

## 2. Escalonamento de Processos em Sistemas Computacionais Distribuídos

Este capítulo aborda a utilização de sistemas computacionais distribuídos, enfocando as atividades envolvidas no escalonamento de processos e no balanceamento de cargas nesses sistemas.

O problema do escalonamento de processos em sistemas computacionais distribuídos não é novo, mas é, ainda, amplamente investigado e discutido na literatura. As seções que seguem sintetizam as questões chaves discutidas na literatura.

### 2.1 *Considerações Iniciais*

A utilização de sistemas computacionais distribuídos tem aumentado cada vez mais em virtude do crescente avanço tecnológico aliado à necessidade de compartilhamento de informações em diferentes locais. Assim, a computação distribuída vem ganhando importância nas últimas décadas, como um dos paradigmas de computação preferido quando comparada à computação centralizada.

Existem vários campos novos e atrativos de pesquisas em sistemas computacionais distribuídos tais como: agentes móveis (Muhugusa, 1998), multimídia (Little & Ghafoor, 1990; Little & Ghafoor, 1992) e computação móvel (Davies et al, 1996; Davies et al, 1998). Por outro lado, existem áreas tradicionais e que ainda não estão totalmente sedimentadas, sendo alvo de interesse de muitas pesquisas. O desenvolvimento de técnicas e métricas eficientes para a distribuição de processos entre os elementos de processamento é ainda um grande desafio existente em sistemas computacionais distribuídos.

Essa atividade de distribuição, conhecida como escalonamento de processos, visa atingir um conjunto de objetivos relacionados com medidas de desempenho, como por exemplo: melhorar o compartilhamento de recursos, otimizar o tempo médio de resposta, promover o balanceamento de cargas, maximizar a utilização dos recursos, entre outros. Esses objetivos são muitas vezes conflitantes e o

escalador, software responsável pela atividade de escalonamento, deve tomar decisões que são influenciadas por diversos fatores: a carga de trabalho do sistema, a presença de aplicações com diferentes características, o hardware da rede de comunicação, o sistema operacional nativo e o hardware dos elementos de processamento (Souza, 2000).

## **2.2 Conceitos Essenciais de Sistemas Computacionais Distribuídos**

Muitas das diversas áreas nas quais a computação se aplica requerem cada vez mais potência computacional, em virtude dos algoritmos cada vez mais complexos que são utilizados e do tamanho do conjunto de dados envolvido no processamento. Sendo assim, a busca por melhor tempo de processamento e, conseqüentemente, melhor desempenho na execução das aplicações constitui o fator principal que impulsiona o desenvolvimento da computação paralela distribuída.

A computação paralela consiste, basicamente, no uso de elementos de processamento que cooperam e comunicam-se entre si para solucionarem problemas complexos ou não, de maneira mais rápida do que se estivessem sendo solucionados seqüencialmente (Almasi & Guttlielb, 1994). Assim, a computação paralela (executada em arquiteturas paralelas) surgiu com o objetivo de reduzir o tempo de processamento de aplicações específicas que demandam altas taxas de processamento.

Por outro lado, o surgimento dos sistemas computacionais distribuídos deu-se, basicamente, devido à necessidade de compartilhamento de recursos, normalmente de alto custo e fisicamente separados.

O desenvolvimento e disponibilidade de microprocessadores mais potentes e de menor custo, aliado ao avanço na tecnologia de comunicação de dados (propiciando a utilização de redes de computadores de alta velocidade) foram aspectos que contribuíram e estimularam consideravelmente o interesse na utilização de sistemas distribuídos.

Com o avanço tecnológico foi possível a união da computação paralela e dos sistemas computacionais distribuídos, surgindo assim o que constitui a computação paralela distribuída. Nesse caso tem-se a computação paralela sendo implementada sobre uma plataforma MIMD com memória distribuída (Tanenbaum, 1992;

Tanenbaum, 1997), podendo-se explorar características atrativas de ambas as áreas.

