy web*, *firewall*, etc, e assim os celulares seriam os clientes desses serviços.

2.5. Padrões e Especificações para TV Digital e IPTV

O STB deve ser compatível com algum padrão de TV digital ou IPTV para permitir a troca de dados entre o STB, os celulares e a rede de TV. Por motivos de legado, o principal padrão aberto de IPTV é baseado em alguns dos padrões de TV digital.

Como muitas tecnologias no mundo, a TV digital possui diversos padrões, e cada país do mundo escolhe qual padrão quer adotar. Existem dois padrões principais, o Europeu e o Americano, e outros dois menos adotados, o Japonês e o Chinês. Cada um deles define de uma maneira sua especificação para a TV interativa (iTV). O consórcio responsável pelo principal padrão europeu chama *Digital Video Broadcast Project* (DVB). O DVB define muitos padrões para o iTV, sendo o principal o *Multimedia Home Platform* (MHP). Este padrão define um *middleware* aberto para suportar o recebimento de aplicações interativas e sua execução pelas diferentes marcas e modelos de STB. Nos EUA, o principal órgão responsável

pelos padrões de TV digital é o CableLabs, e o nome do grupo de padrões é *Advanced Television System Committee* (ATSC). Sua especificação para a TV interativa é chamada *OpenCable Application Platform* (OCAP) e tem a mesma finalidade do MHP. Há alguns anos, o DVB definiu o *Globally Executable MHP* (GEM), uma especificação de alto nível para retirar o requisito da camada de transporte obrigatória do DVB nas definições do MHP, permitindo assim o OCAP e alguns outros padrões proprietários a incorporar as definições do MHP. Desta forma, a última geração dos dois padrões atingiu um nível considerável de compatibilidade. Para o mundo do IPTV, o MHP tem sido usado para definir os protocolos por seu sucesso no mercado de TV digital (DVB, 2010a). O DVB definiu o padrão MHP-IPTV, similar ao MHP para a TV digital; e da mesma forma definiu o padrão GEM-IPTV para permitir uma independência de sistema de rede IPTV (DVB, 2010b). Esses padrões definem serviços e interfaces de programação de aplicativos (APIs) para serem usadas pelos desenvolvedores de aplicativos de iTV para os STB e os provedores de conteúdo televisivo. A especificação da camada de transporte define como o aplicativo de iTV vai ser transmitido junto com o *streaming* de vídeo.

O padrão japonês chama *Integrated Services Digital Broadcasting* (ISDB), e foi o padrão escolhido como base para o mercado brasileiro. No caso do Brasil, foi criado um padrão baseado no ISDB terrestre (ISDB-T), chamado de Sistema Brasileiro de TV Digital (SBTVD ou ISDB-Tb). O SBTVD incorporou algumas inovações ao padrão japonês, sendo elas:

- Codificação H.264/MPEG-4: O ISDB utiliza o *codec* H.262/MPEG-2 como padrão de compactação de áudio e vídeo, já o brasileiro utiliza a versão mais atual, o H.264/MPEG-4.
- Apresentação de imagem com 30 *frames* por segundo (fps) inclusive para dispositivos móveis: O canal de 1 segundo, direcionado para os dispositivos móveis, possuía somente 15 fps, e teve essa taxa aumentada.
- *Middleware* Ginga: O *middleware* utilizado pelo ISDB é chamado de *Broadcast Markup Language* (BML), e não incorpora algumas funções avançadas suportadas no *middleware* brasileiro, chamado de Ginga.

Com as inovações tecnológicas desenvolvidas no SBTVD, uma comissão Brasileiro-Japonesa unificou os padrões ISBD-T e o SBTVD criando o ISDB-T Internacional. Esse padrão foi oferecido pelo mundo e adotado pela maioria dos países da América Latina, com exceção da Guiana Francesa e a Colômbia que adotaram o DVB, a Guiana e Suriname que ainda não

definiram oficialmente o padrão, e o Uruguai recentemente iniciou a discussão para alterar o seu padrão para o ISDB-T (GLOBO.COM, 2010).

As mudanças no *codec* de compressão propostos no SBTVD permitiram uma melhora na qualidade e da resolução da imagem, assim como o aumento do fps para o canal de 1 segundo para os dispositivos móveis. Sobre o *middleware*, chamado Ginga, possui diversos diferenciais interativos em relação ao BML (SOUZA FILHO; LEITE; BATISTA, 2007). O BML foi definido em 1999 e é baseada no protocolo HTML e XML, e que suporta também classes CSS e ECMAScript. Já o Ginga é dividido em duas partes, o Ginga-NCL e o Ginga-J, sendo o primeiro possuindo uma função similar ao do BML. O Ginga-J é a parte inovadora introduzida pelo SBTVD que suporta aplicativos Java e coloca o padrão compatível com o padrão europeu. Sua arquitetura pode ser vista na Figura 3.

Figura 3 - Arquitetura Ginga

O Ginga-J possui diversas APIs para o desenvolvimento de aplicações interativas para TV, vistas na Figura 4. Uma das inovações do Ginga-J está o suporte à interatividade, que chega até dispositivos terceiros ao STB, ou seja, um programa de televisão pode possuir um aplicativo de iTV que se conecta a um dispositivo Bluetooth próximo ao STB. Este é um dos recursos únicos criado pelo SBTVD na fase de especificação do padrão e que permite, por exemplo, utilizar um telefone celular conectado ao STB como canal de retorno.

Figura 4 – As diversas APIs do Ginga-J

2.6. Sistemas de Recomendação

Com o crescente número de assuntos oferecidos para as pessoas nos dias de hoje, e o grande número de opções de escolha de produtos e serviços, surgiu a necessidade de se realizar uma filtragem prévia do conteúdo de forma a facilitar a escolha do cliente, potencializar a venda de algum produto/serviço específico, ou fidelizar um clientes oferecendo facilidades. Isso está começando a acontecer no mercado televisivo, já que nos canais disponíveis tanto em TV paga quando na TV aberta, o telespectador atualmente perde muito tempo buscando algum programa de interesse. Essa prática do telespectador de buscar programas foi chamada de *zap*, assim muito tempo é perdido *zapeando* pelos canais de TV. O tempo de *zap* só tende a aumentar com a chegada da TV por internet via banda larga, permitindo que qualquer pessoa possa ter seu canal de TV, e assim aumentando a quantidade de canais para alguns milhares.

Para realizar uma filtragem do conteúdo disponível ao telespectador, foi desenvolvida uma série de abordagens para realizar recomendações de opções que possuam grande probabilidade de agradar ao telespectador. Assim, os sistemas de recomendação existentes na atualidade podem ser classificados em três tipos:

1. **Sistemas baseado em conteúdo:** Esse tipo de recomendador utiliza as informações do usuário para modelar seu perfil e fazer as recomendações. Essas informações podem ser, por exemplo, seu histórico ou informações socioeconômicas inseridas pelo usuário.

2. Sistemas colaborativos: Neste tipo de sistema, as recomendações são feitas comparando os hábitos similares de diversos usuários. Ou seja, esse tipo de recomendador apresenta recomendações do tipo “usuários que gostaram desse filme, gostaram também desse outro”.
3. Sistemas híbridos: Finalmente, o uso de recursos dos dois tipos de sistema o classifica como um sistema híbrido.

ADOMAVICIUS e TUZHILIN (2005) descreveram as características de cada um dos tipos de sistema com os trabalhos mais relevantes da época, realizando comparativos entre as diferentes abordagens. Um dos recomendadores mais conhecido é o da loja online Amazon.com, que utiliza a abordagem colaborativa para oferecer produtos comprados por outras pessoas que olharam um produto similar ao acessado pelo usuário, conseguindo assim um aumento significativo em sua receita (LINDEN; SMITH; YORK, 2003).

O ato de assistir TV pode ser ativo ou passivo, conforme estudado por TAYLOR&HARPER (2002). Os autores classificaram os hábitos de busca de conteúdo por parte do telespectador para ajudar no desenvolvimento de guias de programação eletrônicos. Em sua pesquisa, os telespectadores podem seguir três hábitos televisivos:

- 1) Telespectadores ligam a TV ao chegarem a casa: Esses telespectadores ligam a TV e deixam em qualquer canal só para gerar um som de fundo, às vezes até com ninguém assistindo. Nomeado de *switching-on-to-switch-off* (ligando para desligar), em que o telespectador age passivamente e não se importa em nada com a programação do canal escolhido.
- 2) Telespectador do meio da noite: É um hábito coletivo, onde normalmente mais de uma pessoa assiste a TV ao mesmo tempo, e programas específicos como jornais ou novelas.
- 3) Telespectador de tarde da noite: Formado normalmente por público adulto que assiste a programas muito específicos sozinhos. Hábito mais ativo dos três.

Assim, construir um sistema de recomendação de programas de TV é uma tarefa muito mais complexa do que um recomendador de filmes ou músicas, pois os telespectadores quando estão em seu momento passivo, podem gerar muito ruído nos geradores de perfil. Esse problema é prontamente resolvido com o uso de ferramentas de identificação automática dos telespectadores, que detecta quando o usuário deixou a TV ligada sem estar assistindo e quando várias pessoas estão presentes na sala de TV.

2.6.1. Movilens, Netflix, IMDb e Allmovie

Um sistema de recomendação, tanto baseado em conteúdo quanto colaborativo, necessita de um volume inicial de dado para realizar as recomendações. No caso do sistema baseado em conteúdo é necessário um histórico do telespectador para poder criar o seu perfil. Já no sistema colaborativo, um histórico muito maior e de um universo muito mais abrangente de pessoas é necessário para gerar as recomendações.

Assim, para atualizar o sistema com dados reais foi feita uma busca por bases de dados que disponibilizam informações de históricos de programas de TV assistidos que poderiam ser utilizadas no sistema. A base teria que conter no mínimo registros dos usuários, o programa ou canal que cada um assistiu e as informações de data/hora. Porém, as poucas bases de dados de televisão encontradas ou não eram disponibilizadas a público, como o IBOPE, ou eram proprietárias (E-POLL, 2010). O mesmo já não acontece com bases de dados de filmes. Projetos como o Movielens disponibilizam bases de dados com o vetor de dados usuário, filme, nota e data/hora (GROUPLENS RESEARCH, 2010). Existem três diferentes bancos de dados disponíveis:

- 100 mil notas para 1682 filmes dados por 943 usuários
- 1 milhão de notas para 3900 filmes e 6040 usuários
- 10 milhões de notas e 100 mil palavras-chave para 10681 filmes e 71567 usuários

Outra base de dados disponível é uma fornecida pela empresa Netflix (BENNETT; LANNING, 2007). Em 2006, essa empresa anunciou um prêmio de 1 milhão de dólares para quem conseguisse desenvolver um algoritmo que melhorasse a taxa de acerto do seu software de recomendação, denominado Cinematch (BENNETT, 2006), em 10%. A competição começou em outubro de 2006 e o resultado de 10% de melhora foi atingido em junho de 2009, quase 3 anos depois. Como consequência do anúncio deste prêmio, uma base de dados de 100 milhões de notas feitas por 480 mil usuários em um universo de 17.700 filmes foi disponibilizada publicamente. A base de dados contém as quádruplas: usuário, filme, data da nota, nota; sendo que o filme possui, em uma base de dados separada, as informações do título e o ano de lançamento. A empresa também anunciou que algumas perturbações foram inseridas na base de dados para proteger a privacidade dos seus clientes, como notas apagadas, notas ou datas alternativas inseridas, ou data de alguma nota modificada.

Os filmes da base de dados da Netflix possuem somente as informações de data de lançamento e o nome completo, sendo necessário o gênero do filme para criação do perfil do telespectador. Outro agravante é que a data de lançamento pode ser do DVD ou do lançamento no cinema, e o título não necessariamente é o título original, podendo ser uma tradução para o inglês em caso de filmes estrangeiros. Já na base de dados do Movielens, todos os filmes estão relacionados com o site *Internet Movie Database* (IMDb) (IMDB, 2009). O IMDb é um site especializado em dados de filmes e tópicos relacionados, como atores, diretores, fatos relevantes, etc. Este site classifica os filmes em 27 gêneros, sendo que cada filme pode ter um ou mais gêneros associado. São eles:

Tabela 1 - Gêneros do IMDb

Ação	Aventura	Animação	Biografia
Comédia	Crime	Documentário	Drama
Familiar	Fantasia	Film-Noir	Game-Show
História	Terror	Música	Musical
Mistério	Notícias	Reality-TV	Romance
Ficção Científica	Esporte	Talk-Show	Suspense
Guerra	Velho Oeste	Adulto	

Outro site de banco de dados de filmes disponível é o Allmovie (ALLROVI, 2010). Neste site, a classificação de gêneros dos filmes é feita utilizando informações providas pelos produtores, onde na maioria dos casos é utilizado um ou dois gêneros para descrever o filme, e as demais classificações ficam como subcategoria. Um exemplo é o filme “Marte Ataca!”, que no IMDb é classificado simultaneamente como Ação, Comédia, Fantasia, Ficção Científica e Suspense, e no Allmovie é classificado como Ficção Científica somente. Os tipos de classificação de filmes deste *site* estão divididos em 12 principais categorias, sendo elas Drama, Comédia, Terror, Suspense, Ficção Científica, Musical, Ação, Velho Oeste, Romance, Mistério, Família e Histórico.

2.7. Processadores Multi-core Heterogêneos

O número de transistores existentes nos processadores vem crescendo a cada ano, de acordo com a Lei de Moore. Porém, a velocidade dos processadores não está aumentando na mesma velocidade. Isso é devido à dificuldade de dissipar grandes quantidades de calor geradas internamente no processador e os problemas elétricos que se tornam mais

significantes quando uma frequência muito alta é aplicada aos componentes, como capacitância parasitária por exemplo.

Por essas razões, a tendência atual da indústria de microeletrônica é aumentar a capacidade de cálculo de um processador através do aumento do número de núcleos dentro de um mesmo *chip*, em vez de focar no aumento da frequência e recursos de um único núcleo. Assim, processadores de múltiplos núcleos começaram a surgir no mercado, e atualmente a maioria dos computadores existentes no mercado possuem dois ou mais núcleos, inclusive aparelhos celulares *smartphones* e outros dispositivos móveis. Esta tendência de aumentar o número de núcleos nos processadores está tão forte nos últimos anos que processadores com mais de 100 núcleos já podem ser encontrados (Butts, 2007).

Outra forma de contornar as restrições físicas do aumento de frequência e conseguir aumentar a capacidade de processamento do CPU é alterar a sua arquitetura, para seja possível realizar mais cálculos por ciclo de relógio. Com isso, junto com os processadores convencionais de múltiplos núcleos, vêm surgindo processadores de arquitetura diferenciada. Um dos recursos que estão sendo incorporados é a inclusão de instruções que permitem ao processador realizar cálculos vetoriais com uma instrução e em poucos ciclos de relógio. Esse tipo de instrução se chama *Single Instruction Multiple Data* (SIMD).

Enquanto os processadores atuais são *Single Instruction Single Data* (SISD) que operam somente um dado por instrução, os processadores SIMD podem fazer uma soma de vetores, por exemplo, utilizando somente uma instrução e um ciclo de relógio. Esse conjunto de instruções é muito comum em placas gráficas, ou *Graphic Processing Unit* (GPUs) para efetuar cálculos de geometria analítica de maneira mais eficiente.

Ultimamente, o fabricante de GPUs nVidia desenvolveu uma linguagem baseada em C denominada CUDA, que permite desenvolver aplicativos para ser executados na GPU. Assim, foi iniciada uma linha de pesquisa e de produtos identificados por *General Purpose Graphic Processing Unit* (GPGPU), em que as placas GPUs começaram a ser usadas em supercomputação para acelerar algoritmos científicos e de computação intensiva.

2.7.1. Processador Cell BE

O manuseio de toda essa massa de dados exigido pelos programas de recomendação e outros algoritmos de mineração de dados, exige muitos recursos computacionais, e ainda

assim pode ser uma tarefa demorada, mesmo nos computadores convencionais atuais. Assim, arquiteturas novas surgiram para executar alguns tipos de tarefas com melhor desempenho.

Em 2000, foi formada uma aliança entre a Sony, Toshiba e IBM para desenvolver um processador de arquitetura diferenciada para aplicações de processamento intensivo, entretenimento e multimídia. Deste ano então foi dado o início do projeto do processador *Cell Broadband Engine*, ou simplesmente Cell BE ou só Cell. No final de 2005 foi disponibilizado um simulador para testes de funcionamento e no início de 2006 a IBM anunciou o lançamento de um servidor *Blade* utilizando este processador.

O Cell BE possui 241 milhões de transistores, 9 núcleos de processamento, com capacidade de execução 10 *threads* simultaneamente e uma frequência de relógio máxima de 4 GHz. O que diferencia o Cell BE dos processadores convencionais é a existência de uma arquitetura de múltiplos núcleos heterogêneos. Essa arquitetura contém um núcleo Power PC 64 bits, chamado de *Power Processor Unit* (PPU) com capacidade de execução de dois threads simultâneos, e oito núcleos sinérgicos, ou *Synergistic Processor Unit* (SPU) para processamento de grande intensidade de dados (PHAM; AIPPERSPACH; BOERSTLER, 2006; DAY; HOFSTEE, 2005). Esses SPUs possuem capacidade de cálculo vetorial e suportam instruções SIMD para operar vetores com até 4 variáveis de ponto flutuante.

As interconexões internas e externas utilizaram as tecnologias mais modernas na época para atingir altas velocidades. Um anel de interligação interno entre o PPU, os SPUs e a memória permite uma banda de mais de 300 GB/s, que é usada para transmissões simultâneas de dados de até três transferências *Direct Memory Access* (DMA) para um SPU e o PPU, e também entre distintos SPUs. Com o meio externo, o Cell BE atinge velocidades de 25,6 GB/s para se comunicar com a memória e para I/O tem-se 35 GB/s para saída e 25 GB/s para entrada.

2.7.2. Processadores Embarcados CPU+GPU

Com a popularização de dispositivos móveis como *tablets* e *smartphones*, a indústria de aplicativos embarcados começou a demandar uma maior capacidade computacional desses dispositivos. Assim, os fabricantes de CPUs e placas gráficas começaram a desenvolver linhas

de componentes voltados para esse segmento de mercado. Em 2011, diversos fabricantes anunciaram processadores *multi-core* para dispositivos portáteis.

Recentemente (2011), começaram a surgir processadores embarcados em dispositivos móveis com GPUs que suportam CUDA. Desta forma, esses GPUs podem ser usados para acelerar a execução de aplicativos não gráficos embarcados. Um dos GPUs disponíveis no mercado que estão sendo vendido junto com *Netbooks* e *Nettops* é o nVidia Ion (NVIDIA, 2011a). Essa placa gráfica, em sua segunda versão, possui 16 núcleos GPUs com 512 Mb de memória DDR3, com um consumo de 12 Watts. Na área de *tablets* e *smartphones*, os processadores usados nesses dispositivos são um híbrido de CPU e GPUs. A nVidia lançou uma família de *chips* chamada de Tegra (NVIDIA, 2011b). A última versão lançada em 2011 é o Tegra 3, que possui um processador *quad-core* ARM 1,5 GHz e 12 núcleos GPU. A próxima geração prevista para 2012, o Tegra 4 possuirá 8 CPUs ARM junto com até 64 núcleos GPUs.

Outros fabricantes de semicondutores possuem também suas famílias de processadores híbridos, como o Snapdragon da Qualcomm, o OMAP da Texas Instruments, os processadores A da Apple, entre outros. O Snapdragon a ser lançado em 2012 pode ter até quatro núcleos ARM 1,5 GHz e GPUs da família Adreno.

Esses processadores permitem que as aplicações executadas nos dispositivos embarcados utilizem as GPUs como aceleradores GPGPU. Assim, as aplicações podem ser desenvolvidas para explorar o paralelismo e as instruções SIMD, e assim serem executadas em um menor tempo.

2.8. Star Superscalar

Hoje em dia, os computadores estão se tornando cada vez mais poderosos, aumentando o número de núcleos e oferecendo diversos tipos de processadores e aceleradores para usos específicos. Para programar o Cell BE, por exemplo, é necessário gerenciar o sincronismo entre todos os *threads*, todas as comunicações entre os SPUs e o PPU e também gerenciar todas as transferências DMA. Assim, o trabalho do programador está se tornando cada vez mais complexo, pois trabalhar com estes tipos heterogêneos de arquiteturas e explorá-las ao máximo não é uma tarefa fácil. Por este motivo é necessário uma interface de programação intuitiva e mais centrada na forma de pensar humana, para permitir a exploração dos recursos da arquitetura de forma transparente ao programador. O modelo do Star

Superscalar (StarSs) ajuda o programador a desenvolver aplicações paralelas, possuindo atualmente quatro implementações distintas para diferentes arquiteturas: Cell Superscalar (CellSs), SMP Superscalar (SMPsSs), Nested Superscalar (NestedSs) e o GPU Superscalar (GPUSs). Todos eles podem ser utilizados com C, C++ e Fortran.

2.8.1. Cell Superscalar

O CellSs é um modelo de programação baseado em tarefas para o Cell BE. De um código sequencial dotado de anotações usando as diretivas `#pragma`, similar às usadas no OpenMP (OPENMP, 2011), ele pode explorar o paralelismo da aplicação dividindo sua execução em varias pequenas partes, chamadas de tarefas. A biblioteca de tempo de execução se responsabiliza por manter constantemente atualizado o diagrama de dependência de dados, de forma a executar as tarefas que satisfazem essa dependência, ou as que necessitam aguardar até que o dado necessário seja calculado, explorando assim o paralelismo intrínseco da aplicação (PÉREZ et al., 2007).

As tarefas são executadas nos SPUs do Cell BE e não podem conter sub-tarefas. Assim, uma tarefa tem que ser uma parte de código que acessa somente os seus parâmetros e não pode acessar variáveis globais, ou dados externos. O biblioteca de tempo real do CellSs, além de coordenar a execução de todas as tarefas, se responsabiliza por gerenciar as transferências entre o PPU e o SPU de maneira transparente ao programador.

2.8.2. SMP Superscalar

Similarmente, o SMPsSs é equivalente ao CellSs, porém usando CPUs convencionais em vez do Cell BE. Ele foi desenvolvido para focar nas arquiteturas de múltiplos núcleos e múltiplos processadores simétricos (SMP). De forma transparente para a aplicação, diversos *threads* são criados internamente, e são executados em distintos núcleos do computador de forma a executar todas as tarefas. As tarefas do SMPsSs possuem as mesmas limitações do que as tarefas do CellSs (PEREZ; BADIA; LABARTA, 2008), não podendo acessar dados externos à tarefa nem variáveis globais.

2.8.3. Nested Superscalar

O NestedSs é um modelo de programação baseado em tarefas para plataformas heterogêneas e/ou hierárquicas e foi primeiramente implementando para a arquitetura Cell BE. Ele possui dois níveis de tarefas, se adaptando à arquitetura alvo. No caso do Cell BE, o primeiro nível é executado no PPU e o segundo nível no SPU. As tarefas hierárquicas permitem dividir a computação em partes menores ainda, explorando ainda mais o paralelismo da aplicação. Do mesmo modo, o NestedSs oculta sua arquitetura e controle complexos ao mesmo tempo em que oferece uma interface simples de programação usando as diretivas `#pragma` (PLANAS et al., 2009).

A principal diferença entre o CellSs e o NestedSs é o fato de que as tarefas do NestedSs possuem dois níveis. Assim, tarefas pertencentes ao primeiro nível podem criar tarefas de segundo nível, mas não o caminho contrário. Enquanto as tarefas do CellSs são todas executadas nos SPUs do Cell BE, o NestedSs pode enviar tarefas para serem executadas no PPU e no SPU, permitindo o uso de mais de um processador Cell BE simultaneamente pela aplicação.

2.8.4. GPU Superscalar

Recentemente, foi iniciada uma implementação do StarSs para a arquitetura de placas gráficas (GPUs). Essa implementação foi chamada de GPUSs e possui algumas das características do CellSs, junto com outras do SMPSSs e NestedSs. Como o SMPSSs, as tarefas do GPUSs podem ser executadas nos diversos CPUs existentes no computador. Porém, o mecanismo de envio de tarefas para as GPUs se assemelha ao NestedSs, onde somente tarefas de segundo nível são enviadas à GPU (AYGUADÉ; BADIA; IGUAL, 2009). Assim, o GPUSs permite usar tanto os diversos processadores simétricos disponíveis, quanto também às várias placas gráficas conectadas a esses CPUs. Porém seu desenvolvimento estava em progresso e seu uso ainda não estava estável. Com isso, o GPUSs não suportava todas as funcionalidades para fazer uma implementação completa de um algoritmo complexo até o término desta pesquisa.

2.9. Conclusão

Como pode ser visto, várias são as tecnologias envolvidas nesta pesquisa, começando com sistemas de RFID e o uso do Bluetooth para identificação de pessoas, passando por padrões de TV digital e sistemas de recomendação, e finalizando com arquiteturas não convencionais de processadores. O Bluetooth vai servir de ferramenta para realizar a identificação automática dos telespectadores, sem necessitar colocar um identificador pelo controle remoto ou qualquer outro tipo de autenticação. Já o padrão de TV adotado no Brasil, o SBTVD, suporta o envio de aplicações para serem executadas em dispositivos conectados ao STB. Ou seja, um programa de TV pode enviar um *software* para o celular de um telespectador próximo ao STB.

Quanto aos sistemas de recomendação, podem ser classificados em: colaborativo, baseado em conteúdo e híbrido. O colaborativo funciona cruzando informações de outros usuários, enquanto que o baseado em conteúdo funciona analisando somente o histórico do usuário em questão. Os sistemas híbridos são os sistemas que incorporam funcionalidades dos dois sistemas, como o desenvolvido nesta tese.

Finalmente, as arquiteturas heterogêneas superam a capacidade computacional dos processadores convencionais, exigindo assim uma abordagem diferente na programação. O processador Cell BE, por exemplo, possui 9 núcleos, sendo 8 deles feitos para computação vetorial usando instruções SIMD. Já as placas gráficas presentes nos processadores embarcados também podem ser usadas como acelerador de cálculo, reduzindo o tempo de execução de aplicativos.

3. REVISÃO BIBLIOGRÁFICA

Diversos trabalhos foram publicados na área de análise de sistemas de recomendação e na área de comportamento de usuário. Quanto ao uso de dispositivos Bluetooth para identificação, também foram feitas várias pesquisas sobre o tema.

Primeiramente serão apresentadas as referências que utilizam Bluetooth como tecnologia de identificação, sendo para identificar pessoas ou objetos. A seguir, alguns sistemas de recomendação televisivos são apresentados citando seus diferenciais. Um grupo especial de sistemas de recomendação que utilizam tecnologias de identificação dos usuários é apresentado a seguir. Depois, os sistemas de recomendação para grupos desenvolvidos são apresentados, seguidos dos sistemas de recomendação que focam em escalabilidade.

3.1. Bluetooth

O uso de telefone Bluetooth para identificar pessoas foi proposto por (EAGLE; PENTLAND, 2006), que criaram uma aplicação chamada BlueAware. Este software é parte de um sistema de análise de interações sociais chamado Serendipity, que analisa as interações entre os diversos telefones Bluetooth que executam o BlueAware durante o dia-a-dia de uma pessoa. PAULOS&GOODMAN (2004) desenvolveram um sistema similar para encontrar o estranho conhecido (*Familiar Stranger*). Este sistema analisa as interações entre os telefones Bluetooth para encontrar pessoas que se encontram sempre em lugares como restaurantes, ônibus, etc, mas que não se conhecem. A definição de estranho familiar é a observação frequente de uma pessoa por outra, mas sem qualquer interação entre elas. Um pequeno aplicativo chama Jabberwocky foi desenvolvido para identificar os dispositivos com Bluetooth ativado localizados nas proximidades. Um dispositivo Bluetooth dedicado para essa aplicação com esse software embarcado com alguns LED coloridos e botões também foi desenvolvido.

A patente 7.215.976 descreve um *tag* de RFID que utiliza o padrão e frequência do Bluetooth para se comunicar com um leitor (BRIDGELALL, 2007). O *tag* inventado pode tanto se comunicar na frequência do Bluetooth, coexistindo com os dispositivos Bluetooth existentes, quanto se comunicar com leitores RFID de UHF. SIEGEMUND&FLORKEMEIER

(2003) utilizaram os recursos do Bluetooth como *tag* de RFID e os apelidam de BTnode. Os BTnodes são pequenos circuitos eletrônicos com um microcontrolador ATmega128 e um rádio 2,4 GHz, e que executa um *firmware* que implementa a pilha Bluetooth. Esses BTnodes foram desenvolvidos para serem conectados a um objeto, e se comunicarem com um telefone celular via Bluetooth. Assim o celular tem o papel de leitor de RFID e o BTnode funciona como um *tag*.

A diferença entre a pesquisa realizada nesta tese e os sistemas descritos nos outros artigos está no fato de que o telefone do telespectador somente irá se comunicar com o STB, já que a função do telefone com Bluetooth é somente realizar o *login/logout* automatizado do telespectador no sistema de recomendação. Desta forma, o celular não necessitará executar nenhuma aplicação específica, já é suficiente apenas ter a configuração do seu Bluetooth alterada para que o dispositivo responda as inquisições Bluetooth já é suficiente. Outra diferença é que nenhum *hardware* novo teve que ser desenvolvido para isso, o próprio telefone é o *tag* e o STB o leitor.

3.2. Sistemas de Recomendação para TV

O autor (SMYTH, 2007) descreve de forma completa a estrutura de sistemas de recomendação, com um foco em *e-commerce*. Na área de análise comportamental de usuários, HARA, TOMOMUNE e SHIGEMORI (2004) fizeram um trabalho de categorização de telespectadores baseado em canais assistidos no Japão, relacionando o gênero de programa preferido com a idade e sexo do telespectador. Ele descreve também como seus resultados poderiam ser aplicados em um sistema de recomendação de canais, escolhendo programas favoritos das pessoas que possuem mesma a idade e sexo do telespectador. VILAS et al. (2007) realizaram um trabalho de quatro anos de análise de usuários e de modelagem do seu comportamento para o caso específico de VoD. Seu foco foi no número de interações do tipo pausa, avançar, parar, etc, nos vídeos selecionados. Estes trabalhos fornecem modelos para desenvolver sistemas de recomendação e análise de perfil, e serão fonte de conhecimento para esta pesquisa. Um sistema chamado TV Genius cria listas de recomendações utilizando um algoritmo híbrido e rede Bayesiana (TV GENIUS, 2011). O algoritmo monta um mapa de relevância para identificar as similaridades entre os diferentes programas de TV. Outro sistema existente é o *Personalised Television Listings* (PTV) que

utiliza o metadado dos programas para compilar uma lista de recomendações para um dado perfil (SMYTH; COTTER, 2000). Essa lista é disponibilizada pela internet como um guia de TV, e sua montagem utiliza um algoritmo baseado em conteúdo e elementos colaborativos.

YU e ZHOU (2008) desenvolveram um sistema de recomendação cliente/servidor, em que o cliente pode ser executado em aparelhos de baixa capacidade computacional. Este sistema é colaborativo e utiliza os dispositivos ubíquos, como o controle remoto, celular e STB para capturar as informações necessárias para montar o perfil do usuário. Também com um foco nos dispositivos de baixa capacidade computacional, LAI, HUANG H. e HUANG Y (2007) desenvolveram um recomendador que pode ser executado em um controle remoto. O algoritmo utiliza uma matriz de três dimensões que armazena o canal, o usuário e o horário em cada um dos eixos. O desenvolvimento do sistema teve como foco o baixo consumo de energia e de recursos computacionais, e o ambiente de teste foi uma plataforma 8051 utilizada como controle remoto.

Analizando os trabalhos que exploram a recomendação de publicidade, uma tese de mestrado detalha um sistema de seleção de publicidade dirigida (ABRANTES, 2009). O sistema foi chamado de TargetAd, e ele utiliza o perfil dos telespectadores para selecionar os comerciais a serem apresentados. Outro controle do sistema garante que todas as publicidades possuem igual possibilidade de serem apresentadas. Este sistema necessita de uma identificação prévia do telespectador para que seu perfil seja analisado, esses perfis são criados estaticamente, e as únicas informações que eles contêm são a idade e o sexo da pessoa. O iMedia é outro sistema de recomendação de propagandas televisivas que é executado no STB (BOZIOS; LEKAKOS; SKOULARIDOU, 2001). Assim, o iMedia sobrepõe os comerciais transmitidos pelos canais por um comercial dirigido ao público presente em frente ao televisor. O iMedia, apesar de ter um artigo publicado, é um sistema comercial e não fornece muita informação sobre como o perfil é criado, porém, no artigo é mencionado que o sistema classifica o perfil do telespectador em um cluster, e com base nesse cluster é escolhida a publicidade a ser apresentada.

Outros sistemas de seleção de publicidade também são proprietários e não disponibilizam muitas informações técnicas sobre o funcionamento do algoritmo de recomendação. Entre eles está o Advatar e o intelliSpot (NIELSEN, 2011; VISIBLE WORLD, 2011). O Advatar também é executado no STB e utiliza o perfil do usuário para fazer as

recomendações, enquanto que o intelliSpot utiliza informações geográficas do telespectador para enquadrá-lo em um público alvo e selecionar a publicidade.

3.3. Sistemas de Recomendação com Recurso de Identificação

Quanto aos sistemas completos similares ao proposto, existe um sistema de guia eletrônico para museu usando computador de mão (PDA) chamado PEACH, que analisa o comportamento do usuário e cria dinamicamente o conteúdo das apresentações (STOCK et al., 2007). Esse sistema usa infravermelho para localizar o visitante no museu e analisa o comportamento baseado nas exposições que ele visitou, quais itens expostos ele visitou e o tempo gasto em cada um. Outro projeto chamado EPG-Board é um sistema de análise e recomendação de perfil com características de rede de contatos, que permite ao telespectador adicionar amigos, ver o que eles estão assistindo e trocar mensagens (IATRINO; MODEO, 2007). O EPG-Board possui um sistema de agendamento de programas que se deseja assistir. Seu sistema de recomendação usa as informações dos canais assistidos, dos programas agendados e de informações de nota dada pelos usuários sobre o programa assistido. Outra característica deste sistema é que o usuário pode colaborar submetendo sinopses e informações do gênero dos programas, chamados de metadados, que não estão cadastrados. Essas informações são compartilhadas para os outros usuários do sistema. TSUNODA e HOSHINO (2006) desenvolveram um sistema de recomendação que analisa os programas assistidos e usa um cadastro preenchido pelo telespectador com suas preferências de programa e palavras-chave de metadados preferidas. Esse sistema também compara os filmes assistidos pelas pessoas de mesmo perfil e recomenda para o telespectador os filmes que a comunidade já viu e que ele ainda não assistiu. Na Europa, um grupo de trabalho chamado *Live Staging for Media Events* formado em 2006 se propôs a desenvolver sistemas para reprodução de transmissões ao vivo com várias imagens simultâneas de diferentes pontos de vista (LIVE, 2007).

Entre os trabalhos desenvolvidos, existe um artigo publicado no IEEE sobre autenticação de pessoas no STB por reconhecimento facial para identificar o telespectador (MLAKAR; ZALETELJ; TASIC, 2007). Esse sistema foca em captar alterações de expressões faciais, consegue identificar somente uma pessoa por vez e só consegue boas taxas de acerto com condições ideais. Finalmente, LIM et al. (2007) apresentam algoritmos de identificação de

gosto de usuário baseado em vetores, com dois estágios. Neste mesmo trabalho foi mostrada a implementação do algoritmo aplicado para selecionar comerciais para os telespectadores, baseado na base de dados de uma operadora da Coreia. Outro projeto desenvolvido pela Microsoft, chamado EasyLiving, focado no desenvolvimento de espaços inteligentes, utiliza um leitor de impressão digital no controle remoto para identificar os usuários (KRUMM; SHAFER; WILSON, 2001). As justificativas apresentadas pelos autores deste tipo de identificação são as imprecisões dos sistemas de detecção automática disponíveis na época, e a questão da privacidade, em que o usuário tem que saber quando ele está sendo observado. Outro sistema de recomendação de TV integra um sistema de reconhecimento facial para identificar o telespectador (ZUO; WITH, 2005).

O diferencial do sistema proposto em relação aos existentes é a capacidade de identificação de mais de uma pessoa simultaneamente e em tempo real, e também detectar quando as pessoas deixam as proximidades do televisor. Essa diferença exige alterações nos sistemas de análise de perfil e de recomendação de conteúdo para levar em consideração a análise grupal, e não só a individual.

3.4. Sistemas de Recomendação para Múltiplos Usuários

Recentemente começaram a surgir algoritmos para analisar comportamento de grupos e realizar recomendações para mais de uma pessoa. Porém estes sistemas ainda carecem de uma maneira eficiente de detectar e atualizar os telespectadores presentes. A mudança de abordagem de um sistema de recomendação de usuários únicos para um sistema de recomendação de grupos traz novos desafios e problemas, apontados por JAMESON (2004). Entre eles está a mescla das preferências, para quem será enviada a recomendação e quem decidirá dentre as opções existentes. Esses desafios são levantados em algumas das pesquisas já feitas nesta área.

YU et al. (2006) definem em seu trabalho três possíveis abordagens para sistemas de recomendação para grupo, sendo elas:

- 1) Agente de grupo: Criação de um usuário que representaria o grupo de telespectadores;
- 2) Mescla de recomendações: Primeiro cria a recomendação para cada um dos usuários logados, e depois faz a mescla das recomendações;

- 3) Mescla dos perfis: As características da cada telespectador são juntadas e é gerado um perfil comum que serve como referência para a recomendação.

Em seu trabalho, a estratégia implementada foi a terceira, a de mesclar os perfis. No caso do 2S ACT, a abordagem adotada foi a segunda, de mesclar as recomendações e o porquê desta opção será detalhado nos capítulos seguintes.

Outro sistema de recomendação para grupos foi desenvolvido por GARTRELL et al. (2010). Em sua pesquisa, os autores montaram um grupo de estudo por dois anos para definir como as relações do grupo influem nas recomendações globais. Em seus estudos de decisões tomadas por grupos, foram definidas três opções: 1) satisfação média; 2) mínima insatisfação; e 3) máxima satisfação. A satisfação média é alcançada realizando uma média das recomendações dos membros do grupo. Assim, a nota global será essa média. Já na mínima insatisfação, a nota é calculada de acordo com a pior nota dada por algum dos membros. Reciprocamente, a máxima satisfação é quando a nota do grupo é a nota mais alta dada por um dos membros. Assim, detectadas as estratégias de decisão de grupos, o autor buscou a relação entre a estratégia de decisão do grupo e a relação social entre os seus membros. Finalmente, um último fator é a experiência de algum membro do grupo. Caso um participante tenha visto muito mais filmes do que outro membro, a sua nota terá uma relevância maior no cálculo da média final.

BONNEFOY et al. (2009) criaram um sistema de recomendação de grupos em ambiente televisivo que não necessita autenticação de usuários. O seu funcionamento é baseado em clusters de recomendações e um *feedback* do usuário que informa se a recomendação foi satisfatória ou não. Quando a TV é ligada, uma recomendação é feita. Caso essa recomendação agrade, o usuário ou o grupo de usuários respondem ao sistema uma sinalização positiva da recomendação. Com isso, os clusters que possuem esse programa são promovidos e outras recomendações destes clusters serão oferecidas. Caso o usuário avalie negativamente a recomendação, as recomendações deste cluster não serão mais enviadas aos usuários. Assim, não importa se é um telespectador ou se é um grupo, porque o sistema reconhece somente um perfil, seja representando um usuário ou a mescla dos perfis dos telespectadores de um grupo.

Finalmente, o trabalho publicado pelos brasileiros LUCA e ZORZO (2009) está bem alinhado ao 2S ACT. A publicação descreve um sistema de recomendação multiusuário para o sistema de TV brasileiro, chamado de RePTVD. O RePTVD classifica o grupo de forma

implícita, sem intervenção dos telespectadores. Para isso, o sistema monitora os canais assistidos e o tempo de permanência em cada um deles, e usa o algoritmo Apriori como ferramentas de mineração de dados para extrair as informações para as recomendações. Juntamente com o C4.5, o uso do k-Means é avaliado no artigo, porém pela abordagem de identificação de grupo em vez das pessoas individualmente, este algoritmo não gerou bons resultados. Como o 2S ACT identifica cada pessoa do grupo, uma a uma, o algoritmo adotado pode ser baseado no k-Means.

3.5. Sistemas de Recomendação Escaláveis

Uma última linha de pesquisa foi analisada quanto aos trabalhos desenvolvidos, que é a de sistema de recomendação escaláveis. ROSACI, SARNÉ e GARRUZZO (2009) desenvolveram um sistema distribuído de recomendação chamado de MUADDIB. A área foco desse sistema é a área de recomendação para sites *web*, como sites de compra por exemplo. O MUADDIB é um sistema de recomendação que permite a diferenciação das recomendações baseado no ponto de acesso do usuário. Caso o usuário esteja acessando de um telefone celular, as recomendações são diferentes de quando o mesmo usuário acessa de um PC, por exemplo. Assim o sistema permite a criação dessa regra, sendo esse o diferencial do MUADDIB em comparação com os outros sistemas equivalentes. O MUADDIB também usa um mecanismo de clusterização de usuários baseado no algoritmo CLARANS para detectar similaridades entre os diversos usuários. A plataforma usada para transformar o sistema em um sistema distribuído foi o JADE, que gerencia a comunicação entre os diversos serviços. Assim, a estratégia do MUADDIB para suportar a escalabilidade é por meio da adição de mais servidores na rede, gerando um cluster.

Outro sistema de recomendação que explora a escalabilidade de uma forma diferente é o sistema usado pelo site Youtube.com (DAVIDSON et al., 2010). Como o site tem que gerar recomendações para milhões de usuários e bilhões de eventos de atividade em sua base de dados, a abordagem do autor foi desenvolver um sistema que fica constantemente pré-computando as recomendações, em vez de um sistema que gera as recomendações em tempo real. A vantagem desta abordagem é que os recursos computacionais estão constantemente em uso de forma uniforme, sem picos nem vales de processamento, e a latência para a apresentação das recomendações fica muito menor do que processar toda a

informação quando solicitado pelo usuário. Este sistema atualiza constantemente uma tabela de relação de pares de vídeos, com a finalidade de encontrar os vídeos relacionados entre si. Então para gerar as recomendações, o sistema cruza tabela de relação de vídeos com as preferências e histórico do usuário, chamado de *seed data*. Com as recomendações geradas, um último processo é feito, o de ranquear essas recomendações. Três critérios são usados para isso, sendo eles a qualidade do vídeo, as especificidades do usuário e finalmente a diversificação.

Já mais diretamente relacionado com a realidade televisiva, ALI e STAM (2004) descrevem o algoritmo utilizado pelo sistema TiVo para gerar as recomendações (TiVo, 2011). O sistema do TiVo possui uma abordagem colaborativa, baseado na avaliação explícita e implícita do telespectador sobre o programa assistido. A avaliação explícita é feita pelo controle remoto, selecionando o botão de polegar para cima, ou *thumbs-up*, para avaliação positiva, e polegar para baixo, ou *thumbs-down*, em caso contrário. Já a forma implícita consiste em que, caso o usuário decida gravar um programa que foi avaliado negativamente, o peso negativo da avaliação é diminuído. Para reduzir a carga computacional no servidor central, o sistema de recomendação possui uma arquitetura cliente/servidor, onde os STBs são os clientes. Parte da computação envolvida na geração das recomendações é feita no STB, aumentando o custo de comunicação do sistema, porém reduzindo o tamanho do lado do servidor. O cálculo realizado por este último também é feito de forma distribuída e escalável, dividindo o dado a ser processado em partes, e o processo de cálculo também é dividido em camadas. Assim, cada parte de cada camada pode ser calculada em um servidor individual.

3.6. Conclusão

A linha de pesquisa de sistemas de recomendação não é nova, até porque muitos *sites* de venda *on-line* utilizam esses sistemas para oferecer melhores produtos para os clientes. A diferença do 2S ACT em relação aos outros sistemas são os elementos não computacionais envolvidos.