Apesar de serem amplamente disseminados e existirem diversas linhas de pesquisa, verifica-se que não existe consenso sobre uma definição única e exata do que seja um sistema distribuído. Mesmo diante da divergência do conceito, existem alguns pontos em comum tais como interligação de computadores através de rede de comunicação e gerenciamento por um software de sistema distribuído, que sempre são apresentados quando da explanação desses sistemas.

Neste trabalho, um sistema distribuído será visto como uma coleção de computadores autônomos e interligados por uma rede de comunicação, possuindo um sistema operacional distribuído que será responsável por coordenar as atividades desenvolvidas, além de permitir o compartilhamento dos recursos existentes no sistema.

Os sistemas computacionais distribuídos apresentam uma série de características que podem torná-los atrativos quando comparados às máquinas paralelas convencionais ou aos sistemas centralizados: disponibilidade, confiabilidade, expansibilidade, tolerância a falhas, transparência, abertura, concorrência, compartilhamento de recursos, e um possível aumento de desempenho (Tanenbaum, 1992; Müllender, 1993; Coulouris, 1994).

A heterogeneidade constitui outra característica importante dos sistemas computacionais distribuídos (Zhou et al, 1993). Essa heterogeneidade pode ser apresentada de inúmeras formas: heterogeneidade configuracional, onde máquinas possuem diferentes capacidades de processamento, espaço de memória, armazenamento em disco e assim por diante; heterogeneidade arquitetural, a qual dificulta a execução de um mesmo código em diferentes máquinas; heterogeneidade do sistema operacional onde as características dos sistemas nas diferentes máquinas variam, podendo ser incompatíveis; e por último a heterogeneidade temporal ou dinâmica, proposta nesta tese com objetivo de indicar se a plataforma está ou não heterogênea (que será apresentada no capítulo 5).

Diversos estudos indicam a importância de se explorar adequadamente o potencial representado pelo uso de plataformas distribuídas. Minimizar o tempo de execução, minimizar os atrasos na comunicação, maximizar a utilização dos recursos, maximizar o *throughput* do sistema, entre outros, são alguns dos objetivos

desses estudos. Uma das formas de se alcançar esses objetivos é através de um melhor gerenciamento e melhor alocação dos recursos relacionados à carga computacional do sistema.

Assim, as plataformas distribuídas e heterogêneas possuem algumas características interessantes, que contribuem para a sua grande utilização e destacam a importância da atividade de escalonamento. A melhor relação custo x benefício é um dos fatores responsáveis pela atenção dada à atividade de escalonamento nessas plataformas.

### **2.3 Escalonamento em Sistemas Distribuídos**

O escalonamento é estudado e praticado em áreas do conhecimento onde se deseja atribuir ou distribuir tarefas para recursos, os quais realizarão algum processamento sobre esses serviços. O escalonamento de processos pode ser considerado, de uma maneira bem simplificada, como a atividade responsável pela alocação dos processos aos processadores.

As pesquisas desenvolvidas sobre esse tema abordam, de modo geral, uma visão algorítmica do problema (Baumgartner & Wah, 1991; Chapin, 1993; Lüling et al, 1993; Henderson, 1995; El-Rewini et al., 1995; Alanyali & Hajek, 1995; Xu & Lau, 1997; Schopf, 1997; Beaumont et al, 2003).

#### **2.3.1 Componentes de um Algoritmo de Escalonamento**

Os algoritmos de escalonamento são compostos por políticas e por mecanismos, sendo os últimos responsáveis por manipular os objetivos desses algoritmos. Esses mecanismos são responsáveis pela definição de como o escalonamento será efetuado (Song et al, 1997). As políticas são responsáveis por definir o que deve ser feito para que ocorra o escalonamento. Embora não haja um consenso na determinação dos componentes de um escalonador de processos, os seguintes são bastante difundidos na literatura (Shivaratri et al., 1992):