Primeiramente, o uso do Bluetooth para identificar telespectadores em uma sala de TV é um recurso nunca antes feito, sendo até passível de patente (FOINA; RAMIREZ-FERNANDEZ, 2008a). Com essa tecnologia, os usuários podem ser detectados automaticamente quando

chegam à sala de TV. A movimentação deste usuário para outro cômodo afastado da TV também é detectada, permitindo o sistema de recomendação sempre saber quem está presente e quem foi embora da sala, em tempo real.

O segundo elemento é o foco em sistemas de recomendação multiusuários. Como os sistemas anteriores utilizam majoritariamente autenticação pelo controle remoto para identificar os telespectadores, poucos pesquisadores focaram em sistemas de recomendação de grupos, porque este tipo de autenticação dificulta muito a praticidade do sistema. Com isso, poucos trabalhos sobre recomendação de grupos foram feitos, e cada um com a sua peculiaridade. Alguns trabalhos estimam o perfil do grupo, fazendo uma média dos perfis presentes, enquanto que outros tentam identificar um perfil de comportamento do grupo como um todo.

O elemento final é o uso de técnicas de programação paralela e distribuída para utilizar os recursos computacionais presentes nos processadores embarcados de múltiplos núcleos e de arquiteturas heterogêneas, enquanto que os algoritmos que exploram as mesmas técnicas não foram desenvolvidos para execução em sistemas embarcados.

4. METODOLOGIA

Este capítulo apresenta a metodologia adotada na pesquisa. Inicialmente foi feita a escolha da arquitetura do sistema, definindo o número de componentes, a função de cada um deles e como eles se comunicariam entre si. Depois da arquitetura, é exposta a metodologia de testes escolhida, explicando como foi escolhida a base de dados e como ela foi adaptada para possibilitar os testes. Também é explicado como os parâmetros do sistema foram alterados para suportar esta base. Finalmente, é apresentado o critério para comparação dos resultados.

4.1. Arquitetura

A implementação do 2S ACT permite uma arquitetura dividida em módulos independentes que podem ser implementados separadamente, e depois integrados entre si. Um desenho esquemático de como os blocos se comunicam pode ser visto na Figura 5. Essa comunicação, apesar de ocorrer localmente no STB, foi desenhada para que o sistema de recomendação possa ser centralizado para todos os telespectadores, sendo executados diversos servidores em um *datacenter*, de forma escalável.

Figura 5 - Comunicação dos serviços do 2S ACT

Os componentes necessários para fazer o 2S ACT funcionar como um todo são:

- Set-Top Box com um rádio Bluetooth embarcado para identificar os telefones celulares nas proximidades (STB RFID).
- Telefones celulares com Bluetooth habilitado e configurado em modo visível para responder as inquisições do STB (BtTag).
- Um *middleware* de gerenciamento, que é responsável por se comunicar com os diferentes sistemas, fazer toda a troca de mensagens e conversão de protocolos (STBM).
- Um sistema de gerência dos perfis existentes (PMS)
- Um módulo de análise de perfil do cliente baseado no histórico (UBA).
- Uma base de dados com o histórico de canais assistidos por todos os clientes.
- Um aplicativo de seleção de programação baseado nos perfis existentes e nos programas disponíveis (ReS).

O 2S ACT, quando embarcado em um STB possui todos os módulos sendo executados no mesmo *hardware*. Assim, o modelo esquemático deste uso pode ser visto na Figura 6.

Figura 6 - Diagrama dos módulos do 2S ACT embarcados em um STB.

4.2. STBM

Assim como os telefones celulares de hoje em dia têm que fazer mais coisas além do que realizar chamadas, os set-top boxes também estão indo pela mesma linha. No caso do 2S

ACT, uma das responsabilidades do STB é de gerenciar uma rede Bluetooth *Ad hoc* entre o STB e diversos telefones celulares.

Porém, como as tecnologias mudam a uma grande velocidade, a camada de comunicação entre o STB e os dispositivos foi isolada do resto do 2S ACT. E assim, foi um *middleware* como um dos componentes do sistema que foi chamado de *Set-top Box Middleware* (STBM). Este isolamento traz a vantagem que se, por exemplo, em um futuro próximo, o Wi-Fi, o NFC ou o USB sem fio (WUSB) virarem o padrão universal dos telefones móveis para troca de dados entre eles e identificação, a tecnologia de rede usada no 2S ACT pode ser facilmente trocada sem a necessidade de modificar todo o sistema, e sim somente algumas partes deste *middleware*.

O STBM é responsável por toda a comunicação Bluetooth do sistema, tanto de envio como de recebimento de dados entre o STB e os telefones. Uma de suas funções é realizar uma busca periódica dos dispositivos Bluetooth e comparar com uma lista de presença. Caso a relação dos dispositivos encontrados seja diferente da relação presente na lista de presença, a última é atualizada e as alterações são informadas à camada seguinte do sistema.

Porém, como frequentemente alguma interferência pode fazer com que um dos dispositivos presentes não seja detectado por alguns instantes, ou também fazer com que um dispositivo fora de alcance responda a alguma inquirição, o STBM precisa realizar algumas filtrações básicas das informações. Os três principais tipos de eventos a que um sistema de identificação por rádio está suscetível são (BAI; WANG; LIU, 2006):

1. Falso negativo: Este tipo de evento ocorre quando um transponder está dentro da área de leitura, porém não é lido. Isso pode acontecer por ruídos externos interferirem no momento da identificação, colisão de dados por diversos dispositivos transmitindo ao mesmo tempo ou algum material, como metal ou água, obstruindo a comunicação.
2. Falso positivo: O falso positivo é quando uma leitura não esperada de um transponder é feita. Um exemplo é um transponder que teoricamente está fora da área de leitura é lido temporariamente devido a uma interferência construtiva, ou algum problema na captura do identificador do transponder fazendo com que a sua identificação possua um número inexistente.

3. Leituras duplicadas: Os transponders estacionados dentro da área de leitura do STB irão responder todas as inquisições, gerando assim leituras duplicadas do mesmo item. Esses valores repetidos não precisam ser repassados para a camada superior do sistema, assim essas leituras duplicadas devem ser filtradas.

O STBM possui três filtros para tratar essas três situações. O falso negativo é tratado através de um tempo Xf_{neg} em que o telefone tem que ficar sem responder nenhuma inquisição para efetivamente considerar que o aparelho saiu da área. O falso positivo é tratado de maneira inversa, em que um transponder tem que responder consecutivamente um número Xf_{pos} de vezes para ser considerada uma leitura válida. E finalmente, a filtragem de leituras duplicadas é feita com um controle de estados dos transponders. Inicialmente, um telefone é considerado com estado “ausente”. Quando o telefone responde ao número Xf_{pos} de inquisições Bluetooth, o seu ID passa a ser considerado com status “presente” e suas leituras consecutivas são ignoradas. Então, quando o telefone fica Xf_{neg} tempo sem responder inquisições, seu status passa a ser “ausente”, retornando assim ao status inicial.

Após a filtragem das leituras, o STBM irá atualizar uma lista de presença, sempre que algum dos identificadores dos telefones Bluetooth mudarem de estado tanto de “ausente” para “presente” quanto vice-versa. A lista de presença será o ponto de integração entre o STBM e a camada seguinte, a responsável pela gerência dos perfis presentes.

Outra tarefa realizada pelo STBM é enviar informações para os telefones celulares quando requisitado pelas outras camadas. Por exemplo, recomendações dirigidas para somente um usuário são enviadas diretamente para o seu telefone celular, via Bluetooth, quando solicitado. Programas de iTV também podem futuramente explorar essa funcionalidade, enviado pequenos aplicativos para serem executados pelos usuários ou até contatos de anunciantes via cartão de visita eletrônico (vCard).

4.3. PMS

A camada logo acima do STBM é a parte do sistema responsável por gerenciar os perfis presentes, batizada de *Profile Manager Subsystem* (PMS). Entre suas funções também está a coordenação entre os diversos subsistemas, assim como a interação direta com o usuário via Bluetooth dos celulares.

O PMS observa a lista de presença constantemente, e a cada mudança dos membros presentes, ele altera os perfis a serem analisados pelo sistema de recomendação. O PMS contém uma tabela própria na base de dados que possui todos os perfis já analisados junto com informações adicionais, como a qual grupo pertence cada perfil. A cada novo ID adicionado à lista de presença, o PMS busca em sua base o perfil deste ID caso exista, para enviar ao sistema de recomendação. Caso o perfil não exista na base, ou seja, é um ID novo, o PMS inicia um processo de criação deste ID, que, caso tenha sido realizado com sucesso, dá seguimento à armazenagem de seu histórico para montagem do perfil.

O PMS também é o responsável por selecionar as recomendações a serem enviadas para o grupo. Como a abordagem escolhida neste sistema é a mescla de recomendações, a camada de recomendações envia ao PMS os programas recomendados de cada um dos perfis presentes. O PMS se encarrega de verificar quais dessas recomendações estão de acordo com os perfis de todos os telespectadores e as apresenta na interface gráfica. Caso exista alguma recomendação que agrada muito somente poucos membros da lista, essas recomendações são enviadas diretamente para os telefones celulares desses telespectadores através do STBM.

A análise dessas recomendações é feita de forma similar ao definido por KIRMEMIS e BIRTURK (2008), que usa três intervalos de valor para cada recomendação:

- Intervalo “tanto faz” – Caso o valor da previsão calculado pelo ReS de um dado programa para este usuário esteja neste intervalo, o programa não pesa nem positivamente nem negativamente, assim não são recomendados para o telespectador e nem influem na recomendação deste programa na avaliação do grupo.
- Intervalo “adoro” – Já os programas que se enquadram neste intervalo de “adoro”, significa que são avaliados positivamente para o telespectador e serão recomendados para o mesmo, e se qualificam para serem recomendados para o grupo.
- Intervalo “detesto” – Finalmente, os programas que tiveram sua nota calculada com valor dentro deste último intervalo pesam negativamente na avaliação do telespectador e assim não serão recomendados, e também podem anular o envio da recomendação deste programa para o grupo.

Como a base a ser usada é uma base de filmes existente, esses intervalos já estão implicitamente definidos pela própria definição da nota. A Netflix ou o GroupLens instruem os usuários a darem uma nota de 1 a 5 para os filmes, sendo que 4 e 5 são para quando o usuário gostou, 3 quando achou médio, e 1 e 2 caso não tenha gostado. Com isso, o intervalo de nota menor ou igual a 2 é definido como “detesto”; o intervalo maior ou igual a 4 é definido como “adoro”; e finalmente o intervalo maior que dois, e menor que 4 é definido como “tanto faz”.

Baseado nestes intervalos, a regra de mescla para o grupo foi definida. Todas as recomendações calculadas para o grupo são ordenadas. A ordenação das recomendações segue as seguintes regras, formando assim 4 grupos na ordem:

1. Primeiro ficam as recomendações que possuem o maior número de “adoro” e nenhum “odeio”. As recomendações que se enquadram neste caso são as chamadas “ótimas”.
2. A seguir, ficam as recomendações com o maior número de “adoro” e menor número de “odeio”. O conjunto das recomendações que possuem essa característica é chamado de “boa”.
3. Depois ficam as recomendações que possuem “tanto faz” e menor número de “odeio”. Essas recomendações que são deste tipo são denominadas “normais”.
4. Finalmente ficam as que só possuem “odeio”, que foram chamadas de “ruins”.

Com essa classificação, o PMS trata cada um dos grupos de recomendação de forma diferente. No grupo das “ótimas”, a recomendação é enviada para todos os membros do grupo, sendo apresentada na tela. Já quando a recomendação do programa para grupo é “boa”, e caso o número de “adoro” for maior que o número de “odeio”, então essa recomendação também é enviada para todos os presentes. Caso contrário, somente os telespectadores que tiverem a recomendação calculada no intervalo de “adoro” receberão essa recomendação via celular. O grupo de “normais” geralmente é descartado, com exceção de quando não existir nenhuma recomendação do grupo das “ótimas” nem das “boas”. Neste caso a opção do tipo “normal” é apresentada para todos os telespectadores. Para terminar, as recomendações do tipo “ruim” são automaticamente descartadas pelo sistema.

4.4. UBA

A parte do 2S ACT responsável por criar o perfil dos telespectadores baseado no seu histórico é o *User Behaviour Analyzes subsystem* (UBA). O UBA possui a base de dados de todos os programas vistos por cada um dos telespectadores cadastrados. O perfil gerado é um vetor de dimensão igual à quantidade de tipos de classificações de programas existente. Cada um dos valores do vetor corresponde ao índice de relevância daquela classificação para aquele telespectador. Sempre que novos registros de histórico são inseridos na base, o UBA atualiza os perfis dos telespectadores que assistiram a esses programas. Com certa periodicidade, o UBA também recalcula os clusters existentes e a qual cluster pertence cada telespectador.

Existem diversas formas de se gerar o perfil de um telespectador baseado em seu histórico de programas assistidos. Algumas informações dos metadados do programa podem ser usadas, como as palavras chave dos programas assistidos (FERNANDEZ et al., 2006). Já o 2S ACT utiliza somente o gênero dos programas assistidos, por ser uma primeira forma de filtragem de conteúdo por parte do telespectador, como visto em locadoras de filmes que classificam os filmes pelo gênero.

A forma da qual o 2S ACT define o perfil dos usuários é um pouco distinta. Inicialmente o foi pensado em gerar o perfil baseado nas notas atribuídas pelo telespectador a cada gênero. Porém, uma análise da base de dados e alguns estudos prévios mostraram que todos os perfis seguiam um padrão de ter preferência por Drama e por Comédia. Então foi definida metodologia a seguir para criar o perfil.

Primeiramente, para cada telespectador, o histórico dos programas assistidos é recuperado da base de dados. Depois, cada programa do histórico tem seu gênero relacionado com uma base de gêneros, e caso um programa tenha mais de um gênero, o registro é duplicado. O resultado final deste cruzamento de dados é uma tabela que demonstra a quantidade de vezes que o telespectador assistiu cada gênero. Um exemplo de resultado pode ser visto na Tabela 2.

Tabela 2 – Tabela de Gêneros do telespectador de ID 885013 da base da Netflix.

Programa	Gênero	Nota	Data
Dinosaur Planet	Animação	4	2005-10-19
Dinosaur Planet	Documentário	4	2005-10-19
Dinosaur Planet	Família	4	2005-10-19
X2: X-Men United	Ação	3	2005-10-21
X2: X-Men United	Sci-Fi	3	2005-10-21
X2: X-Men United	Suspense	3	2005-10-21
The Hebrew Hammer	Comédia	4	2005-10-21
Ed Wood	Documentário	4	2005-10-25
Ed Wood	Comédia	4	2005-10-25
Ed Wood	Drama	4	2005-10-25
Dogma	Aventura	4	2005-10-20
Dogma	Comédia	4	2005-10-20
Dogma	Fantasia	4	2005-10-20
Dogma	Mistério	4	2005-10-20
Kill Bill: Vol. 2	Ação	1	2005-10-20
Kill Bill: Vol. 2	Suspense	1	2005-10-20
The Matrix: Revolutions	Ação	3	2005-10-21
The Matrix: Revolutions	Aventura	3	2005-10-21
The Matrix: Revolutions	Sci-Fi	3	2005-10-21
The Matrix: Revolutions	Suspense	3	2005-10-21

O passo seguinte é calcular a porcentagem de cada gênero vista por esse telespectador, criando um gráfico de participação do gênero no perfil, visto na Figura 7.

Figura 7 – Gráfico do perfil de gêneros do telespectador 885013.

Um passo a mais é necessário para se extrair a informação dos perfis dos telespectadores. Independentemente do perfil do telespectador, as pessoas em geral

assistem aos programas e filmes que são mais famosos sem se preocupar com o gênero. Por exemplo, um mesmo telespectador assiste aos filmes Titanic, que é um filme documental histórico; Matrix, de ação e ficção científica; e Senhor dos Anéis, um filme de aventura e fantasia. Os três são de classificações muito distintas, porém o público em geral assiste a esses filmes mesmo não gostando do gênero. Com isso, grande parte dos perfis de gênero seguem um padrão na composição. Para retirar essa tendência média e evidenciar somente os gêneros preferenciais do telespectador, um perfil médio virtual é criado tirando a média de todos os perfis da base. Finalmente, o perfil de cada telespectador é subtraído pelo perfil médio, gerando o perfil de diferença. Esse perfil possui valores positivos e negativos para cada gênero. Os valores positivos representam os gêneros preferenciais do telespectador, enquanto que os valores negativos destacam os gêneros que o telespectador não gosta.

O sistema também funciona de forma implícita, ou seja, capturar as notas dos programas sem que os telespectadores tenham que atribuir uma nota via controle remoto. Desta forma, a nota de cada programa assistido por um dado telespectador é calculada com base na porcentagem de tempo que o telespectador assistiu a este programa. O momento em que o usuário visualizou o programa pela primeira vez é usado como tempo inicial da contagem. Assim, considerando um programa A anterior em outro canal, cujo horário de término é depois do horário de início de um programa B, e assim se o telespectador desejasse assistir o programa B mesmo pela metade, então o tempo total considerado pelo 2S ACT deste programa para aquele usuário será da metade até o final somente. Se o telespectador assistir essa metade inteira, a nota será máxima. Caso o telespectador comece a assistir e depois troque de canal logo no início da sua visualização, o sistema automaticamente atribui uma nota baixa a esse programa.

Esta abordagem é similar à adotada em outro projeto de recomendação para IPTV baseado em conteúdo. KIM et al. (2011) utilizam uma base de dados real de telespectadores coreanos capturada entre 2002 e 2003. O perfil é criado baseado no histórico da base de dados e também separado em clusters usando algoritmos de agrupamento de dados.

4.5. ReS

Sabendo quem são as pessoas presentes em frente ao televisor num dado momento, possuindo o perfil dessas pessoas, e conhecendo todos os programas que estão disponíveis

na televisão neste mesmo momento, já se tem todas as informações necessárias para gerar as recomendações de quais programas esses telespectadores terão maior probabilidade de gostar. Assim, o sistema que recebe todas essas informações e executa o algoritmo de seleção é o *Recommender Subsystem* (ReS).

O ReS utiliza um sistema híbrido de recomendação de programas, tendo uma parte colaborativa e uma parte baseada em conteúdo. O vetor perfil de preferências por classificação do usuário é usado na parte baseada em conteúdo, buscando programas disponíveis que encaixem no perfil da pessoa. A parte colaborativa utiliza a informação de qual grupo de similaridade de telespectadores pertence o perfil analisado e seleciona programas já vistos por pessoas deste mesmo grupo para recomendar.

Para cada perfil recebido pelo ReS, uma lista de recomendação é montada em ordem de maior nota para menor nota. Ao final, todas as recomendações são enviadas ao PMS que irá fazer a mescla das recomendações.

4.6. Netflix, Movielens, IMDb e Allmovie

Um sistema de recomendação, tanto baseado em conteúdo quanto colaborativo, necessita de um volume de dados inicial para realizar as recomendações. Um sistema baseado em conteúdo precisa de um histórico do telespectador para poder criar o seu perfil. E no caso do sistema colaborativo o histórico muito maior é necessário, e de diversas pessoas também.

Como não foi encontrada nenhuma base de históricos de programas de TV assistidos por uma grande população de pessoas disponível publicamente, o 2S ACT teve seus parâmetros configurados para operar em bases de filmes, de forma que as dimensões do perfil do usuário foram relacionadas com os diversos gêneros de filmes existentes.

Os sistemas de recomendação para filmes são baseados em uma nota atribuída pelo usuário a cada filme assistido, e essa nota é a base do sistema colaborativo. Um sistema de recomendação de TV não pode usar um recurso de avaliação ativa como este, já que a duração dos programas é em média menor e não seria prático ter que avaliar cada programa assistido. Assim, o conceito de nota no sistema de recomendação televisivo pode ser deduzido baseado no tempo assistido de um programa em relação ao seu total. Se por exemplo o usuário assiste 100% do programa, a nota será máxima. Caso o telespectador

assista somente metade do programa e depois ele muda o canal, a avaliação será metade, e a nota é mínima se a pessoa não assiste ou assiste muito pouco do programa. Abordagem similar é adotada em outros sistemas de recomendação para TV, e outras ferramentas podem ser usadas para obter uma maior precisão no cálculo desta nota. TAKAMA e MUTO (2009) utilizaram, além do tempo assistido de um programa, um sistema de reconhecimento de voz para detectar as emoções e comentários do telespectador sobre este programa para avaliá-lo positivamente ou negativamente.

Entre as duas bases de dados de filmes disponíveis, Netflix e Movielens, cada uma possui uma vantagem. A base da Movielens até 100 mil notas já tem suas informações cruzadas com o IMDb e informações socioeconômicas dos usuários, como faixa de idade, sexo, etc. A base de 10 milhões não possui as informações de socioeconômicas, mas possui 100 mil palavras chaves criadas pelos usuários vinculadas ao filme. Já a base da Netflix é muito maior, possuindo 100 milhões de notas criadas por 400 mil usuários reais que são assinantes da Netflix. Porém, esta a base não está relacionada com o IMDb e nem com nenhuma outra base de filmes, necessitando assim buscar as informações de gênero dos filmes de alguma outra maneira.

Como o foco desta pesquisa é testar o algoritmo para grandes volumes, o critério é selecionar a maior quantidade de dados possível. Desta forma, a Netflix foi escolhido para ser a base que vai popular os dados iniciais do sistema, com o ônus de ser necessário fazer o cruzamento entre os registros da Netflix e do IMDb. A base do IMDb é disponível para *download* (IMDB, 2011), e existem algumas ferramentas que permitem reproduzir a base do IMDb localmente, para depois realizar pesquisas por título, ano, etc, tornando esta tarefa mais fácil. Dois artigos descrevem um procedimento similar para participar do prêmio Netflix, calculando perfis dos usuários da Netflix baseado na classificação dos filmes no IMDb (GOEL; BATRA, 2008; BHATIA; PATNAIK, 2008).