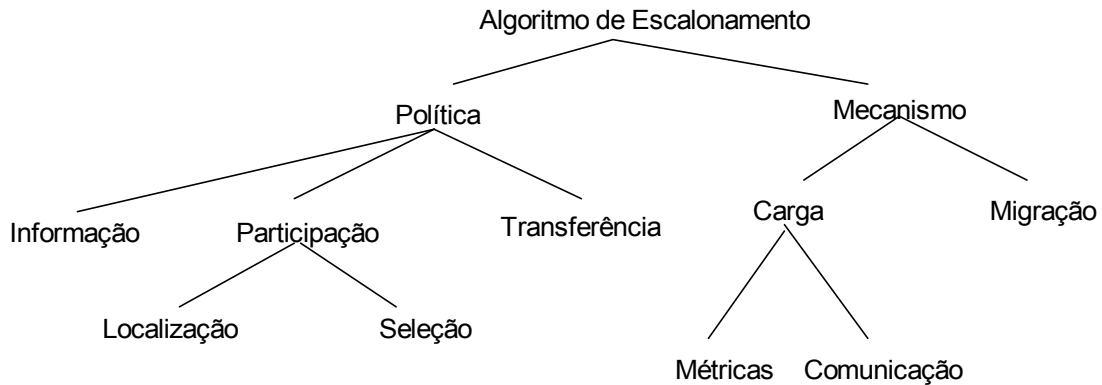
- política de transferência: determina se uma máquina está apta a participar do processo como emissora ou como receptora, conforme sua carga;
- política de seleção: escolhe a tarefa a ser transferida (geralmente a iniciada mais recentemente);

- política de localização: responsável por encontrar uma máquina parceira de transferência (emissora ou receptora) adequada para uma outra máquina, assim que a política tenha decidido que esta máquina é emissora ou receptora;
- política de informação: decide quando as informações sobre os estados de outras máquina no sistema devem ser coletadas, de onde serão coletadas, e quais informações serão coletadas. Existem três tipos de políticas de informação:
  - política orientada à demanda: uma máquina coleta o estado das outras máquina somente quando ela se torna emissora ou receptora;
  - política periódica: as informações são coletadas de tempos em tempos;
  - política orientada à mudança de estado: as informações das máquina são divulgadas conforme mude o grau de seu estado.

Os mecanismos, por outro lado, podem ser divididos em:

- mecanismo de métrica da carga: indica o método utilizado para medir a carga em cada uma das máquinas;
- mecanismo de comunicação da carga: define o método através do qual será efetuada a comunicação das informações de carga entre as diversas máquinas; e
- mecanismo de migração: define o protocolo a ser utilizado quando da ocorrência de migração de processos entre as máquinas.

Funcionalmente e hierarquicamente falando, a atividade de escalonamento poderia estar disposta como apresentado na Figura 2.1.



**Figura 2.1 - Classificação hierárquica da composição dos algoritmos de escalonamento.**

Sendo assim, o escalonamento é, normalmente, tratado segundo as políticas e os mecanismos inseridos no algoritmo do escalonador.

Cada algoritmo de escalonamento possui seus objetivos e, atrelados a esses objetivos devem estar presentes as métricas, cuja meta principal é a avaliação do escalonamento. Essas métricas devem estar voltadas para os objetivos do escalonamento e, assim como eles, devem ser flexíveis.

Mesmo fixando um objetivo, um algoritmo de escalonamento pode apresentar desempenhos distintos, ficando extremamente difícil afirmar que exista um único algoritmo capaz de produzir resultados excelentes em todas as possíveis combinações de hardware e software. Essas variações ocorridas no desempenho devem-se principalmente à influência direta de três fatores: a plataforma computacional (principalmente as distribuídas, por serem em sua maioria heterogêneas), o algoritmo de escalonamento e as classes de aplicações executadas. Assim, todo algoritmo de escalonamento tem como objetivo atuar em tipos específicos de aplicações, de modo a tentar prever o tipo e a quantidade de trabalho que irá ocorrer.