4.7. Forma de Análise dos Resultados

O ponto inicial para comparar o desempenho dos algoritmos do 2S ACT foi a criação de um conjunto de dados para ser usado pelos algoritmos. Assim, para medir o tempo de execução do algoritmo de agrupamento, foram criados arquivos dados artificiais usando a função de geração de conjuntos de dados *gen_gaussian_datafile()* oferecida no código do k-

Means, desenvolvido por PELLEG e MOORE (2000). Entre eles, dois foram escolhidos, um contendo 600 mil pontos e outro com 30 mil pontos, todos os pontos com 60 dimensões e divididos em 30 clusters gaussianos. Esses *datasets* serão referenciados como 600k60d e 30k60d respectivamente. A quantidade de pontos foi escolhida de forma a serem aproximados ao número de usuários da base de dados de histórico e a dimensão de forma a aproximar com o número de classificações de programas de TV existentes no DVB. Um conjunto de dados real com 28 mil pontos de duas dimensões gerados por uma ferramenta de depuração também foi usado nos testes. Os testes seguiram o plano de primeiro desenvolver o código sequencial, e depois criar as tarefas iniciais do StarSs. A seguir, foi feita a adição das instruções SIMD e a criação das tarefas para o resto do algoritmo. Os resultados entre cada melhora foram comparados com a etapa anterior. Depois do código finalizado, foi feita a execução do algoritmo em diversos ambientes diferentes, como supercomputadores e servidores.

Já quanto ao sistema de recomendação, o objetivo não é desenvolver o sistema de recomendação mais preciso, mas um que possua uma precisão aceitável, que seja escalável e que realize recomendações para grupos. O Cinematch, aplicativo de recomendação da Netflix, foi usado como referência, pois seu funcionamento também é focado na escalabilidade. Assim, espera-se obter recomendações com uma precisão similar aos resultados divulgados do Cinematch na competição promovida pela Netflix. Quanto à escalabilidade, a curva de ganho por número de processador foi avaliada para constatar se o seu ganho cresce linearmente com o número de processadores sem muito *overhead*.

O código foi escrito em C, com exceção do STBM que foi feito em Java SE 1.6, e compilado com o CellSs versão 2.2, e o SMPsS usado é de versão 2.1 usando o compilador Intel ICC versão 11. O NestedSs e o GPUSs por ainda estarem em desenvolvimento não possuem versão oficial, assim como o Nanox, que é a versão nova do SMPsS. O nível de confiabilidade do algoritmo de clusterização usado foi de $\alpha = 0,00001$. Os ambientes Cell BE usados foram um Playstation 3 de segunda geração e um IBM BladeCenter QS22 com dois processadores PowerXCell, rodando um Linux de kernel versão 2.6.16-128.el5 SMP, chamado aqui de Maricel. Para comparação, também foi usado um Intel P4 3.0GHz HyperThread para executar o código sequencial. Para comparar com ambientes SMP, dois diferentes supercomputadores foram usados. Um deles foi um IBM BladeCenter JS22 com quatro núcleos POWER6 e um Linux versão 2.6.29.6-217.2.8.fc11.ppc64, presente junto com

os *blades* Cell BE do Maricel. O outro foi uma máquina SGI Altix 4700 com 128 processadores Dual Core Montecito (IA-64), rodando um Linux de kernel 2.6.16.60-0.42.5-default. Finalmente, para os testes e medições em ambiente GPU foi utilizado um AMD Dual-Core Opteron 2222 com duas placas Nvidia G280 com um Linux de kernel 2.6.39-2-amd64, apelidado de Obelix.

4.8. Conclusão

Como explicado, o sistema foi dividido com módulos, de forma a isolar os componentes um dos outros. Sendo o primeiro módulo, o STBM, responsável pela comunicação com os dispositivos Bluetooth. O PMS é o módulo de integração que faz toda a comunicação entre os outros módulos, que consulta o STBM para saber quem está presente e busca os perfis destas pessoas no UBA, módulo que possui o banco de dados de perfil. Com esses perfis, o PMS se comunica com o módulo que gera as recomendações, o ReS. Ao final, o PMS apresenta essas recomendações ou as envia para o STBM para ser transmitida para algum telespectador específico.

5. IMPLEMENTAÇÃO

Esta seção é dedicada a descrever como foi feita a implementação do sistema. As camadas do sistema foram desenvolvidas uma a uma de forma independente, e depois unidas para formar o 2S ACT. Cada parte funciona de maneira isolada, e pode tanto ser executada em um hardware embarcado, quanto ser executada em um servidor dedicado caso seja necessário maior escalabilidade.

5.1. STBM

Para validar a real funcionalidade do Bluetooth como tecnologia de RFID, foi desenvolvido um aplicativo de busca e identificação de dispositivos Bluetooth, implementado o princípio do BtReader e batizado de BtReaderApp. O ambiente de teste da aplicação é um PC com processador x86-32 com um adaptador USB, ou *dongle*, Bluetooth classe 2 conectado e sistema operacional Windows.

Para implementar o BtReaderApp foram usadas algumas bibliotecas do Java, entre elas a JSR 82 (APIs Java para Bluetooth). O JSR 82 usa o *Generic Connection Framework* (GCF), que é uma plataforma criada para suportar qualquer tipo de conexão. O GCF foi definido dentro do *Connected Limited Device Configuration* (CLDC) 1.0 e é uma configuração comum do Java ME para telefones celulares. Como aqui o desenvolvimento seria feito em uma plataforma PC, o Java ME não é a máquina virtual mais adequada. Então, em vez de usar o CLDC para definir o GCF, foi usado o *Connected Device Configuration* (CDC) que é definido no Java SE através de uma API opcional chamada JSR 197 – *Generic Connection Framework Optional Package for Java SE Platform*. Assim, tendo o GCF dentro do Java SE, foi feita a busca de uma biblioteca que programasse o JSR 82.

Dentre as bibliotecas pesquisadas, duas são as mais comuns, sendo uma para Windows/Linux e outra somente para Linux:

- Bluecove – Ela é compatível com algumas pilhas Bluetooth do Windows, como o WIDCOMM, BlueSoleil e *Microsoft Bluetooth Stack* (BLUECOVE, 2008). O BlueCove teve o início de seu desenvolvimento financiado pela Intel e agora é um grupo independente. Esta biblioteca também possui suporte Linux usando a pilha BlueZ, permitindo a portabilidade do sistema.

- Avetana – É uma versão *open source* da biblioteca que é executada no Linux também usado a pilha BlueZ (AVETANA, 2008).

Como o ambiente de testes tinha sistema operacional Windows e o STBM seria executado também em um ambiente Linux, então a biblioteca que implementa o JSR 82 escolhida foi a Bluecove. A pilha Bluetooth usada foi a WIDCOMM, que vinha em um CD junto com o *dongle* Bluetooth usado. A máquina virtual Java usada é de versão 1.6 e a IDE de desenvolvimento foi o Eclipse.

O BtReaderApp então foi codificado, usando os métodos da classe *DiscoveryAgent* do Bluecove. O aplicativo foi projetado para buscar por dispositivos Bluetooth; quando um é encontrado, o aplicativo apresenta uma mensagem de “*New BtTag Arrived*” e a seguir mostra o *Bluetooth Address* e o nome do dispositivo encontrado. A seguir, uma nova busca é feita, e caso o dispositivo encontrado anteriormente não seja encontrado de novo, uma mensagem de “*BtTag Left*”, também com o *Bluetooth Address* e o nome ao lado. O fluxograma do BtReaderApp pode ser visto na Figura 8.

Figura 8 - Diagrama de fluxo do STBM

Como BtTag foram usados dois aparelhos celulares, um N95 da Nokia e um Blackberry 7100i da RIM (NOKIA, 2008; BLACKBERRY, 2008). Ambos tiveram suas configurações de Bluetooth alteradas para serem encontrados, e tiveram seus nomes alterados para MobileN95 e Mobile7100i respectivamente. Na Figura 9 está mostrado algum *screenshot* das mensagens geradas pela aproximação e afastamento dos BtTags dentro da área de cobertura do BtReader.

```
BT Reader Started!
08/09/2009 18:04:21:609 New BtTag Arrived! BtAddress = 00 1C 9A 2D 49 A2 (MobileN95)
08/09/2009 18:04:36:796 Device Inquiry completed!
08/09/2009 18:04:48:718 Device Inquiry completed!
08/09/2009 18:05:02:515 New BtTag Arrived! BtAddress = 00 0F 86 AE 89 AF (Mobile7100i)
08/09/2009 18:05:02:609 Device Inquiry completed!
08/09/2009 18:05:14:531 Device Inquiry completed!
08/09/2009 18:05:26:453 Device Inquiry completed!
08/09/2009 18:05:26:453 BtTag left! BtAddress = 00 0F 86 AE 89 AF (Mobile7100i)
08/09/2009 18:05:38:375 Device Inquiry completed!
08/09/2009 18:05:50:296 Device Inquiry completed!
08/09/2009 18:06:02:218 Device Inquiry completed!
08/09/2009 18:06:14:250 Device Inquiry completed!
08/09/2009 18:06:14:250 BtTag left! BtAddress = 00 1C 9A 2D 49 A2 (MobileN95)
08/09/2009 18:06:26:281 Device Inquiry completed!
BT Reader Stopped!
```

Figura 9 - Logs de entrada e saída dos dispositivos da área de cobertura

A seguir, foram feitos testes de atraso para a detecção de um dispositivo Bluetooth e medições de alcance. A rotina de busca de dispositivos Bluetooth é configurada de fábrica para buscar dispositivos na área de alcance por 10 segundos. Nesse intervalo de tempo, os BtTags podem ser encontrados imediatamente no início da busca, ou em alguns casos somente no final. O gráfico da Figura 10 mostra a distribuição do tempo necessário para se encontrar os dispositivos em 50 ensaios de busca.

Como visto no gráfico, existem duas áreas de concentrações onde na maioria das vezes os dispositivos são encontrados. Uma dessas áreas está entre T+0s e T+1s (1 segundo do início da varredura), representando 59% das buscas, e outra entre T+2,5s e T+3,5s, representando 38% das buscas. Estudando a especificação do Bluetooth foi visto que o tempo de varredura padrão é de 2,56 segundos (BLUETOOTH, 2007). Assim, durante a varredura, 4 buscas são feitas, uma a cada 2,56 segundos. Considerando cada uma das buscas independentemente, 97% dos encontros acontecem no primeiro segundo. Assim, é possível afirmar que o tempo de detecção de um dispositivo Bluetooth neste cenário é de um segundo. Se por acaso alguma interferência impedir a detecção de algum dispositivo na 1ª varredura, ainda existe a possibilidade de descoberta nas varreduras seguintes.

Outro ensaio feito foi a medição do alcance de detecção dos BtTags. O teste foi dividido em duas partes, primeiro para medir o melhor caso, e depois medir casos reais. Considerando o melhor caso, seria o celular com Bluetooth colocado sem nenhuma obstrução entre ele e o *dongle*. Já no caso real os testes foram feitos com o telefone

colocado dentro de bolsas ou em bolsos de calças vestidas. Tanto o rádio Bluetooth dos celulares quanto do *dongle* são classe 2, onde o alcance máximo teórico é de 10 metros. Como não se tem o diagrama de ganho das antenas dos celulares e nem do *dongle*, diversos ensaios foram feitos em várias orientações. O maior alcance encontrado foi de 19 metros para o melhor caso, visto na Figura 11 junto com os resultados de caso real.

Como pode ser visto no gráfico, a comparação entre a distância máxima obtida no caso real e as distâncias práticas difere por pouco. Foi notado que com o telefone no bolso, a distância máxima reduz seu valor em quase 2 metros. Isso acontece pela proximidade do celular com o corpo humano, que possui muita água, que é um material que absorve as ondas eletromagnéticas.

Assim, neste cenário, o ambiente possui baixa influência no alcance final da comunicação. Desta forma, o Bluetooth pode ser usado como dispositivo de identificação para aplicações que necessitam até 15 metros de alcance, com ou sem obstáculos, e sem a necessidade de ambientes controlados para alinhamento do dispositivo Bluetooth, tendo como desvantagem o tempo de detecção que pode tardar até 10 segundos no pior caso.

Figura 10 - Gráfico dos testes de demora na detecção dos celulares

Figura 11 - Gráfico dos alcances obtidos

5.2. UBA

O UBA contém dentro de sua arquitetura a base de dados alimentada com os históricos de visualização de todos os programas assistidos por cada telespectador. Junto com essa base e suas rotinas de pesquisa de usuários, o UBA possui também um algoritmo de clusterização para detectar os grupos de telespectadores, a qual grupo cada um deles pertence e os perfis médios de cada grupo.

5.2.1. Carga Inicial da Base de Dados

Para popular a base do UBA com informações de histórico reais, foi usada a base de dados do *Netflix Prize*. Porém, a relação de filmes existentes nesta base não está relacionada com o gênero, informação crítica para criar o perfil dos telespectadores. Assim, o primeiro passo para criar a base de dados foi relacionar os 17.700 filmes da base da Netflix com a base de filmes do IMDb. A ferramenta IMDbPY feita em Python permite navegar pela base do IMDb e realizar consultas de informações sobre os filmes (IMDBPY, 2010). Usando essa ferramenta foi criado um *script* em Python para acessar a tabela de filmes da Netflix, que pode ser visto no apêndice A. O script realiza a busca na base do IMDb pelo filme que possua o título igual, usando a função *search_movie(title)*. Esta função retorna uma lista dos filmes que possuem similaridade com o nome buscado, em ordem de relevância.

Os nomes existentes na base da Netflix não são necessariamente os nomes originais dos filmes, portanto a função de buscar filmes pode retornar um resultado que não tenha relação com o filme buscado. Assim, para aumentar a garantia de que as informações utilizadas são realmente do filme buscado, uma segunda verificação foi inserida, que é a comparação das datas de lançamento da base da Netflix e do IMDb. Caso as datas de lançamento dos filmes registradas da base da Netflix e as registradas na base do IMDb não sejam as mesmas, este filme é descartado. Como as datas de lançamento cadastradas na base da Netflix também não são exatamente as mesmas da base do IMDb, nem todos os filmes da base da Netflix foram correlacionados. Do total de 17.700 filmes existentes na base da Netflix, 7.500 foram pareados tanto em nome quando em data de lançamento com a base do IMDb.

5.2.2. k-Means

Entre os algoritmos de clusterização, existe um algoritmo iterativo muito popular chamado k-Means (MACQUEEN, 1967). Dado um banco de dados de X pontos de d dimensões, o k-Means divide os pontos em k clusters, onde cada ponto pertence ao cluster cujo centro possui a menor distância média. O algoritmo inicia escolhendo randomicamente o ponto inicial dos centros, e assim atribui cada ponto x pertencente a conjunto X ao centro c_j , sendo c_j pertencente ao grupo de centros C e j sendo um valor entre 1 e k . A seguir, é recalculado cada centro c_j de acordo com a média dos pontos que pertencem ao cluster j , e assim outra iteração do algoritmo é executada. Apesar do k-Means ser muito usado em aplicações de *data mining*, ele possui suas desvantagens. Primeiramente, é necessário saber o valor de k , ou seja, o número de clusters que existem na sua base de dados. Segundo, o tempo de execução do algoritmo é muito grande para grandes volumes de dados.

Para resolver o primeiro problema, algumas modificações no k-Means foram desenvolvidas pela comunidade científica, sendo uma delas o G-Means (HAMERLY; ELKAN, 2003). Este algoritmo, depois de executar o k-Means e encontrar os centros, realiza um teste em cada cluster para ver a distribuição dos pontos segue uma distribuição Gaussiana. Para os clusters em que a distribuição não for Gaussiana, o G-Means divide o seu centro em dois centros novos. Após verificar todos os *clusters*, se algum dos centros for dividido, o k-Means é executado novamente com o novo valor de k . O G-Means repete esse ciclo até que todos os centros possuam uma distribuição Gaussiana de acordo com certa confiabilidade α . O algoritmo pode ser visto no Algoritmo 1.

Algoritmo 1 G-Means(X, α)

- 1: Sendo C o conjunto de dois centros c_1 e c_2 aleatoriamente escolhidos dentro de X .
 - 2: $C \leftarrow \text{k-Means}(C, X)$
 - 3: Sendo X_j o conjunto de pontos associados ao centro c_j .
 - 4: Faça um teste estatístico para cada X_j para verificar se a sua distribuição é Gaussiana (a um nível de confiança α).
 - 5: Se X_j passa no teste, mantenha c_j . Caso contrário, substitua c_j por dois novos centros, atualizando C .
 - 6: Volte ao passo 2 com o novo valor de C até que nenhum novo centro seja adicionado.
-

O teste estatístico usado pelo G-means para verificar a distribuição dos clusters contém o teste de normalidade de Anderson-Darling (A&D). Considerando um cluster de pontos X_j , sendo seus pontos atribuídos a um centro c_j , o teste consiste no algoritmo descrito em Algoritmo 2.

Algoritmo 2 Anderson-Darling(X_j, α)

- 1: Sendo C o conjunto de dois centros c_1 e c_2 aleatoriamente escolhidos dentro de X_j .
 - 2: $C \leftarrow k\text{-Means}(C, X_j)$
 - 3: Seja $v = c_1 - c_2$ o vetor de dimensão d conectando esses dois centros.
 - 4: Projete cada ponto de X_j em v , sendo X_j' o vetor de uma dimensão contendo o resultado dessa projeção.
 - 5: Normalize X_j' para deixá-lo com média 0 e variância 1.
 - 6: Faça o teste de normalidade de Anderson-Darling. Se $A_2(X_j')$ está dentro do alcance de valores não críticos para o nível de confiança α , mantenha c_j , e descarte c_1 e c_2 . Caso contrário, substitua c_j por c_1 e c_2 .
-

Em resumo, o G-Means possui 6 fases de execução. A primeira é o *grande k-means*, onde é executado o k-Means para todos os pontos da base de dados X e com o número inicial de centros k . Ao final dessa fase, cada ponto x pertencente a X vai estar atribuído a um dos centros c_j pertencente a C . Neste ponto, a base de dados vai ter seus dados reorganizados para agrupar na memória os pontos que pertencem a cada centro. Assim, a segunda fase é chamada de *agrupamento de dados*. As fases seguintes são executadas para cada conjunto de pontos X_j pertencentes a cada cluster em separado. A terceira fase é o primeiro passo do teste estatístico, que é o *k-means de dois centros*, onde outra execução do k-Means é feita usando como dado somente o conjunto de pontos X_j e tendo dois centros arbitrários. No final dessa execução inicia a fase seguinte, onde vetor v é calculado e os pontos projetados neste vetor. Assim, a quarta fase se chama *projeção*. Nesta hora, a média e o desvio padrão são calculados simultaneamente com a projeção, usando a fórmula do desvio padrão simplificado. Depois da projeção vem a fase da *ordenação*, em que o vetor de projeções é ordenado em ordem ascendente. Finalmente, é executada a fase final, o teste A&D. Com o vetor de projeções ordenado, é calculada a função de distribuição cumulativa normal padrão. No fim da execução, se todos os clusters passaram no teste de normalidade A&D, a

execução do G-Means é terminada e os resultados, como o número de clusters k encontrados, seus respectivos centros C e a qual cluster pertence cada ponto são apresentados. Caso contrário, é feita outra iteração das 6 fases com o novo número de centros k . O fluxograma pode ser visto na Figura 12.

Figura 12 - Diagrama de fluxo da implementação do G-means. CV é o valor crítico calculado usando o nível de confiabilidade α , k é o número de clusters, C é o conjunto de centros dos clusters, X é o conjunto de dados e X_j é o conjunto de pontos que pertencem ao centro c_j .

Já sobre o segundo problema do k-Means mencionado, desde que esse problema é intrínseco ao algoritmo, existem duas abordagens para melhorar a sua performance: 1) Modificar o algoritmo para ser executado em uma arquitetura computacional diferente, como processadores SIMD ou GPUs; 2) Modificar a execução para explorar o paralelismo de processadores de múltiplos núcleo ou ambientes multiprocessados como clusters e supercomputadores.

Com essas duas abordagens em mente, o G-Means foi modificado para ser executado no processador Cell BE para explorar tanto os recursos de SIMD quanto a execução paralela dos seus 8 SPUs, usando o StarSs. A implementação foi dividida em alguns passos, e cada passo teve seu resultado comparado com o passo anterior. O primeiro passo para comparar os resultados é a definição de uma versão do código de referência para comparar o ganho de performance obtido com cada nova modificação. Desta forma, foi definido que o tempo de referência é o tempo necessário para executar a implementação do G-Means usando o código paralelo para o Cell BE do k-Means descrita por BUEHRER e PARTHASARATHY (2007). Porém, para realizar essa medição, é necessário primeiro implementar um código sequencial do k-Means, para depois iniciar a codificação do G-Means.

A implementação do k-Means foi feita seguindo a estrutura do algoritmo de Buehrer&Parthasarathy para suportar as instruções SIMD e a paralelização, com a diferença de que o desenvolvimento foi feito utilizando a plataforma do StarSs para gerenciar a criação das tarefas. Como a arquitetura inicial escolhida para os testes foi o Cell BE, então o compilador CellSs foi usado. Após a implementação do k-Means sequencial e testes com uma base de dados conhecidas com cinco mil pontos e 2 dimensões, as melhorias iniciaram. A primeira melhoria no código do k-Means foi a modificação na função de cálculo de distância para usar as instruções SIMD do SPU. A fórmula usada é uma simplificação da distância euclidiana, visto em (1).

$$d_1(p, q) = \sum_{i=1}^n |p_i - q_i|^2 \quad . \quad (1)$$

Esta fórmula de distância foi usada, pois o k-Means precisa saber somente qual dos centros é o mais próximo, sendo que o valor da distância calculada em si não importa. Assim, o cálculo da raiz-quadrada feita no cálculo da distância euclidiana padrão não é necessário. Com isso, esta fórmula foi modificada para usar as chamadas SIMD do Cell BE *spu_sub()* e *spu_madd()*, visto no Algoritmo 3.