É desejável que uma estratégia de escalonamento de processos possua as seguintes propriedades (Wang & Morris, 1985):

- desempenho ideal do sistema como um todo: a capacidade total de processamento deve ser maximizada;
- equidade de serviço: o desempenho deve ser uniforme, independente da origem das tarefas;

- tolerância a falhas: a robustez do desempenho deve ser mantida na presença de falhas parciais do sistema.

O escalonamento pode ser dividido em fisicamente distribuído e fisicamente não distribuído. No fisicamente distribuído, cada elemento de processamento possui autonomia para decidir a respeito do escalonamento, ou seja, o processo de decisão é distribuído entre os elementos de processamento do sistema. O processo de escalonamento pode ser feito de forma cooperativa (considerando o estado dos outros processadores que fazem parte do sistema) ou não cooperativa (não considerando o estado dos outros processadores).

No escalonamento fisicamente não distribuído, existe um único processador responsável pelo escalonamento em todo o sistema. Essa categoria engloba três grupos:

a) abordagem determinística: mecanismo mais simples, onde o algoritmo envia o  $i$ -ésimo processo para o processador  $i \bmod N$ . Esse algoritmo funciona bem em sistemas com poucos processadores e processos independentes uns dos outros;

b) abordagem aleatória: semelhante ao anterior, entretanto, quando um processo é gerado, é verificado o estado de carga desse processador. Caso sua carga seja baixa, o processo é executado, caso contrário o processo é distribuído aleatoriamente a outro processador;

c) distribuição mestre-escravo (farmer): existe um processo mestre que distribui um conjunto de dados de um problema para um conjunto de processos escravos que residem em outros processadores. A distribuição dos processos é feita durante a compilação e os dados são distribuídos durante a execução.

### **2.3.2 Classificação dos Métodos de Escalonamento**

Com o intuito de contemplar o maior número possível de características existentes nos algoritmos de escalonamento, diversos autores tem sugerido taxonomias para a área de escalonamento de processos (Casavant & Kuhl, 1988; Shirazi et al, 1995a, 1995b; Xu & Lau, 1997; Lüling et al, 1993; Lüling & Monien, 1993).

No que concerne à classificação e taxonomia dos escalonadores de processos, também não existe uma regra ou consenso geral, de modo que os algoritmos de escalonamento podem ser classificados de várias maneiras.

Dentre as diversas taxonomias propostas destaca-se a de Casavant (Casavant & Kuhl, 1988) por ser mais abrangente e de grande aceitação. A estratégia proposta por Casavant é apresentada na Figura 2.2, sendo destacados os itens de classificação que são de interesse para este trabalho.

Essa estratégia de escalonamento é feita com base: (i) nos tipos de informações usadas para que as tarefas sejam escalonadas; (ii) onde as tarefas serão alocadas quando efetuado o re-escalonamento; (iii) onde e de que forma são efetuadas as tomadas de decisão (centralizadas ou distribuídas) e a obtenção de informações. As estratégias de interesse para este trabalho, em particular, são representadas pelas caixas pontilhadas.