Algoritmo 3 Parte do código da tarefa mostrando a implementação do algoritmo de cálculo de distância usando instruções SIMD.

```
vector float acc = {0,0,0,0};

for (i = 0; i < dimensions/4+1; i++) {
    result = spu_sub(recVec[i], centerVec[i]);
    acc = spu_madd(result,result,acc);
}
distance = spu_extract(acc, 0) + spu_extract(acc, 1) + spu_extract(acc, 2) +
           spu_extract(acc, 3);
```

A segunda modificação feita foi na organização do dado em partes consecutivas de memória para introduzir uma independência de memória entre a execução das funções, possibilitando assim a definição das funções como tarefas do CellSs e permitindo os SPUs trabalharem com seus dados em paralelo. Com isso, foi possível definir as tarefas CellSs com as #pragmas. Desta forma também, o código do k-Means pode ser compilado pelo CellSs e executado em uma arquitetura Cell BE utilizando os SPUs.

Desta forma, as melhorias feitas no k-Means para o Cell BE reportadas por outros autores foram implementadas e verificadas usando a plataforma CellSs. A implementação do k-Means básico foi finalizada, um código principal foi feito englobando as chamadas ao k-Means para que o G-Means pudesse ser implementado de acordo com o trabalho de Harmerly&Elkan. O código principal chama o k-Means em duas ocasiões: A primeira na fase do *grande k-means*, onde o k-Means é executado para todos os pontos pertencentes ao banco de dados X; e a segunda, na fase do *k-means de dois centros*, onde os pontos de cada cluster são submetidos ao k-Means, como parte do teste A&D para verificar se o centro vai ser dividido ou não. Na 2ª chamada, os outros cálculos do teste devem ser implementados para verificar a distribuição e decidir se o centro será ou não dividido. Essas implementações foram feitas seguindo ainda o trabalho de Hamerly&Elkan, e finalizando assim a primeira versão funcional do código. Com isso, o G-Means está usando uma versão melhorada do k-Means que utiliza os SPUs e suas chamadas SIMD dentro de tarefas do CellSs, e o restante do código utiliza execução sequencial para realizar o restante das operações. Essa versão do código é usada no capítulo de resultados para medir o tempo de execução do G-Means quando comparado com as melhorias feitas no código.

Para definir a estratégia de otimização, o algoritmo foi analisado de forma a encontrar partes que possam ser modificadas para serem executadas em paralelo e/ou usar instruções SIMD. Das 6 etapas do G-Means, a etapa do *grande k-means* e o *k-means de dois centros* já

estão modificados para execução paralela e usar instruções SIMD. Sendo assim, restam 4 etapas para serem melhoradas: a de *agrupamento da memória, projeção, ordenação* e o *teste de A&D*. As etapas de *agrupamento de memória* e a *ordenação* não foram foco de melhorias para usar instruções SIMD por não envolverem cálculos computacionais, só movimentação de dados na memória. Com isso, essas etapas somente obtiveram ganho de escalabilidade em ambientes SMP.

5.2.2.1. Primeira melhora: A projeção no vetor de resultados

A projeção foi modificada para utilizar as instruções SIMD e dividir o cálculo entre os diversos SPUs. A fórmula da projeção está apresentada em (2), e consiste no produto escalar entre cada um dos pontos x e o vetor de referência v encontrado, dividido pelo módulo de v .

$$|C| = \frac{P \bullet V}{\|V\|^2} . \quad (2)$$

Primeiro, o vetor de d dimensões v é calculado usando o resultado do *k-means de dois centros*. Com o vetor v , seu módulo $|v|$ é calculado. Assim, o conjunto de pontos é dividido em pedaços, ou *chunks*, com o mesmo tamanho usado pelo k-Means. Depois disso um *chunk*, o vetor v e o valor de $|v|$ são enviados para o SPU via transferência DMA para calcular a projeção de cada ponto do *chunk*. Ao mesmo tempo, o SPU acumula o valor da projeção e a sua potência de dois em duas outras variáveis, para ajudar no cálculo da média e do desvio padrão na hora do teste de A&D. Baseado na fórmula da projeção, a implementação da tarefa do SPU foi criada de acordo com o Algoritmo 4.

Algoritmo 4 Código da tarefa de cálculo da projeção do SPU

```
#pragma css task input(dimension, powv, chunk) \
                input(recVec, v) \
                inout(projVec, meanSum, stdSum)

void projab_task (int dimension, float powv, int chunk, float *recVec, float *v,
float *projVec, float *meanSum, float *stdSum) {

    for (j = 0; j < chunk; j++) {
        vector float res = {0,0,0,0};
        for (i = 0; i < dimension / 4; i++) {
            res = spu_madd(recVec[i], v[i], result);
        }
        float pdotv = spu_extract(res,0) + spu_extract(res,1) +
            spu_extract(res,2) + spu_extract(res,3);
```

```

    float c = pdotv / powv;
    projVec[j] = c;
    *meanSum += c;
    *stdSum += c * c;
}
}

```

5.2.2.2. Segunda melhora: Teste de Anderson-Darling

A próxima melhora foi no cálculo do teste de Anderson-Darling. Tendo o vetor de projeções ordenado X_j' , a fórmula (3) é aplicada a todos os valores, um a um.

$$S = \sum_{k=1}^n \frac{2k-1}{n} [\ln F(Y_k) + \ln(1 - F(Y_{n+1-k}))], \quad (3)$$

onde $F(x)$ é a função normal cumulativa distribuída padrão e os valores de Y são os pontos do vetor de projeção normalizado. Após calcular o valor de S , (4) é aplicada.

$$A^2 = -n - S. \quad (4)$$

A estratégia para paralelizar esse cálculo foi a criação de duas tarefas, uma chamada de $ADF()$, que normaliza o ponto do vetor X_j' e calcula todos os valores de $F(x)$ em paralelo. Assim, esta tarefa cria um vetor F do mesmo tamanho n com os parâmetros a serem usados por outra tarefa. Esta outra tarefa foi chamada de $ADS()$, que foi criada para calcular em paralelo cada um dos elementos da somatória de S , acumulando o seu valor em uma variável, visto no Algoritmo 5.

Algoritmo 5 Tarefas de cálculo do teste de Anderson&Darling

```

#pragma omp taskwait beforeinput(x[n], n, mean, std) \
               inout(fli[n], f2i[n])

void adf_task(float *x, int n, float mean, float std, float *fli, float *f2i) {
    for (i = 0; i < n; i++) {
        float res = normcdf((x[i] - mean)/std);
        fli[i] = res;
        f2i[n-i-1] = (1 - res);
    }
}

```

```

#pragma cxx task input(n, offset, ntot) \
    input(fli[n], f2i[n]) \
    inout(s)

void ads_task(int n, int offset, int ntot, float *fli, float *f2i, float *s) {
    float si = 0;
    for (i = 0; i < n; i++) {
        si += (2 * (I + offset + 1) - 1) * (log(fli[i]) + log(f2i[i])) / ntot;
    }
    *s += si;
}

```

Como o CellSs gerencia a dependência de dados entre as tarefas, quando um *chunk* do vetor F necessário para as tarefas $ADS()$ estiver disponível, esta última inicia sua execução sem a necessidade de nenhuma sincronização adicional. Depois que todas as tarefas forem executadas para todos os *chunks*, os valores parciais de S são somados e o valor final de A^2 é calculado.

5.2.2.3. Terceira melhora: Configuração do CellSs e Afinamento dos Parâmetros

A terceira e última melhora no código está relacionada com o afinamento dos parâmetros do G-Means e da biblioteca de tempo de execução do CellSs. Para cada tarefa é definido um tamanho de pedaço, ou *chunk*, que é ser enviado para os SPUs. Se o tamanho for muito grande, o SPU vai ficar muito tempo aguardando pelo fim da transferência DMA, aumentando o tempo desperdiçado em espera do processador. Em contraste, se o valor é muito pequeno, a sobrecarga gerada pelo CellSs e pelo custo para iniciar uma transferência DMA reduzem a performance do algoritmo. Assim, os parâmetros de tamanho de *chunk* foram analisados para cada tarefa com a finalidade de encontrar o melhor valor para cada caso. Como o k-Means e a projeção são dependentes da dimensão do dado, o tamanho do *chunk* usado no código de referência foi calculado usando o algoritmo *GetChunksize()* descrito no trabalho de Buehrer&Parthasarathy. Para as tarefas que são executadas depois da projeção, como todos os dados terão somente uma dimensão, o tamanho de *chunk* inicial foi definido como 1024 registros por SPUs. Este valor foi escolhido, pois o banco de dados a ser processado possui 20 mil pontos por cluster, o que criaria 20 tarefas por cluster para ser enviada para os 16 SPUs.

Similarmente, o CellSs tem seus próprios parâmetros de execução em tempo real. Por exemplo, um deles define quantas tarefas vão ser enviadas para cada SPU ao mesmo tempo. Dois parâmetros específicos foram estudados, o *min_task* e o *max_strand_size*. O *min_task*

especifica o mínimo necessário de tarefas prontas para execução antes de iniciar o agendamento das tarefas para os SPUs. O parâmetro *max_strand_size* define qual o máximo de tarefas agrupadas que serão enviadas para cada SPU simultaneamente. Esses parâmetros também foram analisados para encontrar a melhor configuração que minimiza o tempo de execução do G-Means.

5.2.2.4. Resultados de Performance

Para iniciar a análise dos resultados do G-Means no processador Cell BE, uma função de medição foi adicionada no código referência para medir o tempo gasto em cada parte do código. A execução do G-Means usando um código de 600 mil pontos de 60 dimensões, chamada aqui de 600k60d, teve seus resultados apresentados na Tabela 3.

Tabela 3 – Tempo de execução do G-Means para cada parte do algoritmo.

Fases do G-Means	Tempo	Percentual
Grande k-Means	2.00s	7%
Agrupamento de dados	3.35s	11%
k-Means de dois centros	8.06s	27%
Cálculo da Projeção	3.42s	12%
Ordenação do Vetor	3.79s	13%
Teste de Anderson-Darling	8.69s	30%
Total	29.31s	100%

Como visto na Tabela 3, a *projeção* e o cálculo do *teste A&D* são responsáveis por 42% do tempo de execução. Como o k-Means já está otimizado e as fases de ordenação e agrupamento de dados só demonstram ganhos em ambientes SMP, as fases de projeção e A&D são as únicas que podem ser melhoradas.

A execução da versão referência do G-Means com o banco de dados 600k60d usando somente um SPU resultou num tempo de execução de 78,35 segundos. Essa execução e o seu tempo vão ser referenciados aqui como *execução base* e *tempo base*. O melhor tempo de execução alcançado foi de 29,12 segundos quando foram usados 16 SPUs, e esse tempo vai ser endereçado aqui como *tempo base paralelo*. Este tempo aponta uma melhora de 2,7 vezes em relação à execução com somente 1 SPU. Após a melhora na fase de projeção, o melhor tempo caiu para 26,81 segundos com o mesmo número de SPUs, resultando numa melhora total de 2,9 vezes quando comparado com o tempo base e 8% mais rápido se comparado com a execução paralela. O pequeno ganho já era esperado, pois a *projeção* é responsável somente por 12% do tempo total de execução.

Agora, quanto à segunda melhoria, o cálculo do A&D, os resultados foram mais expressivos. O melhor tempo alcançado foi de 19,27 segundos com todos os SPUs, representando uma melhora de 4,0 vezes quando comparado com a execução base. Até agora, as modificações no código reduziram o tempo de execução em 34% em relação ao *tempo base paralelo*.

Para finalizar, todas as variáveis relativas ao tamanho do *chunk* e do compilador foram afinadas para encontrar os melhores valores. Sobre o tamanho dos *chunks* do k-Means e da projeção, o valor ótimo de 4kB descrito no trabalho de Buehrer&Parthasarathy não se aplica, pois o CellSs gerencia todas as transferências e possui recursos de *double buffer* implementado (transferir enquanto processa). Assim, o tempo de execução foi medido usando tamanhos de *chunk* que variam entre 4kB, 8kB, 16kB e 32kB. Para valores maiores que 32kB, a execução retorna erro por causa da memória limitada do SPU. Os resultados podem ser vistos na Tabela 4.

Tabela 4 – Tempo de execução do G-Means para tamanhos de *chunk* diferentes usando o conjunto de dados 600k60d.

Tamanho do Chunk (bytes)	4k	8k	16k	32k
Tempo (s)	40.04s	23.29s	18.12s	19.71s

Conforme os dados organizados na Tabela 4, o menor tempo de execução, avaliado em 18,12 segundos corresponde a um tamanho de *chunk* medido de 16kB. Como o CellSs gerencia as transferências, o limite máximo de 16kB para transferências DMA no Cell BE não se aplica. Porém, para valores maiores que essa restrição são necessárias duas transferências DMA, o que significa duas vezes o custo para inicializá-las.

Para os parâmetros do cálculo do A&D, como este não depende da dimensão do dado e pode ter valores diferentes para cada tarefa, o parâmetro é medido em números de registros em vez de *kilobytes*. Os intervalos de valores testados foram de 64, 128, 256, 512, 1024, 2048 e 4096 para ambas as tarefas. Os melhores valores encontrados para a tarefa de cálculo dos valores de $F(x)$ e de S foram, respectivamente, 256 e 512, apesar de que valores entre 256 e 1024 também apresentaram resultados próximos. Quando o número de registros fica menor que 256, o tempo de execução cresce significativamente devido ao grande número de pequenas tarefas, evidenciando a sobrecarga do CellSs para gerar e gerenciar as tarefas. Não obstante, quando o tamanho do *chunk* ultrapassa os 1024 registros o sistema começa a perder granularidade, criando poucas tarefas que demoram

muito. Isso cria um desbalanceamento na distribuição de tarefas e assim deixando processadores desocupados.

Após os parâmetros do código serem afinados, os dois parâmetros de execução do CellSs, o *min_task* e o *max_strand_size*, foram analisados. Os valores padrão para cada um desses parâmetros são 32 e 8 respectivamente. Como as tarefas foram criadas e afinadas para explorar os limites de memória do SPU, cada tarefa individual ocupa quase todos os recursos disponíveis no SPU. Como consequência, quando ambos os parâmetros *max_strand_size* e *min_task* têm seus valores ajustados para um, o G-Means apresenta o melhor tempo de execução. Ou seja, as tarefas começam a ser executadas imediatamente, e elas são transmitidas uma por uma.

Com todos os parâmetros afinados, o melhor tempo de execução do G-Means diminuiu para 12,51 segundos usando 16 SPUs. Esse tempo representa um ganho de 6,3 vezes em relação à *execução base*, e 57% mais rápido que o *tempo base paralelo*.

Para concluir esta parte, a medição do tempo gasto em cada fase feito na Tabela 3 foi feita novamente, porém com os tempos gastos pela versão melhorada do G-Means. Esses tempos estão apresentados na Tabela 5.

Comparando ambas as tabelas, facilmente pode ser vista a redução da participação das duas fases que foram melhoradas no tempo final do G-Means. A fase de projeção teve seu tempo reduzido de 3,42 segundos para 0,5 segundos (6,8x) e o cálculo do A&D foi reduzido de 8,69 segundos para 0,33 segundos (26,3x). Esses ganhos foram conseguidos devido à paralelização da execução entre os 16 SPUs, uso de instruções SIMD e o melhor agendamento das tarefas feitos pelo CellSs depois da afinação dos parâmetros. A Figura 13 mostra os ganhos de cada implementação, variando o número de SPUs e comparando com a *execução base*.

Tabela 5 – Tempo de execução do G-Means para cada parte do algoritmo melhorado.

Fases do G-Means	Tempo	Percentual
Grande k-Means	1,16s	9%
Agrupamento de dados	3.34s	27%
k-Means de dois centros	3,38s	27%
Cálculo da Projeção	0,50s	4%
Ordenação do Vetor	3.80s	30%
Teste de Anderson-Darling	0,33s	3%
Total	12,52s	100%

Figura 13 - Gráfico comparando quatro diferentes execuções do G-Means. A linha de baixo (quadrado) representa o ganho do código contendo somente as melhoras do k-Means. A segunda linha (triângulo) apresenta o ganho do G-Means com as melhoras na função de projeção, enquanto a seguinte linha (círculo) mostra o ganho do código contendo o a melhora do cálculo do Anderson&Darling. A última linha (estrela) representa a execução do G-Means com o código final após a afinação dos parâmetros de execução e de tamanho de *chunk*.

5.2.2.5. Resultados Comparativos com o StarSs

O G-Means foi executado em um *blade computer* QS22 usando duas diferentes ferramentas da família StarSs para o Cell BE, sendo eles o CellSs e o NestedSs. O NestedSs suporta dois níveis hierárquicos de tarefas e pode criar tarefas para serem executadas tanto no CPU quando no SPU. Assim, foi criada outra tarefa de CPU que engloba todas as etapas do teste estatístico, como o *k-means de dois centros*, a *projeção*, a *ordenação* e o *teste de A&D*. Isso permite ao NestedSs realizar o teste de A&D simultaneamente em diversos *clusters* de pontos, cada CPU calculando um grupo. Para executar o G-Means em um ambiente SMP usando o SMPsSs, as tarefas SPU foram trocadas por tarefas CPU, permitindo assim utilizar os diversos processadores de um cluster. Com o NestedSs e o SMPsSs, as etapas de *agrupamento de dados* e *ordenação* exploram o paralelismo da arquitetura para obter ganho de performance.

O tempo de execução do G-Means em uma estação de trabalho Intel P4 HT foi usado como tempo de referência, e todos os ganhos mencionados são relativos a esse tempo. A Figura 15 e a Figura 15 mostram a comparação de ganhos entre as diferentes arquiteturas usando dois bancos de dados, o 600k60d e um menor com apenas 30 mil pontos de 60 dimensões, chamado de 30k60d.

O ganho em todos os ambientes, exceto os que possuem Cell BE, é praticamente constante para ambos os arquivos de dados, com uma pequena variação de $\pm 0,8x$. Porém, os ganhos da execução do G-Means usando o CellSs e o NestedSs possuem comportamento diferenciado. O ganho do PS3, do Maricel usando o CellSs com 8 e 16 SPU's e usando o NestedSs para o *dataset* de 30k60 foram 5,50x, 6,00x, 6,60x e 6,95x respectivamente. Agora, quando comparado com a execução do G-Means com a base de dados de 600k60d, os ganhos cresceram para 6,99x, 8.14x, 9.17x e 11.18x respectivamente, colocando o Cell BE como a melhor opção entre os outros ambientes com o mesmo número de processadores.

A diferença entre o conjunto de dados de 600k60d e de 30k60d sendo executados pelo CellSs e pelo NestedSs é causada pela diferença entre a relação dados/computação. Como o SPU possui uma memória muito limitada e somente aceita transferências DMA, o custo para realizar essas transferências é muito alto para as pequenas computações do 30k60d. O mesmo fenômeno não ocorre no SMPs, já que o ambiente possui uma arquitetura de memória compartilhada.

Figura 14 - Gráfico comparativo entre diversos ambientes computacionais de um processador, mostrando o ganho do algoritmo usando como referência o tempo de execução em um Intel P4.

Figura 15 - Gráfico comparativo entre diversos ambientes computacionais multi-processado mostrando o ganho do algoritmo usando como referência o tempo de execução em um Intel P4 da figura anterior.

5.2.2.6. Resultados com Dados Reais

Como todos os conjuntos de dados usados nos testes foram volumes de dados sintetizados, foi feito um teste do G-Means utilizando dados gerados por uma aplicação real. Um pequeno *dataset* de 28 mil pontos de duas dimensões, chamado aqui de 28k2d, foi gerado por uma ferramenta de análise de dados de performance para supercomputação. Essa ferramenta de análise de arquivos de depuração utiliza o algoritmo de clusterização DBScan para encontrar os *clusters* na massa de dados (GONZALEZ; GIMENEZ; LABARTA, 2009). Os autores tentaram utilizar o X-Means, mas sem sucesso devido à distribuição irregular dos dados. O X-Means tende a dividir os clusters com formato elíptico em diversos pequenos clusters circulares. O G-Means foi executado para esse conjunto de dados e encontrou um resultado de número *clusters* igual ao do DBScan, como visto na Figura 16, sendo necessária somente uma modificação no valor crítico do teste de A&D para aceitar distribuições mais distantes de uma distribuição com forma gaussiana perfeita. Uma diferença em relação aos resultados do DBScan foi o fato que este algoritmo marca alguns pontos muito distantes de qualquer *cluster* como ruído, enquanto o G-Means vincula-os a algum cluster próximo ou cria um *cluster* somente para esses pontos.

O tempo de execução do DBScan para esse conjunto de dados foi de 18 minutos e 46 segundos em um *desktop* convencional Intel P4 HT, enquanto que o G-Means foi executado em 0,41 segundos usando apenas um processador Cell BE, o que significa um ganho de 2.746x.

Figura 16 - Representação gráfica de um arquivo de depuração da aplicação GAPGEOfEM com os 13 clusters encontrados representado por símbolos e cores diferentes.

5.2.2.7. Testes Usando GPU

Devido à tendência de que os processadores atuais, como o nVidia Tegra, comecem a vir com elementos de placas gráficas (GPU) na mesma pastilha de silício que o CPU, a viabilidade de portar tarefas para a arquitetura GPU foi analisada.

Uma característica de se desenvolver algoritmos para GPUs é que sua arquitetura foca em executar centenas de *threads* de forma paralela, porém de forma mais lenta, em vez de executar um *thread* rapidamente, como é feito pelos CPUs. Assim, apenas algoritmos que possam ser tratados como milhares de *threads* independentes conseguem demonstrar resultados satisfatórios nesta arquitetura.

Desta forma, as fases do G-Means que poderiam explorar um ganho de performance significativo são a do *grande k-means*, *k-means de dois centros* e de *projeção*, pois cada

cálculo de distância entre os ponto e os centros pode ser feito por um *thread*, da mesma forma como cada projeção também. Porém, devido a uma limitação da versão atual do GPUs, a tarefa do *k-means de dois centros* não pode ser desenvolvida, somente as outras duas.

Assim, a implementação da tarefa do *grande k-means* e da *projeção* foram feitas para a arquitetura GPU. Uma vantagem de se usar o StarSs como ferramenta de paralelização é a possibilidade de se utilizar simultaneamente vários CPUs, junto com vários GPUs e aceleradores ao mesmo tempo, com uma relativa facilidade de programação e sem preocupar com o gerenciamento da memória e do uso de cada recurso. Desta forma, a tarefa do *grande k-means* foi implementada de forma a utilizar os CPUs e os GPUs disponíveis no servidor Obelix. Para isso, foi necessário encontrar a melhor relação Tarefas CPU vs Tarefas GPU de forma a manter todos os recursos com o menor tempo de ociosidade possível.

A tarefa normal do CPU para calcular o centro mais próximo de um *chunk* de pontos e centros, bem como para atualizar o histograma leva em média 28 milissegundos (ms). Já o mesmo cálculo para a arquitetura GPU foi dividido em duas tarefas, uma que seria o cálculo da distância, que é executada na GPU. A atualização do histograma foi separada em outra tarefa, e ainda é feito no CPU. O cálculo de histograma exige exigiria o uso da memória compartilhada do GPU e estaria restrita a quantidade deste tipo de memória disponível na GPU, assim, este cálculo foi mantido no CPU. A GPU para calcular as distâncias para o mesmo *chunk* necessita somente 4,0 ms, sendo 2,6 ms para execução e outros 1,4 ms para as transferências de memória entre o GPU e o *host*. Já a tarefa de atualizar os histogramas demora 1,7 ms. Assim, o tempo total da abordagem para GPUs leva 5,7 ms, sendo 4 ms executados na GPU e 1,7 ms executados no CPU. Esse tempo é 4,9x mais rápido que o tempo de um CPU individual.

Após alguns testes com os parâmetros e a medição do tempo final de execução, foi constatado que a melhor relação entre as tarefas do CPU e as do GPU é de 14 por um, ou seja, com tempo necessário para executar uma tarefa de CPU gerada, 14 tarefas de GPU são executadas.

Como a arquitetura do GPU favorece significativamente o processamento das tarefas do *grande k-means* e da *projeção*, o tempo de execução foi reduzido em 72%, indo de aproximadamente 40 segundos para aproximadamente 11 segundos, que significa um

speedup de 3,56x com o uso de duas placas GPU. A comparação entre as linhas de tempo geradas pela ferramenta Paraver do StarSs pode ser vista na Figura 17, e uma aproximação da linha de tempo pode ser vista na Figura 18.

Figura 17 – Comparação na mesma escala de tempo da execução do G-Means usando os 2 CPUs e usando dois CPUs com dois GPUs. As duas primeiras linhas representam o uso do CPU e as duas últimas representam o uso do GPU. As cores, com exceção do azul claro, representam as diferentes tarefas em execução. O azul claro marca o estado de ociosidade do processador.

Figura 18 - (Acima) *Timeline* da execução usando GPUs; (Abaixo) Zoom nas tarefas, sendo as vermelhas tarefas de CPU; as rosas são as tarefas do GPU; e as transferências entre a memória RAM e a memória dos GPUs são ilustradas nas cores azul e branca.

5.2.3. Geração e Clusterização dos Usuários da Netflix

Com a base de dados já filtrada e pareada com a base do IMDb e com o algoritmo de clusterização já implementado, a tarefa seguinte é a geração dos perfis dos telespectadores e a sua respectiva clusterização.

A metodologia adotada pelo 2S ACT para a geração do perfil foi baseada na percentagem de tempo assistido de todos os gêneros em relação ao total. Como foi usada uma base de filmes para o teste do sistema, foi definido que todos os filmes possuem igual duração, assim

o cálculo foi reduzido para a porcentagem da quantidade de filmes assistidos em relação ao todo. Ou seja, primeiro é analisado todo o histórico do telespectador para contar quantos programas de cada gênero ele assistiu. Sendo G o vetor de contadores de n gêneros, e g_x o contador de programas de um dado gênero x , com $0 < x < n$, o perfil do telespectador é definido pela divisão do vetor G pelo número total de programas n_{tot} assistidos por uma pessoa. O vetor resultante representa a participação de cada gênero no perfil da pessoa, sendo que gêneros com maior frequência no histórico terão o g_x maior do que os outros gêneros.

$$G = [g_1, g_2, g_3, \dots g_x, \dots g_{n-1}, g_n]$$

$$G\% = G / n_{tot}$$

Após o cálculo dos 478.865 perfis existentes na base da Netflix, foi feita a sua classificação por cluster. A função da clusterização é retornar os grupos de perfil com gostos similares, reduzindo assim o espaço amostral da base de dados para a recomendação baseada em conteúdo. Com essa menor quantidade de dados para enviar ao motor de recomendação, as previsões são calculadas mais rapidamente e assim é possível calcular mais recomendações sem consumir todo o processamento.

Para encontrar os clusters desta base de perfis da Netflix relacionados com o IMDb, foi utilizado o G-Means descrito anteriormente. Como as distribuições dos perfis nas 27 dimensões não são perfeitamente gaussianas, uma etapa de *tunning* dos parâmetros do teste de A&D teve que ser realizada.

O valor crítico do A&D define o quão perfeito uma distribuição tem que ser para considerá-la gaussiana. Se uma distribuição não for perfeitamente gaussiana e o valor crítico for muito baixo, o algoritmo vai dividir clusters que apesar de não serem perfeitamente gaussianos, formam clusters reais. Este tipo de divisão pode gerar um loop no algoritmo onde ele dividirá um único cluster muitas vezes, pois se o cluster real já não se parecia uma distribuição gaussiana perfeita, a distribuição resultante dessa divisão certamente não será uma distribuição deste tipo. Esse acontecimento pode ser detectado quando o número de *clusters* cresce rapidamente a cada execução, e a solução desse problema é o aumento do valor crítico. Porém, se o valor crítico foi muito alto, o G-Means pode acabar considerando dois clusters muito próximos como uma distribuição gaussiana imperfeita, e assim não os dividindo. Como não é possível saber quantos clusters e nem o tipo de distribuição que existe no banco de dados de perfis, o valor crítico do teste de A&D foi variado em um

intervalo delimitado a fim de descobrir uma convergência do número de clusters encontrado, e de forma a manter este parâmetro o menor possível. No algoritmo, o número máximo de clusters foi limitado em 100 para garantir uma condição de término para caso o algoritmo comece a dividir os clusters indefinidamente. Como visto na Figura 19, o melhor valor crítico encontrado foi de 100, pois antes deste valor o número de *cluster* varia muito com a variação do valor crítico. Entre 100 e 140 o número de *clusters* estabiliza em 56, e a partir de 140 o número de clusters volta a decair. O tempo de execução para essa clusterização foi de 16,92s usando somente um processador Cell BE com 8 SPUs.

Figura 19. Gráfico mostrado a variação do número de clusters encontrados em relação ao valor crítico.

No fim da execução, o G-Means classificou os 478 mil perfis em 54 *clusters*, tendo assim uma média de oito mil perfis por *cluster*. Sete gêneros foram omitidos dos gráficos por não possuírem nenhum filme na base de dados filtrada da Netflix. Os gêneros omitidos são: Biografia, Film-Noir, Game-Show, Notícias, Reality-TV, Talk-Show. Cada cluster possui um ou mais gêneros mais relevantes em relação ao perfil médio, visto na Figura 20, e reciprocamente, um ou mais gêneros menos relevantes. Os gêneros mais relevantes serão chamados de gêneros fortes, enquanto que os menos relevantes serão chamados de gêneros fracos.

Para evidenciar os gêneros fortes e fracos, o vetor de perfil de cada cluster é subtraído do vetor do perfil médio. Assim, valores positivos indicam gêneros fortes e valores negativos correspondem a gêneros fracos. Para exemplificar, a Figura 21, Figura 22 e Figura 23 apresentam alguns dos *clusters* encontrados.

Figura 20 – Perfil médio gerado pela média dos perfis existentes na base de dados.

Figura 21 – Cluster número um onde o gênero forte é terror, seguido de suspense e ficção científica, e os gêneros fracos são comédia, drama e romance.

Figura 22 – Cluster número seis, com drama e romance sendo gêneros fortes, e ação, terror e ficção científica com gêneros fracos.

Figura 23 – Cluster número oito, mostrando uma situação provável de família com filho pequeno, onde os gêneros fortes são família, animação e fantasia. Os gêneros fracos são drama, comédia, ação, suspense e romance.

5.3. PMS e ReS

O módulo responsável por interligar todos os outros módulos é o PMS. Este módulo recebe um vetor do STBM com os identificadores de cada telefone Bluetooth no alcance. Com essa lista, o PMS solicita ao UBA os perfis dos respectivos telespectadores, montando uma tabela *hash* em que a chave é o identificador do telefone e o valor é o seu respectivo perfil.

A cada telespectador presente na sala, o PMS busca o seu perfil na base de dados e o disponibiliza ao ReS para que sejam feitas as recomendações. Esses dois módulos foram implementados juntos, já que o PMS é basicamente o sistema responsável pela gerência de toda a comunicação interna do sistema.

Apesar do ReS possuir um motor de recomendação híbrido, os resultados das recomendações foram comparados com um dos algoritmos baseado em conteúdo desenvolvidos no prêmio Netflix, em que a documentação completa foi disponibilizada. Este algoritmo foi escolhido por ter ganhado o prêmio anual da Netflix em 2008, forçando os autores a publicarem os métodos e resultados no fórum (BELL; KOREN; VOLINSKY, 2008). Este algoritmo venceu os algoritmos híbridos e baseados em conteúdo que participaram da competição na época.

Para cada perfil desta tabela, o PMS consulta o ReS para calcular as recomendações deste telespectador. Ao final, o ReS retorna para o PMS as notas de cada programa disponível na programação para cada um dos perfis da tabela *hash*. Com essa lista de recomendações de cada perfil disponível, o PMS faz a análise e a mescla de cada recomendação.

5.3.1. Tarefas e Algoritmo

A implementação do PME do ReS foi feita para suportar escalabilidade, de forma a utilizar os muitos processadores disponíveis no ambiente. O ReS necessita acesso à base de dados de forma paralela, por múltiplos *threads* executados pelos diversos processadores. Assim, a melhor abordagem para conseguir ganho de performance foi embutir um

gerenciador de banco de dados para cada *thread*. Por isso então, o SQLite foi escolhido para ser utilizado como gerenciador de dados (SQLITE, 2011).

O SQLite é um gerenciado de base de dados *open-source* que foi desenvolvido para permitir que aplicações tenham uma rápida integração a um banco de dados sem necessitar de muitos recursos computacionais. Este banco utiliza o padrão de consultas SQL e, graças a essa característica de baixo custo computacional, é comumente utilizado para aplicações embarcadas e aplicativos de celular. A base de dados, junto com os índices, as tabelas de uso interno e demais configurações estão todas salvas em um único arquivo. No caso da base do 2S ACT, contendo os históricos da Netflix mais a relação de perfis de todos os usuários, depois de exportada do MySQL e importada pelo SQLite com todos os índices, gerou um arquivo de base de dados de 8Gb. O procedimento de importação de dados no SQLite está descrito no anexo A.

Unido ao SQLite, o StarSs foi utilizado pelo ReS para gerar, agendar e executar as tarefas de geração das recomendações de forma a utilizar ao máximo todos os recursos computacionais disponíveis no ambiente. Como o StarSs não permite variáveis de acesso global, cada tarefa do StarSs instancia um gerenciador independente do SQLite. Assim, a função de conexão com banco foi implementada para usar o parâmetro *SQLITE_OPEN_FULLMUTEX*, que permite a conexão de múltiplos *threads* ao banco.

Cada tarefa do ReS é responsável por gerar as recomendações de um usuário. Cada tarefa criada recebe o ID do perfil do telespectador na base de dados, e com esse ID a tarefa busca todos os programas disponíveis para gerar as recomendações. Caso exista um número de programas maior que um parâmetro *movie_max_chunk*, esta tarefa se subdivide em duas recursivamente para utilizar dois processadores até que a quantidade de recomendações a ser calculada seja menor que o parâmetro. A parte baseada em conteúdo recomenda programas com base puramente no perfil do telespectador, e a parte colaborativa busca por programas assistidos por outras pessoas que fazem parte seu cluster de perfil. O diagrama de funcionamento do ReS pode ser visto na Figura 24.

Figura 24- Diagrama de funcionamento do PMS junto com o ReS

5.3.2. Comparação BellKor com o ReS Colaborativo

O sistema de recomendação colaborativo utiliza a similaridade de históricos e notas entre diferentes pessoas para identificar telespectadores que possuam mesmo gosto. Baseado neste grupo de afinidade, o motor de recomendação verifica as notas atribuídas pelos outros usuários na base de dados de históricos para prever qual nota seria atribuída pelo telespectador em questão a um dado programa.

No desafio da Netflix, a forma de avaliar a qualidade das recomendações dos algoritmos se deu por meio da comparação da raiz do erro médio quadrático, ou RMSE - *Root Mean-Square Error*. O RMSE é uma medida de diferença entre os valores das previsões e os valores reais. O cálculo do RMSE é feito de acordo com (5). Quanto menor o RMSE, mais próximo do valor real é o valor da previsão.

$$RMSE(D_{real}, D_{prev}) = \sqrt{\frac{\sum_{i=1}^n (d_{real,i} - d_{prev,i})^2}{n}}, \quad (5)$$

onde D_{real} é a matriz com as notas reais dadas pelos telespectadores; D_{prev} são as previsões do algoritmo; e n é o número de notas na matriz.

Entre os participantes da Netflix diversas abordagens foram tentadas e compartilhadas no fórum de discussão. Alguns grupos conseguiram grande destaque nas discussões por estarem frequentemente entre os cinco melhores algoritmos. Entre eles estão o Bellkor, BigChaos e o Pragmatic Theory, sendo que o algoritmo vencedor foi a mescla do algoritmo destas 3 equipes. Durante os estudos da equipe Bellkor em sua participação no Netflix Prize, diversas análises no conteúdo da base de dados foram feitas para reduzir o erro das previsões (BELL; KOREN; VOLINSKY, 2008). Os resultados da parte colaborativa do 2S ACT foram comparados com os resultados encontrados seguindo a metodologia da equipe Bellkor aplicada à base de dados filtrada que foi relacionada ao IMDb.

Uma das filtrações estudadas por Bellkor foi a remoção dos fatores globais. Alguns usuários são mais bondosos quando dão notas para os filmes, se eles gostaram ou não do filme, eles dão uma nota um pouco maior que a média das outras pessoas. Outros usuários são mais rígidos, dando notas menores que a média. Ambas as tendências são chamadas fatores globais, e no caso deste exemplo, o fator global é chamado efeito de usuário, ou *user_effect*. Alguns filmes recebem notas mais altas quando eles são lançados do que quando ele já está há alguns anos no mercado. Este fator seria o fator de filme por tempo. Os dois fatores que serão utilizados pelo ReS são o efeito do usuário e o efeito dos filmes, ou *movie_effect*.

O *movie_effect* descreve como as notas se comportam quando variamos os filmes. O seu cálculo é simples e consiste no calculo da média de notas dada em cada filme. Já o *user_effect* mostra a variação das notas de acordo com cada usuário. O cálculo desse efeito é baseado na média de todas as notas de cada telespectador da base.

O processo de retirar os fatores globais das recomendações consiste em calcular o efeito e retirá-lo das notas, um a um, repassando o residual para o próximo efeito. As fórmulas de cálculo do *movie_effect* e do *user_effect* são descritas a seguir.

Inicialmente é retirada a média global para encontrar o primeiro residual, visto em (6):

$$r_{ui}(1) = r_{ui} - avg, \quad (6)$$

onde r_{ui} são as notas e avg é a média global das notas.

A seguir, é calculado o fator θ de cada filme usando (7) e (8).

$$\hat{\theta}_i = \frac{\sum r_{ui}(1)}{n_i}. \quad (7)$$

$$\theta_i = \frac{n_i \hat{\theta}_i}{n_i + \alpha_{movie}}, \quad (8)$$

onde θ_i é o fator de cada filme, n_i é o numero de filmes e o α_{movie} é uma constante obtida experimentalmente.

Em seguida, o *movie_effect* é retirado do primeiro residual para calcular o segundo residual usando (9).

$$r_{ui}(2) = r_{ui}(1) - \theta_i. \quad (9)$$

Finalizado a remoção do *movie_effect*, é feito o cálculo do residual sem o *user_effect*. Os passos para retirar o fator do usuário são muito parecidos com a remoção do fator dos filmes, e são ilustrados em (10) e (11).

$$\hat{\theta}_u = \frac{\sum r_{ui}(2)}{n_u}. \quad (10)$$

$$\theta_u = \frac{n_u \hat{\theta}_u}{n_u + \alpha_{user}}, \quad (11)$$

onde θ_u é o fator de cada filme, n_u é o numero de filmes e o α_{user} é uma constante também obtida experimentalmente.

Com isso, o residual final pode ser calculado utilizando (12).

$$r_{ui}(3) = r_{ui}(2) - \theta_u. \quad (12)$$

Utilizando essas fórmulas para reproduzir os valores encontrados do RMSE, o primeiro passo para comparar os resultados pelo método de Bellkor com os obtidos pelo ReS é repetir os passos feitos pela equipe nessa amostra da base da Netflix. Em seu trabalho, Bellkor utilizou os valores de α_{movie} e α_{user} iguais a 25 e 7, respectivamente. Esses valores serão usados também pelo 2S ACT na comparação dos resultados. O procedimento de Bellkor foi o seguinte:

1. Calcular a média geral de todas as notas da base.
2. Atualizar as notas a serem previstas com o valor da média.
3. Calcular o *movie_effect*.
4. Retirar o *movie_effect* das notas a serem previstas.
5. Calcular o *user_effect*.
6. Retirar o *user_effect* das notas a serem previstas.

Os passos para esse cálculo foram feitos na base de dados, conforme apêndice B. A nota média geral calculada na amostra da base da Netflix foi de 3,5685. Definindo todas as previsões para esse valor gera um RMSE de 1,1248. Retirando o *movie_effect*, o RMSE cai para 1,0595 e ao final, sem o *user_effect*, o RMSE fica reduzido a 0,9667, o que é 1,5% acima do valor encontrado pelo Cinematch (RMSE de 0,9525).

Agora, o ReS baseia o seu cálculo na média das notas das pessoas do mesmo *cluster*, em vez de usar média global. Assim, para um dado filme, a previsão é a média das notas atribuídas a este filme pelos telespectadores do mesmo cluster que o ranquearam. Após esta média, somente o fator *user_effect* é removido, já que com esse cálculo, o *movie_effect* é removido automaticamente. Assim, os RMSEs de Bellkor e do ReS podem ser comparados.

O RMSE medido após a execução do ReS para todas as previsões da base de prova foi de 1,0254, enquanto que Bellkor obteve um RMSE de 1,0595 após a remoção do *movie_effect*. Retirando o *user_effect* da base, o RMSE final encontrado foi de 0,9521, 1,5% menor que o encontrado por Bellkor (0,9667) e 0,1% melhor que o RMSE do Cinematch (0,9525).

Sobre a performance e escalabilidade do algoritmo, para analisar o seu tempo de execução foi adotada a unidade de eficiência de previsões por segundo. Na Figura 25 pode ser visto um gráfico de previsões por segundo por número de processadores de um Altix 4700. O ambiente Altix foi escolhido por permitir a medição da escalabilidade para muitos processadores. O uso dos recursos de SIMD das arquiteturas heterogêneas e da arquitetura das GPUs no algoritmo do ReS é feito diretamente pelas bibliotecas do SQLite, e não foi explorado neste momento na pesquisa por ter seu desenvolvimento ainda em progresso por outros grupos de pesquisa (BAKKUM; SKADRON, 2010).

Figura 25 - Gráfico de previsões por segundo pelo número de processadores.

O ReS apresenta uma escalabilidade linear, com um *speedup* médio de 1,875x para cada vez que o número de processadores é dobrado. Ou seja, comparando o ganho de um processador para 64 processadores, o ganho ideal seria de 64x, porém o ganho real é de 43,45x. Utilizando um núcleo somente, o algoritmo é capaz de calcular 2,7 recomendações por segundo, e o máximo de 117,3 recomendações por segundo foi atingido utilizando os 64 processadores disponíveis, considerando a base histórica de 56 milhões de notas e 478 mil telespectadores.

Assim, supondo um ambiente doméstico em que numa TV a cabo existam 200 canais e haja 4 telespectadores presentes na sala de TV, seriam necessários 800 cálculos de recomendação a cada hora, estimando-se que um programa de TV tenha uma duração média de uma hora. Considerando que o STB tenha um processador de 4 núcleos, o 2S ACT levaria um minuto e meio para calcular todas recomendações para todas as pessoas na mesma base de dados de 56 milhões de registros. Se for considerada uma base restrita somente aos telespectadores que frequentam a residência, a base seria significativamente menor, tendo algo em torno de 5000 registros e um banco de 50 perfis. Essa menor base reduziria significativamente o tempo para calcular as notas.

5.3.3. Envio das Recomendações

Ao final, as recomendações calculadas pelo ReS são devolvidas para o PMS, que faz o agrupamento das notas. As notas que estão no grupo das recomendações “ótimas”, “boas” e em alguns casos “normais” são enviadas para os telespectadores. Esse envio é feito através do protocolo Bluetooth de forma espontânea. O padrão de comunicação para envio dessas recomendações para os celulares é o *Object Exchange* (OBEX). Este protocolo foi criado pelo grupo que padroniza a comunicação de dados via infravermelho, porém posteriormente, esse protocolo foi incorporado ao Bluetooth. O OBEX permite a troca de informações entre dispositivos Bluetooth sem a necessidade de pareamento, utilizando o perfil *Object Push Profile* (OPP).

O alcance de envio das recomendações é similar ao alcance medido para a detecção do celular, usada para identificação do telespectador. Com isso, todos os celulares detectados pelo 2S ACT podem receber essas recomendações.

5.4. Conclusão

Neste capítulo foram mostrados os passos do desenvolvimento de cada um dos componentes do sistema. O STBM foi feito utilizando a biblioteca Bluecove, e os resultados obtidos mostraram a viabilidade de se utilizar um telefone celular como meio de identificação de pessoas. Já o UBA concentrou boa parte do esforço desta pesquisa no desenvolvimento de um algoritmo de clusterização para encontrar os grupos de similaridades dos perfis, de forma a aproveitar todos os recursos computacionais disponíveis pelos processadores heterogêneos de múltiplos núcleos. No desenvolvimento do ReS e do PMS, outro algoritmo escalável foi desenvolvido também para obter aumento do número de recomendações por segundo, com o aumento do número de núcleos. Ao final, o sistema como um todo obteve um RMSE 0,1% melhor que o Cinematch, e com uma escalabilidade de ganho de 1,83 vezes a cada vez que é dobrado o número de processadores. Esses resultados estão de acordo com os objetivos da pesquisa, mostrando a viabilidade de se utilizar conjuntamente aparelhos celulares para identificar pessoas e um STB com um processador heterogêneo para calcular recomendações para grupos de forma mais eficiente.