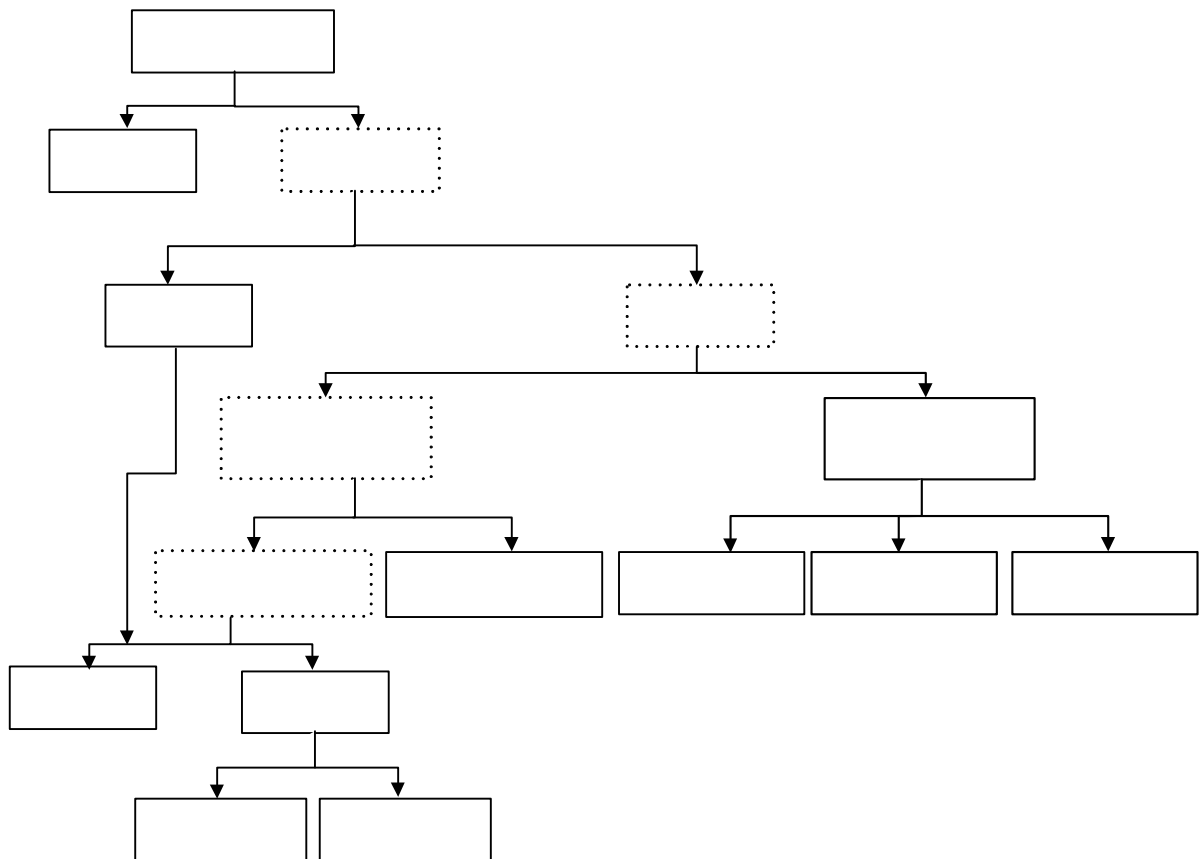


Figura 2.2 – Classificação Hierárquica Proposta por Casavant (Casavant & Kuhl, 1988)

A estratégia *local* consiste em distribuir os períodos de tempo de um processador entre os processos em execução (*time-slice*) (Tanenbaum, 1992). Nesse caso é considerada apenas a existência de um processador. A estratégia



*global* consiste na definição do elemento de processamento no qual um processo deverá ser executado em um sistema multiprocessador. Esse controle pode ser feito de forma *não distribuída* onde as tarefas e as responsabilidades da distribuição são feitas por apenas um processador, ou de forma *distribuída*, na qual as tarefas e responsabilidades são feitas por diversos elementos de processamento. A estratégia de distribuição *fisicamente distribuída* é a alternativa preferida quando as tarefas podem ser iniciadas em qualquer elemento do sistema.

A estratégia *estática* é feita antes que os processos comecem a ser executados, não levando em consideração a condição de tempo de execução das cargas já existentes no processador. Isso só é possível tendo-se conhecimento prévio sobre os tempos de execução das tarefas e sobre os recursos de processamento. Entretanto, métodos desse tipo não admitem preempção (Casavant & Kuhl, 1988; Shirazi et al, 1995a, 1995b). A estratégia *dinâmica*, por outro lado, efetua decisão de escalonamento em tempo de execução, principalmente fazendo uso de índices de carga para distinguir entre máquinas ociosas e sobrecarregadas.

A estratégia dinâmica pode ainda ser *preemptiva* ou *não-preemptiva*, sendo que a primeira permite a migração de processos que estão sendo executados, enquanto que a segunda não. Se as cargas forem alteradas drasticamente durante uma execução, uma estratégia dinâmica *preemptiva* permite que re-alocações das cargas sejam feitas, enquanto que a dinâmica *não-preemptiva* não permite.

Shivaratri (Shivaratri et al., 1992) propõe ainda que os algoritmos de escalonamento de processos sejam caracterizados como adaptativos: os algoritmos adaptativos compõem uma classe especial dos algoritmos dinâmicos. Esse tipo de algoritmo adapta sua atividade dinamicamente, mudando seus parâmetros ou mesmo sua política. É interessante utilizar esses algoritmos em sistemas que estejam uniformemente sobrecarregados, pois a transferência de carga entre as máquinas não traria benefício algum. Nessa situação, o algoritmo adaptativo suspenderia sua atividade por algum tempo.

### **2.3.3 Balanceamento de Carga em Sistemas Distribuídos**

A atividade de escalonamento consiste em selecionar uma tarefa e o local onde esta tarefa será executada. Essa atividade é comum tanto aos sistemas distribuídos quanto aos sistemas com multiprocessadores (Zaluska, 1991).

Apesar dos termos “*escalonamento de processos*” e “*balanceamento de carga*” freqüentemente serem empregados como sinônimos, o escalonamento de processos deve ser visto como uma atividade que tem como objetivo o balanceamento da carga entre os vários elementos de processamento (Wang & Morris, 1985; Casavant & Kuhl, 1988; Shivaratri et. al., 1992; El-Rewini et al., 1995; Becker, 1995; Berman, Charikar & Karpinski, 1997).

O balanceamento de carga tenta distribuir uniformemente os recursos compartilhados, evitando a situação em que um elemento de processamento está sobrecarregado enquanto que outro está com a carga de processamento leve. A Figura 2.3 ilustra um sistema distribuído sem balanceamento de carga (Shivaratri et al., 1992).



**Figura 2.3. Um sistema distribuído sem balanceamento de carga (Shivaratri et al., 1992)**

Sabendo-se que o objetivo principal dos sistemas computacionais distribuídos tem convergido para a busca de alto desempenho e que, para que esse objetivo seja alcançado, não basta apenas existirem sistemas com grande número de processadores ou de estações, fica clara a necessidade em garantir que todos os elementos de processamento pertencentes ao sistema sejam bem aproveitados.

O bom aproveitamento do sistema pode ser obtido pela distribuição dos processos entre os processadores existentes. Entretanto, para que se possa obter o melhor aproveitamento desse sistema, essa distribuição deve garantir que haja trabalho suficiente e compatível com a capacidade total do sistema e que todos os elementos de processamento tenham uma carga “uniforme” (levando-se em consideração as características peculiares de cada máquina e a carga de trabalho existente nessa), de modo que se possa evitar que alguns se tornem ociosos

enquanto outros estão sobrecarregados. A distribuição adequada de tarefas nos diversos processadores é fundamental para o melhor desempenho do sistema. A identificação dessa distribuição ideal de tarefas caracteriza o problema de balanceamento de carga.

Na literatura, inúmeras vezes, as nomenclaturas e as terminologias são confusas, de modo que balanceamento e escalonamento podem ser confundidos (Becker, 1995; Berman, Charikar & Karpinski, 1997; Xu & Lau, 1997). O balanceamento de cargas é apenas um dos objetivos do escalonamento de processos podendo ser alcançado através da utilização de políticas bem definidas. Escalonar de modo a obter um balanceamento de cargas implica, não somente em distribuir a carga através dos recursos disponíveis no sistema, mas garantir que a carga esteja equilibrada em todos esses recursos.

A função principal de uma estratégia de escalonamento de processos visando o balanceamento de cargas é recomendar decisões que permitam um aumento de desempenho. A escolha do objetivo do desempenho depende basicamente do usuário e do tipo de aplicação.