6. CONCLUSÃO E PERSPECTIVAS DE TRABALHOS FUTUROS

O projeto do 2S ACT foi iniciado efetivamente em 2007, e, após os estudos e implementações preliminares, deu origem a dois pedidos de patentes. Uma delas é o 2S ACT em si, descrevendo seus módulos e comunicação interna (FOINA; RAMIREZ-FERNANDEZ, 2008a). A segunda é um complemento que descreve o uso do 2S ACT para publicidade direcionada em ambientes públicos (FOINA; RAMIREZ-FERNANDEZ, 2008b). Prontamente, após o depósito das patentes, a publicação de um resumo estendido no *PhD Forum* do PerCom 2009 e do artigo completo no Eurocon 2009 foram aceitos, o que serviu para expor a concepção do projeto pelo meio acadêmico (FOINA, 2009a; FOINA; RAMIREZ-FERNANDEZ, 2009b).

Os resultados aqui apresentados comprovam a usabilidade do Bluetooth para a identificação dos telespectadores, com tempos de detecção e alcance adequados para a aplicação. O alcance de 20 metros e o tempo médio de 3 segundos necessários para encontrar o celular são suficientes para detectar as chegadas e saídas das pessoas da sala de televisão. Informações auxiliares dos rádios Bluetooth, como o RSSI, podem até informar quando a pessoa se afasta ou se aproxima do televisor, sendo um desenvolvimento futuro. Os estudos envolvendo RFID ativo e suas funcionalidades de localização geraram duas pesquisas paralelas durante este doutorado, com publicações em conferências internacionais, revista e um capítulo de livro. O artigo chamado de PlayerTracker descreve um sistema de localização de jogadores de futebol em campo usando triangulação de antenas de RFID, e foi publicado no congresso PerCom 2010 (FOINA et al., 2010a). Outro projeto é o chamado Hunter, que descreve um sistema de controle de processo de descarga de produto a granel utilizando como base *tags* de RFID ativos. A forma inovadora de detecção de entrada e saída através do sinal transmitido pelo *tag* foi publicada no *International Microwave and Optoelectronics Conference 2007* (FOINA; BARBIN; RAMIREZ, 2007). Já a descrição do sistema completo foi publicada na revista do IEEE *Latin America Transactions* (FOINA; RAMIREZ-FERNANDEZ; BARBIN, 2009c), e os resultados obtidos com a automatização do processo de descarga geraram um capítulo do livro *Contemporary Robotics - Challenges and Solutions* (FOINA; RAMIREZ-FERNANDEZ, 2009d).

Quanto aos processadores de arquitetura Cell BE e CPU+GPU escolhidos, os resultados apontam que essas arquiteturas oferecem um real ganho de performance entre os ambientes com o mesmo número de núcleos homogêneos. O algoritmo de clusterização desenvolvido para o 2S ACT demonstra um ganho mínimo de cinco vezes entre o processador Cell BE e um processador P4 HT lançado na mesma época. Já comparando o Cell BE com processadores de supercomputação, o primeiro atingiu um ganho de 11,18x enquanto que o Altix com dois núcleos Itanium 9030 alcançou 9,03x, comprovando um ganho real de 23%. Já agregando as placas GPU, uma redução de 72% no tempo de execução durante os cálculos do k-Means foi demonstrada. Os primeiros resultados obtidos do G-Means paralelo com o processador Cell BE foram publicados em um congresso acadêmico chamado *Facing Multi-core Challenge*, focado para pesquisadores de doutorado (FOINA; BADIA; RAMIREZ-FERNANDEZ, 2011a). Já os resultados completos do G-Means usando arquitetura Cell BE, SMP e GPU, e estrutura de tarefas com dois níveis foram publicados no *International Conference on High Performance Computing & Simulation 2011* (FOINA et al, 2011b). Os primeiros resultados do 2S ACT obtidos, como a criação de perfis usando o processador Cell BE, foram publicados no *IEEE/IFIP Network Operations and Management Symposium 2010* (FOINA; RAMIREZ-FERNANDEZ; BADIA, 2010b).

O ReS ainda carece de bibliotecas estáveis que explorem as GPUs, apesar de resultados prévios comprovarem ganhos de 20 a 70 vezes em consultas de bases de dados muito grandes (BAKKUM; SKADRON, 2010). Porém, este módulo demonstrou uma relação de ganho de performance de 1,84x a cada vez que o número de processadores dobra, em um ensaio que chega até a 64 processadores. Com isso, o ReS permite um maior número de cálculos de recomendação por segundo, encontrando programas desejáveis em um menor intervalo de tempo. Como o uso das GPGPUs para computação intensiva iniciou recentemente em 2009, pouca coisa fora do universo de algoritmos matemáticos, físicos e de aplicações muito específicas como simulações físicas, explora o seu uso. Assim, este sistema demonstra outra utilidade para este recurso computacional cada vez mais disponível nos dispositivos de eletrônica embarcada, que agora também têm GPUs embarcadas projetadas especificamente para aceleração gráfica e descompactação de vídeo.

Quanto à precisão e a qualidade das recomendações geradas pelo ReS, o RMSE final das recomendações da tabela de amostra atinge um resultado 0,1% melhor que o Cinematch e 1,5% melhor que o de Bellkor, retirando somente os dois fatores globais, o efeito filme e o

efeito do usuário. Como o 2S ACT é projetado para calcular recomendações em tempo real, e todas as técnicas utilizadas pelos primeiros lugares do prêmio Netflix são uma mistura de centenas de algoritmos diferentes, seu uso para recomendação em tempo real em computadores embarcados é inviável. Por isso, a referência usada é o RMSE do Cinematch, que possui uso similar ao 2S ACT. Assim o resultado do RMSE é satisfatório, dado que o foco da pesquisa não é avançar significativamente na qualidade da recomendação, mas sim desenvolver um algoritmo em tempo real para dispositivos embarcados que realize recomendações para grupos no menor tempo possível.

Desde o início da pesquisa, as tendências tecnológicas foram mudando rapidamente. Em 2007, data de início da investigação, os grupos de pesquisa em computação paralela convergiam para o processador Cell BE, por sua arquitetura diferenciada e pela capacidade de computação disponível. Porém, em 2009 o Cell BE se tornou obsoleto e a nova tendência era utilizar o conjunto CPU+GPU. Neste ponto, o uso da ferramenta de agendamento de tarefas StarSs facilitou significativamente o desenvolvimento dos algoritmos paralelos. Como essa ferramenta consiste em uma família de compiladores e bibliotecas para os mais diversos ambientes, foi relativamente fácil e rápido portar parte dos códigos otimizados para o Cell BE para o ambiente CPU+GPU. Desta forma, o 2S ACT pode ser executado no Altix 4700 usando 64 processadores, e no Obelix com suas duas GPUs Nvidia G280, simplesmente alterando o compilador e as bibliotecas conectadas, junto com pequenas alterações no código fonte.

Um dos pontos não desenvolvido nesta pesquisa foi a integração do 2S ACT com o sistema de TV digital brasileiro. Isso ocorreu devido aos atrasos na finalização da especificação do Ginga-J, o middleware do SBTVD, que foi disponibilizada na sua versão estável somente em 2010, enquanto que a previsão inicial de finalização era em 2007 (ABNT, 2010a; ABNT, 2010b). O sistema RePTVD dos autores LUCA e ZORZO (2009) teve o mesmo problema com os prazos do Ginga-J, e como solução implementaram seu sistema utilizando um emulador de *Xplet* do padrão europeu MHP. Com isso, um passo futuro é a integração do sistema de recomendação com o sistema de DTV ou de IPTV brasileiro para que o 2S ACT possa utilizar informações do EPG para alimentar a sua base de dados, permitindo assim um teste real envolvendo voluntários para validar o seu funcionamento.

Outra questão não tratada aqui é a da privacidade. Monitorar o comportamento de usuários e transmitir essa informação para outros locais vai de encontro a esta questão, e

isso causa uma barreira natural quanto ao uso de sistemas de recomendação. Por este motivo, o 2S ACT trabalha localmente no STB somente com os perfis das pessoas que frequentam o local. Porém, pretende-se desenvolver uma versão do 2S ACT centralizada para ter um volume de dados maior, similar ao do prêmio Netflix, permitindo ao sistema gerar recomendações cada vez melhores. Só que essa questão traz novamente a discussão sobre a privacidade, também tratados pelo sistema da TiVo, que mascara o identificador dos usuários, não armazena nenhuma informação no servidor, e altera todas as datas do log para o dia 01/01/1970 (ALI; STAM, 2004). Outra forma de tratar a questão da privacidade é usar a estrutura de compartilhamento de informações de redes sociais. Como nesses tipos de rede as pessoas controlam quem pode e quem não pode acessar cada informação pessoal, tem-se com objetivo desenvolver um aplicativo para redes sociais que se comunica com o 2S ACT central. Através desse aplicativo, os usuários podem modificar o seu perfil calculado, mudar de cluster, e até ver o perfil calculado de seus amigos, se estes assim permitirem. Com isso as informações do telespectador só serão compartilhadas com seus amigos autorizados. Outra opção de compartilhamento de perfil é a transmissão de perfil de um STB a outro. Esse envio de dados pode ser feito de duas formas, diretamente através do canal de retorno ou internet, ou usando uma mídia física de transporte. UHLMANN e LUGMAYR (2008), por exemplo, desenvolveram uma abordagem de armazenagem do perfil no telefone celular, com o objetivo de compartilhar essa informação com outdoors eletrônicos, quiosques, etc. Essa mesma abordagem poderia ser usada para compartilhar o perfil de um telespectador com os diversos STBs dos locais que ele frequenta.

Outro desenvolvimento futuro consiste em aproveitar o *link* entre o STB e o celular para uso de iteratividade. Cada pessoa pode se conectar ao sistema e controlar o STB através de seu próprio celular, tendo cada uma sua própria lista de favoritos e informações de cartão de crédito para compra online, por exemplo, armazenadas no telefone. Conteúdos diferenciados podem ser apresentados nos celulares de cada um dos telespectadores do grupo, dependendo o seu tipo de perfil ou da forma que o programa interativo for desenhado. Um exemplo seria programas de perguntas e respostas, onde cada telespectador pode escolher a resposta que considera certa no seu celular e ao final do programa é mostrado qual dos telespectadores teve mais acerto, e qual foi o maior placar de todos os telespectadores que assistiram a esse programa em todo o mundo.

Enfim, os objetivos do trabalho foram cumpridos, com resultados esperados que demonstraram a utilidade prática das novas arquiteturas de processadores na área de sistemas de recomendação, de tecnologias sem fio e de computação embarcada na área de reconhecimento de pessoas.

REFERÊNCIAS

ABRAMSON, A. Zworykin, pioneer of television. **Engineering Science and Education Journal**, v. 5, n. 3, jun 1996.

ABRANTES, P. F. R. **TargetAd: publicidade direccionada em sistemas IPTV**. Portugal: Instituto Superior Técnico - IST, 2009.

ABTA. **Dados do setor**. Disponível em: <http://tvporassinatura.org.br/index.php?option=com_content&view=article&id=19&Itemid=36>. Acesso em: 1 out. 2011.

ABNT. **ABNT NBR 15606-4: Televisão digital terrestre — Codificação de dados e especificações de transmissão para radiodifusão digital Parte 4: Ginga-J - Ambiente para a execução de aplicações procedurais**. [S.l.]: ABNT, 2010a.

ABNT. **ABNT NBR 15606-6: Televisão digital terrestre — Codificação de dados e especificações de transmissão para radiodifusão digital Parte 6: Java DTV 1.3**. [S.l.]: ABNT, 2010b.

ADOMAVICIUS, G.; TUZHILIN, E. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING**, v. 17, p. 734--749, 2005.

ALLROVI. **Movies and films database - Movie search, guide, recommendations, and reviews**. Disponível em: <<http://www.allrovi.com/movies?r=allmovie>>. Acesso em: 10 ago. 2010.

ALI, K.; STAM, W. VAN. **TiVo: making show recommendations using a distributed collaborative filtering architecture**. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. **Anais...**, KDD '04. New York, NY, USA: ACM. Disponível em: <<http://doi.acm.org/10.1145/1014052.1014097>>. , 2004.

AYGUADÉ, E.; BADIA, R.; IGUAL, F. *et al.* An extension of the StarSs programming model for platforms with multiple GPUs. **Euro-Par 2009 Parallel Processing**. [S.l.: s.n.], 2009. p. 851-862.

AVETANA. **Bluetooth JSR82 implementation for Windows, MacOS X and Linux systems.** Disponível em: <<http://www.avetana-gmbh.de/avetana-gmbh/produkte/jsr82.eng.xml>>. Acesso em: 1 out. 2011.

BAI, Y.; WANG, F.; LIU, P. Efficiently filtering RFID data streams. In: **CLEANDB WORKSHOP**, p. 50--57, 2006.

BAKKUM, P.; SKADRON, K. **Accelerating SQL database operations on a GPU with CUDA.** In: Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units - GPGPU '10. **Anais...** Pittsburgh, Pennsylvania: [s.n.]. Disponível em: <<http://dl.acm.org/citation.cfm?id=1735706>>. Acesso em: 17 set. 2011. , 2010.

BHATIA, N.; PATNAIK, P. **Netflix recommendation based on IMDB.** . [S.l: s.n.]. Disponível em: <<http://nipunbhatia.googlepages.com/BhatiaPatnaik-NetFlixMovieRecommendanda.pdf>>. , 12 mar 2008.

BELL, R. M.; KOREN, Y.; VOLINSKY, C. **The BellKor 2008 solution to the Netflix prize.** . [S.l: s.n.]. Disponível em: <http://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf>. Acesso em: 22 jun. 2009. , [S.d.]

BENNETT, J. **The Cinematch system: operation, scale, coverage, accuracy, impact.** . [S.l: s.n.]. Disponível em: <<http://blog.recommenders06.com/wp-content/uploads/2006/09/bennett.pdf#search=%22cinematch%22>>. Acesso em: 23 jun. 2009. , 13 set 2006.

BENNETT, J.; LANNING, S. The Netflix prize. In: **KDD CUP AND WORKSHOP IN CONJUNCTION WITH KDD**, 2007.

BLACKBERRY. **7100i Wireless Handheld Device.** Disponível em: <<http://na.blackberry.com/eng/devices/device-detail.jsp?navId=H0,C61,P61,>> Acesso em: 15 set. 2008.

BLUECOVE. **BlueCove JSR-82 project.** Disponível em: <<http://www.bluecove.org/>>. Acesso em: 15 set. 2008.

BLUETOOTH. **Architecture & terminology overview, Bluetooth specification version 2.1 + EDR**, vol. 1, 26 Jul. 2007.

BONNEFOY, D.; BOUZID, M.; LHUILLIER, N.; MERCER, K. "More Like This" or "Not for Me": delivering personalised recommendations in multi-user environments. In: CONATI, C. (Eds). **User Modeling 2007**. Lecture Notes in Computer Science. [S.l.]: Springer Berlin / Heidelberg, 2009. v. 4511p. 87-96.

BOZIOS, T.; LEKAKOS, G.; SKOULARIDOU, V. Advanced techniques for personalized advertising in a digital TV environment: The imedia system. In: **PROCEEDINGS OF THE EBUSINESS AND EWORK CONFERENCE**, 2001.

BRIDGELALL, R. Bluetooth/802.11 Protocol Adaptation for RFID. **TAGS, PROCEEDINGS OF THE 4TH EUROPEAN WIRELESS CONFERENCE**, 2002.

BRIDGELALL, R. **RFID device systema and method of operation including a hybrid backscatter based RFID tag protocol compatible with RFID Bluetooth and or IEEE 802.11 X infrastructure**. US 7215976. [S.l.] , 8 maio 2007.

BUCHSBAUM W. TV remote controls for the technician. **Radio and TV News**, v. 58, no. 5, pp 58, Nov. 1957.

BUEHRER, G.; PARTHASARATHY, S. **The potential of the Cell Broadband Engine for data mining**. . Ohio State University Technical Report OSU-CISRC-3/07--TR22: [s.n.]. Disponível em: <<ftp://ftp.cse.ohio-state.edu/pub/tech-report/2007/TR22.pdf>>. , 2007.

BUTTS, M. Synchronization through communication in a Massively Parallel Processor array. **IEEE Micro**, v. 27, n. 5, p. 32-40, 2007.

CHANG, Y.-H.; COGGINS, D.; PITT, D. et al. An open-systems approach to video on demand. **IEEE Communications Magazine**, v. 32, p. 68-80, maio 1994.

DAVIDSON, J.; LIEBALD, B.; LIU, J. et al. **The YouTube video recommendation system**. In: Proceedings of the fourth ACM conference on Recommender systems. **Anais...**, RecSys '10. New York, NY, USA: ACM. Disponível em: <<http://doi.acm.org/10.1145/1864708.1864770>>. , 2010.

DAY, M.; HOFSTEE, P. **Hardware and software architectures for the CELL processor**. In: Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, 2005. CODES+ISSS '05. **Anais...** [S.l.]: IEEE. , set 2005.

DISZ, T.; OLSON, R.; STEVENS, R. **Performance model of the Argonne Voyager multimedia server**. In: IEEE International Conference on Application-Specific Systems, Architectures and Processors, 1997. Proceedings. **Anais...** [S.l.]: IEEE. , 14 jul 1997.

DUMAN, H.; HAGRAS, H.; CALLAGHAN, V. Adding intelligence to ubiquitous computing environments. In: LEE, R. S. T.; LOIA, V. (Eds.). **Computational Intelligence for Agent-based Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. v. 72p. 61-102.

DVB. **ETSI TS 102 727 V1.1.1: Digital Video Broadcasting (DVB); Multimedia home platform (MHP) specification 1.2.2**. [S.l.]: DVB. Acesso em: 4 out. 2011, 2010a.

DVB. **ETSI TS 102 728 V1.1.1: Digital Video Broadcasting (DVB); Globally executable MHP (GEM) specification 1.2.2 (including IPTV)**. [S.l.]: DVB. Acesso em: 4 out. 2011, 2010b.

EAGLE, N.; (SANDY) PENTLAND, A. Reality mining: sensing complex social systems. **Personal and Ubiquitous Computing**, v. 10, n. 4, p. 255-268, 1 maio 2006.

E-POLL. **Program Pulse - TV program ratings and measurement of viewer attitudes and behavior**. Disponível em:
<<http://www.epollresearch.com/corp/products/programPulse.view>>. Acesso em: 1 out. 2011.

FERNANDEZ, Y. B.; ARIAS, J. J. .; NORES, M. L.; SOLLA, A. G.; CABRER, M. R. AVATAR: an improved solution for personalized TV based on semantic inference. **IEEE Transactions on Consumer Electronics**, v. 52, n. 1, p. 223- 231, fev 2006.

FOINA, A. G.; BARBIN, S. E.; RAMIREZ, F. J. **A new approach for vehicle access control using active RFID tags**. In: Microwave and Optoelectronics Conference, 2007. IMOC 2007. SBMO/IEEE MTT-S International. **Anais...** [S.l: s.n.]. , 2007.

FOINA, A. G.; RAMIREZ-FERNANDEZ, F. J. **Sistema de seleção automática de conteúdo televisivo pelo telespectador**. BR n. PI 0.801.812-0. 8 fev. 2008a.

FOINA, A. G.; RAMIREZ-FERNANDEZ, F. J. **Sistema de seleção automática de conteúdo publicitário para mídia eletrônica**. BR n. C1 0.801.812-0. 26 sep. 2008b.

FOINA, A. G. **SmarTV - a multi-person user's behaviour analysis and program recommendation system with iTV**. In: Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on. **Anais...** [S.l: s.n.]. , 2009a.

FOINA, A. G.; RAMIREZ-FERNANDEZ, J. **How a cell phone can change dramatically the way we watch tv**. In: EUROCON 2009, EUROCON '09. IEEE. **Anais...** [S.l: s.n.]. , 2009b.

FOINA, A. G.; RAMIREZ-FERNANDEZ, F. J.; BARBIN, S. E. An RFID bulk cargo supervising system. **Latin America Transactions, IEEE (Revista IEEE America Latina)**, v. 7, n. 6, p. 688-693, 2009c.

FOINA, A. G.; RAMIREZ-FERNANDEZ, F. J. HUNTER — HYBRID UNIFIED TRACKING ENVIRONMENT. **Contemporary Robotics - Challenges and Solutions**. [S.l.]: INTECH, 2009d. p. 231-248.

FOINA, A. G.; BADIA, R. M.; EL-DEEB, A.; RAMIREZ-FERNANDEZ, F. J. **Player Tracker - a tool to analyze sport players using RFID**. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on. **Anais...** [S.l.: s.n.]. Disponível em: <10.1109/PERCOMW.2010.5470538>. Acesso em: 7 jul. 2010. , 2010a.

FOINA, A. G.; RAMIREZ-FERNANDEZ, J.; BADIA, R. M. **Cell BE and bluetooth applied to digital TV**. In: Network Operations and Management Symposium (NOMS), 2010 IEEE. **Anais...** [S.l.: s.n.]. Disponível em: <10.1109/NOMS.2010.5488364>. Acesso em: 30 jun. 2010. , 2010b.

FOINA, A.; BADIA, R.; RAMIREZ-FERNANDEZ, J. G-Means improved for Cell BE environment. **Facing the Multicore-Challenge**. Lecture Notes in Computer Science. [S.l.]: Springer Berlin / Heidelberg, 2011a. v. 6310p. 54-65.

FOINA, A. G.; PLANAS, J.; BADIA, R. M.; RAMIREZ-FERNANDEZ, F. J. **P-means, a parallel clustering algorithm for a heterogeneous multi-processor environment**. In: 2011 International Conference on High Performance Computing and Simulation (HPCS). **Anais...** [S.l.]: IEEE. , 4 jul 2011b.