Entretanto, seja qual for o objetivo do aumento de desempenho, a estratégia de balanceamento de carga é composta basicamente de: *medidas de carga* que caracterizam o nível de carregamento de cada elemento do sistema e políticas de decisão, as quais determinam tanto a condição quanto o destino de cada tarefa a ser migrada. Políticas de balanceamento de carga provêm regras que permitem a utilização de medidas de carga para que decisões no escalonamento global possam ser tomadas, como, por exemplo, onde seria melhor a execução de uma tarefa que está chegando.

Um algoritmo de escalonamento, tendo em vista o balanceamento de cargas, deve ser capaz de controlar o efeito de diferentes fatores de desequilíbrio que ocorrem em diversas aplicações e arquiteturas. Dentre esses fatores podem ser destacados: a heterogeneidade tanto da arquitetura quanto do sistema operacional presente em um sistema computacional distribuído; o desconhecimento da quantidade de processamento envolvida em cada tarefa; a criação dinâmica de tarefas; a migração de tarefas; além da heterogeneidade e da variação da carga externa à aplicação nos diversos processadores de um ambiente não dedicado,

adaptativo e não uniforme (instável) (Casavant & Kuhl, 1988; Shirazi et al, 1995a, 1995b; Silva, 1997; Souza et al, 1999; Mehra, 1993).

Atualmente, os problemas de escalonamento, principalmente quando se deseja o balanceamento de cargas, que têm emergido em sistemas computacionais distribuídos são os não-determinísticos, uma vez que informações exatas sobre os recursos e as aplicações raramente estão disponíveis. Pode-se ainda adicionar a isso a não uniformidade ou instabilidade do sistema.

Além de suas características de software e hardware, as classes de software bem como a carga de trabalho que um sistema tem que processar, também influenciam seu desempenho.

## **2.4 Considerações Finais**

Este capítulo discutiu a importância do uso de escalonamento de processos em sistemas distribuídos mostrando que as vantagens da utilização de sistemas computacionais distribuídos sobre os sistemas monoprocessadores são muitas e para se fazer uso dessas vantagens, diversos algoritmos foram propostos com o objetivo de controlar da melhor maneira possível o compartilhamento dos recursos existentes nesse tipo de sistema.

A literatura da área demonstra que os algoritmos de escalonamento se propõem a uma grande variedade de objetivos: compartilhamento de carga, aumento da utilização do processador, redução do tempo de resposta, redução do tempo de execução, balanceamento de cargas, entre outros (Shirazi & Hurson, 1992; Müllender, 1993; Feitelson et al, 1997; Feitelson & Rudolph, 1995, 1996, 1998; Xu & Lau, 1997).

A utilização dos algoritmos de escalonamento, visando o balanceamento de cargas, pode prover um aumento considerável no desempenho do sistema, uma vez que a otimização e a eficácia do sistema estão diretamente relacionadas com a melhor utilização da capacidade de processamento existente. Desse modo, a escolha correta da política de balanceamento de cargas implica diretamente no desempenho final do sistema e, indiscutivelmente, o balanceamento de cargas é indispensável em um sistema computacional distribuído.

Entretanto, qualquer que seja a política de escalonamento a ser utilizada, objetivando, contudo, promover o balanceamento de cargas do sistema, precisa

utilizar informações sobre a carga de trabalho e a situação de operação de cada um dos elementos envolvidos no processo.

Essas informações podem ser obtidas através dos índices de carga, ou mais adequadamente através dos índices de desempenho, que podem, potencialmente, fornecer uma informação que leve em consideração a heterogeneidade das máquinas existentes.

Pesquisas e, mais especificamente, medições, têm mostrado que os benefícios, como o aumento de desempenho, que podem ser obtidos através do balanceamento de carga são extremamente dependentes desses índices de carga. Dessa maneira, acredita-se que um maior aumento de desempenho possa ser obtido com a utilização de índices de desempenho.

Partindo dessas premissas, o próximo capítulo apresenta definições e discute conceitos de índice de carga em sistemas computacionais distribuídos, bem como a caracterização da carga de trabalho.