GARTRELL, M.; XING, X.; LV, Q. et al. **Enhancing group recommendation by incorporating social relationship interactions**. In: Proceedings of the 16th ACM international conference on Supporting group work. **Anais...**, GROUP '10. New York, NY, USA: ACM. , 2010.

GLOBO.COM. **Uruguai adota padrão nipo-brasileiro de TV digital - O Globo Online**. Disponível em: <<http://oglobo.globo.com/tecnologia/mat/2010/12/28/uruguai-adota-padrao-nipo-brasileiro-de-tv-digital-923380736.asp>>. Acesso em: 28 dez. 2010.

GOEL, D.; BATRA, D. **Predicting user preference for movies using NetFlix database**. . [S.l.]: Unpublished. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary10.1.1.115.6646>>. Acesso em: 23 jun. 2009. , 2008.

GONZALEZ, J.; GIMENEZ, J.; LABARTA, J. **Automatic detection of parallel applications computation phases**. In: Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing. **Anais...** [S.l.]: IEEE Computer Society. Disponível em: <<http://portal.acm.org/citation.cfm?id=1587764>>. Acesso em: 30 jun. 2010. , 2009.

GROUPLENS RESEARCH | **About GroupLens**. Disponível em: <<http://www.grouplens.org/>>. Acesso em: 1 out. 2011.

HAMERLY, G.; ELKAN, C. Learning the K in K-Means. In: **Neural Information Processing Systems**, v. 16, p. 281-288, 2003.

HARA, Y.; TOMOMUNE, Y.; SHIGEMORI, M. Categorization of japanese TV viewers based on program genres they watch. **User Modeling and User-Adapted Interaction**, v. 14, n. 1, p. 87–117, fev 2004.

IATRINO, A.; MODEO, S. **EPG-board a social application for the OmegaBox media center**. In: Proceedings of the 5th European conference on Interactive TV: a shared experience. **Anais...**, EuroITV'07. Berlin, Heidelberg: Springer-Verlag. Disponível em: <<http://dl.acm.org/citation.cfm?id=1763017.1763022>>. , 2007.

IMDB. **Alternative Interfaces**. Disponível em: <<http://www.imdb.com/interfaces#plain>>. Acesso em: 1 out. 2011.

IMDBPY. **IMDBPY**. Disponível em: <<http://imdbpy.sourceforge.net/>>. Acesso em: 1 out. 2011.

INOVAÇÃO TECNOLÓGICA. **Smartphone controla TV digital via Wi-Fi**. Disponível em: <<http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=smartphone-controla-tv-digital-wi-fi&id=010150110516&ebol=sim>>. Acesso em: 24 maio. 2011.

JACKSON, M. M. Pervasive direct brain interfaces. **IEEE Pervasive Computing**, v. 5, n. 4, p. 15-17, dez 2006.

JAGER, B. D. D. **United States Patent: D185741 Television Remote Control Handset**. US No. D185741 [S.l.]. Disponível em: <<http://patft.uspto.gov/netacgi/nph-Parser?Sect2=PTO1&Sect2=HITOFF&p=1&u=%2Fnetacgi%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&d=PALL&RefSrch=yes&Query=PN%2FD185741>>. Acesso em: 1 out. 2011, 28 jul 1959.

JAMESON, A. **More than the sum of its members: challenges for group recommender systems**. In: Proceedings of the working conference on Advanced visual interfaces. **Anais...**, AVI '04. New York, NY, USA: ACM. Disponível em: <<http://doi.acm.org/10.1145/989863.989869>>, 2004.

KIM, E.; PYO, S.; PARK, E.; KIM, M. An automatic recommendation scheme of TV program contents for (IP)TV personalization. **IEEE Transactions on Broadcasting**, v. 57, n. 3, p. 674-684, set 2011.

KIRMEMIS, O.; BIRTURK, A. **A content-based user model generation and optimization approach for movie recommendation**. Chicago, Illinois, USA: [s.n.]. Disponível em: <<http://www.aaai.org/Library/Workshops/2008/ws08-06-009.php>>. Acesso em: 21 ago. 2009. , 13 jul 2008

KRUMM, J.; SHAFER, S.; WILSON, A. **How a smart environment can use perception**. In: UBICOMP 2001 Workshop on Perception for Ubiquitous Computing. **Anais...** [S.l: s.n.]. , 2001.

LAI, C.-F.; HSIEN-CHAO HUANG; YUEH-MIN HUANG. **A low complexity TV remote controller with user individual behavior embedded**. In: Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on. **Anais...** [S.l: s.n.]. , 2007.

LEE, J.-J.; SEO, K.-H.; OH, C.; BIEN, Z. Z. Development of a future Intelligent Sweet Home for the disabled. **Artificial Life and Robotics**, v. 11, p. 8-12, 25 jan 2007.

LIM, J.; KIM, MUNJO; LEE, B. *et al.* A target advertisement system based on TV viewer's profile reasoning. **Multimedia Tools and Applications**, v. 36, p. 11-35, 27 jan 2007.

LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: item-to-item collaborative filtering. **IEEE Internet Computing**, v. 7, n. 1, p. 76- 80, fev 2003.

LIVE. **IST LIVE**. Disponível em: <<http://www.ist-live.org/>>. Acesso em: 1 out. 2011.

LUCAS, A. S.; ZORZO, S. D. **Personalização para Televisão Digital utilizando a estratégia de Sistema de Recomendação para ambientes multiusuário**. In: 27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. **Anais...** Recife, Brasil: [s.n.]. 2009.

MACQUEEN, J. **Some methods of classification and analysis of multivariate observations**. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. **Anais...** [S.l: s.n.]. , 1967.

MEDINA, A. G.; CASTELLANI, C. C. **Tecnologia Assistiva: Em busca de um dispositivo assistivo que possibilite o brincar**. Monografia graduação (Medicina), Faculdade de Medicina, Universidade de São Paulo, 2003.

MLAKAR, T.; ZALETELJ, J.; TASIC, J. F. **Viewer authentication for personalized iTV services**. In: Image Analysis for Multimedia Interactive Services, 2007. WIAMIS '07. Eighth International Workshop on. **Anais...** [S.l: s.n.], 2007.

MySQL. **MySQL :: SQLite Migration - Converter tools**. Disponível em: <<http://forums.mysql.com/read.php?145,68269,92627>>. Acesso em: 13 out. 2011.

NET. **Programação - Grade de canais**. Disponível em: <http://www.netcombo.com.br/netPortalWEB/appmanager/portal/desktop?_nfpb=true&_pageLabel=programacao_gradedecanais_gradedecanais_home_page>. Acesso em: 1 out. 2011.

NIELSEN. **Nielsen and INVIDI technologies in agreement to advance measurement of targeted TV ads**. Disponível em: <http://www.nielsen.com/us/en/insights/press-room/2007/Nielsen_and_INVIDI_Technologies_in_Agreement_to_Advance_Measurement_of_Targeted_TV_Ads.html>. Acesso em: 1 dez. 2007.

NOKIA. **Nokia Nseries N95**. Disponível em: <<http://www.nseries.com/products/n95/>>, Acesso em: 15 set. 2008.

NVIDIA. **Next-generation NVIDIA ION graphics processors**. Disponível em: <http://www.nvidia.com/object/sff_ion.html>. Acesso em: 1 out. 2011a.

NVIDIA. **NVIDIA Tegra 2**. Disponível em: <<http://www.nvidia.com/object/tegra-2.html>>. Acesso em: 1 out. 2011b.

OPENMP. **OpenMP.org**. Disponível em: <<http://openmp.org/wp/>>. Acesso em: 14 out. 2011.

PAULOS, E.; GOODMAN, E. **The familiar stranger: anxiety, comfort, and play in public places**. In: Proceedings of the SIGCHI conference on Human factors in computing systems. **Anais...** Vienna, Austria: ACM. Disponível em: <<http://portal.acm.org/citation.cfm?id=985721>>. Acesso em: 4 set. 2009. , 2004.

PELLEG, D.; MOORE, A. X-means: extending k-means with efficient estimation of the number of clusters. In: **Proceedings of the 17th International Conference on Machine Learning**, p. 727--734, 2000.

PÉREZ, J. M.; BELLENS, P.; BADIA, R. M.; LABARTA, J. CellSs: Making it easier to program the Cell Broadband Engine processor. **IBM Journal of Research and Development**, v. 51, n. 5, p. 593-604, 2007.

PEREZ, J. M.; BADIA, R. M.; LABARTA, J. **A dependency-aware task-based programming environment for multi-core architectures**. In: Cluster Computing, 2008 IEEE International Conference on. **Anais...** [S.l: s.n.], 2008.

PHAM, D. C.; AIPPERSPACH, T.; BOERSTLER, D. *et al.* Overview of the architecture, circuit design, and physical implementation of a first-generation cell processor. **Solid-State Circuits, IEEE Journal of**, v. 41, n. 1, p. 179-196, 2006.

PLANAS, J.; BADIA, R. M.; AYGUADÉ, E.; LABARTA, J. Hierarchical task-based programming with StarSs. **Int. J. High Perform. Comput. Appl.**, v. 23, n. 3, p. 284-299, 2009.

ROSACI, D.; SARNÉ, G. M. L.; GARRUZZO, S. MUADDIB: A distributed recommender system supporting device adaptivity. **ACM Trans. Inf. Syst.**, v. 27, n. 4, p. 24:1–24:41, nov 2009.

SATTLER, M. **Internet TV with CU-SeeMe**. [S.l.]: Sams, 1995.

SIEGEMUND, F.; FLORKEMEIER, C. **Interaction in pervasive computing settings using Bluetooth-enabled active tags and passive RFID technology together with mobile phones**. In: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003). **Anais...** [S.l.]: IEEE. , 23 mar 2003.

SMYTH, B.; COTTER, P. A personalised TV listings service for the digital TV age. **Knowledge-Based Systems**, v. 13, n. 2-3, p. 53-59, abr 2000.

SMYTH, B. Case-Based recommendation. **The Adaptive Web**. [S.l: s.n.], 2007. p. 342-376.

SOUZA FILHO, G. L. DE; LEITE, L. E. C.; BATISTA, C. E. C. F. Ginga-J: the procedural middleware for the Brazilian digital TV system. **Journal of the Brazilian Computer Society**, v. 12, p. 47 - 56, 2007.

SQLITE. **SQLITE**. Disponível em: <<http://www.sqlite.org/>>. Acesso em: 1 out. 2011.

STASHOWER, D. **The boy genius and the mogul: The untold story of television**. 1. ed. [S.l.]: Broadway, 2002.

STOCK, O.; ZANCANARO, M.; BUSETTA, P. et al. Adaptive, intelligent presentation of information for the museum visitor in PEACH. **User Modeling and User-Adapted Interaction**, v. 17, n. 3, p. 257–304, jul 2007.

TAKAMA, Y.; MUTO, Y. Profile generation for TV program recommendation based on utterance analysis. **Journal of Advanced Computational Intelligence and Intelligent Informatics**, v. Vol.13, n. No.2, p. pp. 86-90, mar 2009.

TAYLOR, A.; HARPER, R. Switching on to switch off: An analysis of routine TV watching habits and their implications for electronic programme guide design. **USABLEITV**, v. 1, p. 7–13, 2002.

TiVo. **TiVo Premiere: much more than a DVR**. Disponível em: <<http://www.tivo.com/>>. Acesso em: 2 out. 2011.

TSUNODA, T.; HOSHINO, M. Automatic metadata expansion and indirect collaborative filtering for TV program recommendation system. **Multimedia Tools Appl.**, v. 36, n. 1-2, p. 37–54, jan 2008.

TV GENIUS. **An Integrated Approach to TV and VoD Recommendations**. Disponível em: <<http://www.tvgenius.net/resources/white-papers/an-integrated-approach-to-tv-recommendations/>>. Acesso em: 1 out. 2011.

UHLMANN, S.; LUGMAYR, A. **Personalization algorithms for portable personality**. In: Proceedings of the 12th international conference on Entertainment and media in the ubiquitous era. **Anais...** Tampere, Finland: ACM. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1457199.1457225>>. Acesso em: 22 jun. 2009. , 2008.

VILAS, M.; PANEDA, X. G.; GARCIA, R.; MELENDI, D.; GARCIA, V. G. **User behavior analysis of a video-on-demand service with a wide variety of subjects and lengths**. In: Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference on. **Anais...** [S.l: s.n.]. , 2005.

VISIBLE WORLD. **About | Connect**. Disponível em: <<http://www.visibleworld.com/Connect/About>>. Acesso em: 1 out. 2011.

WALLACE, D. A primer for digital television RF facility design and construction. **Broadcasting, IEEE Transactions on**, v. 43, n. 1, p. 3-11, 1997.

WU, N. C.; NYSTROM, M. A.; LIN, T. R.; YU, H. C. **Challenges to global RFID adoption**. In: Technology Management for the Global Future, 2006. PICMET 2006. **Anais...** [S.l: s.n.], 2006.

YU, Z.; ZHOU, X.; HAO, Y.; GU, J. TV program recommendation for multiple viewers based on user profile merging. **User Modeling and User-Adapted Interaction**, v. 16, n. 1, p. 63-82, 1 mar 2006.

YU, Z.; ZHOU, X. A collaborative approach for user profile capturing in ubiquitous environments. **Journal of Information and Computing Science**, v. 3, n. 2, p. 90-96, 2008.

VISIBLE WORLD. **About | Connect**. Disponível em:
<<http://www.visibleworld.com/Connect/About>>. Acesso em: 1 out. 2011.

ZUO, F.; WITH, P. H. . DE. Real-time embedded face recognition for smart home. **IEEE Transactions on Consumer Electronics**, v. 51, n. 1, p. 183- 190, fev 2005.

APÊNDICE A - SCRIPT DE CORRELAÇÃO ENTRE A BASE IMDB E O NETFLIX

```

1  #!c:\Python2.4\python.exe
2  try:
3      import os, sys, time
4      import imdb
5      import pickle
6      from sqlalchemy import *
7  except ImportError:
8      print 'You need to install the IMDBPY package!'
9      sys.exit(1)
10
11  CTIME = int(time.time())
12  BEGIN_TIME = CTIME
13
14  in_encoding = sys.stdin.encoding or sys.getdefaultencoding()
15  out_encoding = sys.stdout.encoding or sys.getdefaultencoding()
16
17  class Netflix2imdb(SQLObject):
18      netflixid = IntCol(length=5, unique=True)
19      imdbid = IntCol(length=11)
20
21  class Movies(SQLObject):
22      title = StringCol()
23      year = IntCol()
24
25
26  i = imdb.IMDb('sql', uri='mysql://user:pass@localhost/imdb')
27  sqlhub.processConnection = connectionForURI('mysql://user:pass@localhost/netflix')
28
29  Netflix2imdb.createTable(ifNotExists=True)
30
31  missing = 0
32  found = 0
33  overlap = 0
34  print "Processing movies."
35
36  count = 0
37  people = {}
38  genre = []
39  countries = []
40  specEffects = []
41  visEffects = []
42  foundstr = ''
43
44  for movie in Movies.select():
45      title = unicode(movie.title, in_encoding, 'replace')
46      try:
47          results = i.search_movie(title)
48          if not results:
49              missing += 1
50              count += 1
51          else:
52              try:
53                  count += 1
54                  done = 0
55                  temp = results[0]
56                  if temp['year'] == movie.year:
57                      foundstr = "Hit: " + title + "(" + str(movie.year) + ") IMDB: " + temp['long imdb canonical title']
58                      record = Netflix2imdb(netflixid=movie.id, imdbid=temp.movieID)
59                      done = 1
60                      found += 1
61                  else:
62                      if missing % 50 == 0: print "Misseds: " + str(missing)
63                      missing += 1
64              except:
65                  print "Unexpected error:", sys.exc_info()[0]
66          except imdb.IMDbError, e:
67              print "Error report:"
68              print e
69              continue
70
71      if count % 50 == 0: print foundstr
72
73      if count % 500 == 0:
74          print str(count) + " movies processed. Found: " + str(found) + " Missed: " + str(missing)
75
76          print '(in %d minutes, %d seconds)' % \
77              divmod(int(time.time())-BEGIN_TIME, 60)
78
79  print "\n ---- Done ---- \n"
80  print "Movies: " + str(count) + ", Missing: " + str(missing) + "\n"
81
82  print 'Hoorah! (%d minutes, %d seconds)' % \
83      divmod(int(time.time())-BEGIN_TIME, 60)

```

APÊNDICE B - CÁLCULO DO RMSE USANDO O PROCEDIMENTO DO BELLKOR

```
mysql> select sum(rating)/count(rating) from ratings_bellkor;
+-----+
| sum(rating)/count(rating) |
+-----+
| 3.5685 |
+-----+
1 row in set (5.17 sec)
```

```
mysql> update probe_bellkor set prediction = 3.5685;
Query OK, 804883 rows affected (0.16 sec)
Rows matched: 804886 Changed: 804883 Warnings: 0
```

```
mysql> update ratings_bellkor r set residual = residual - 3.5685
Query OK, 55912149 rows affected (2 min 23.91 sec)
Rows matched: 55912149 Changed: 55912149 Warnings: 0
```

```
mysql> select * from rmse_bellkor;
+-----+
| rmse |
+-----+
| 1.12486144021291 |
+-----+
1 row in set (0.26 sec)
```

```
mysql> create table movie_effect (movie_id int(5), sum_r_ui_x_ui double,
sum_x_ui_squared double, n_i int(3), theta_i_hat double, theta_i double);
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> insert into movie_effect (movie_id, sum_r_ui_x_ui, sum_x_ui_squared,
n_i) select movie_id, sum(residual), count(residual), count(residual) from
ratings_bellkor group by movie_id;
Query OK, 7500 rows affected (14.45 sec)
Records: 7500 Duplicates: 0 Warnings: 0
```

```
mysql> update movie_effect set theta_i_hat =
sum_r_ui_x_ui/sum_x_ui_squared;
Query OK, 7500 rows affected (0.06 sec)
Rows matched: 7500 Changed: 7500 Warnings: 0
```

```
mysql> update movie_effect set theta_i = (n_i*theta_i_hat)/(n_i + 25);
Query OK, 7500 rows affected (0.03 sec)
Rows matched: 7500 Changed: 7500 Warnings: 0
```

```
mysql> create index index_movie_effect_movie_id on movie_effect(movie_id);
Query OK, 7500 rows affected (0.15 sec)
Records: 7500 Duplicates: 0 Warnings: 0
```

```
mysql> alter table movie_effect engine=memory;
Query OK, 7500 rows affected (0.11 sec)
Records: 7500 Duplicates: 0 Warnings: 0
```

```
mysql> update probe_bellkor p set prediction = prediction + (select theta_i
from movie_effect where movie_id = p.movie_id);
Query OK, 804886 rows affected (3.22 sec)
```

Rows matched: 804886 Changed: 804886 Warnings: 0

```
mysql> select * from rmse_bellkor;
```

```
+-----+
| rmse          |
+-----+
| 1.05952212409533 |
+-----+
1 row in set (0.11 sec)
```

```
mysql> update ratings_bellkor r set residual = residual - (select theta_i
from movie_effect where movie_id=r.movie_id);
```

Query OK, 55912149 rows affected (4 min 42.76 sec)

Rows matched: 55912149 Changed: 55912149 Warnings: 0

```
mysql> alter table movie_effect engine=myisam;
```

Query OK, 7500 rows affected (0.09 sec)

Records: 7500 Duplicates: 0 Warnings: 0

```
mysql> create table user_effect (customer_id int(6), sum_r_ui_x_ui double,
sum_x_ui_squared double, n_u int(3), theta_u_hat double, theta_u double);
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> insert into user_effect (customer_id,
sum_r_ui_x_ui,sum_x_ui_squared,n_u) select customer_id, sum(residual),
count(residual),count(residual) from ratings_bellkor group by customer_id;
```

Query OK, 478865 rows affected (39.50 sec)

Records: 478865 Duplicates: 0 Warnings: 0

```
mysql> update user_effect set theta_u_hat = sum_r_ui_x_ui/sum_x_ui_squared;
```

Query OK, 478865 rows affected (1.03 sec)

Rows matched: 478865 Changed: 478865 Warnings: 0

```
mysql> update user_effect set theta_u = (n_u*theta_u_hat)/(n_u + 7);
```

Query OK, 478865 rows affected (1.06 sec)

Rows matched: 478865 Changed: 478865 Warnings: 0

```
mysql> create index index_user_effect_customer_id on
user_effect(customer_id);
```

Query OK, 478865 rows affected (0.84 sec)

Records: 478865 Duplicates: 0 Warnings: 0

```
mysql> set max_heap_table_size=32000000;
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> alter table user_effect engine=memory;
```

Query OK, 478865 rows affected (0.37 sec)

Records: 478865 Duplicates: 0 Warnings: 0

```
mysql> update probe_bellkor p set prediction = prediction + (select theta_u
from user_effect where customer_id=p.customer_id);
```

Query OK, 804886 rows affected (3.59 sec)

Rows matched: 804886 Changed: 804886 Warnings: 0

```
mysql> update probe_bellkor set prediction = 5 where prediction > 5;
```

Query OK, 3593 rows affected (0.05 sec)

Rows matched: 3593 Changed: 3593 Warnings: 0

```
mysql> update probe_bellkor set prediction = 1 where prediction < 1;
```

Query OK, 21 rows affected (0.04 sec)

Rows matched: 21 Changed: 21 Warnings: 0

```
mysql> select * from rmse_bellkor;
+-----+
| rmse          |
+-----+
| 0.966743613782026 |
+-----+
1 row in set (0.13 sec)
```

ANEXO A - SCRIPT DE CONVERSÃO DE MYSQL PARA SQLITE

Retirado de (MySQL, 2011).

```
#!/bin/sh

mysqldump --compact --compatible=ansi --default-character-
set=binary mydbname |
grep -v ' KEY "' |
grep -v ' UNIQUE KEY "' |
perl -e 'local $/;$_=<>;s/,\\n\\)/\\n\\)/gs;print
"begin;\\n";print;print "commit;\\n"' |
perl -pe '
if (/^(INSERT.+?)\\(/) {
    $a=$1;
    s/\\'\\'\\'\\'/'\\'\\'\\'\\'/g;
    s/\\n\\n/g;
    s/\\),\\(/\\);\\n$a\\(/g;
}
' |
sqlite3 output.db
```