

**Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Computação e Estatística**

SCE 217 – Programação Concorrente

Prof^a Regina Helena C. Santana

INTRODUÇÃO À TAXONOMIA E ESCALONAMENTO DE PROCESSOS

Nomes:

Eduardo Antonio Speranza

Fábio Barbano Martins

Hélder Máximo Botter Ribas

Jelder Marchi Pini

Luís Manoel Silveira de Andrade

Índice

1. Introdução.....	1
2. Escalonamento de Processos em Sistemas Distribuídos.....	2
3. Taxonomia de Escalonamento de Processos em Sistemas Distribuídos.....	4
3.1. Classificação Hierárquica.....	4
3.2. Outros Mecanismos Relacionados.....	9
4. Balanceamento de Carga.....	10
5. Conclusão.....	12
6. Referências.....	20

1.Introdução

Os sistemas computacionais distribuídos representam uma alternativa interessante para aumentar o desempenho sem os enormes custos das máquinas paralelas. Transformar uma rede comum, com máquinas de hardware modesto em um ambiente distribuído, capaz de rodar programas paralelos de forma eficiente, tornou-se uma prática bastante comum nos dias de hoje. Esses sistemas vêm mostrando cada dia mais suas vantagens e rapidamente conquistaram um lugar de destaque no cenário computacional.

O fato de as máquinas paralelas terem memória compartilhada e suas unidades processadoras serem geralmente iguais e sincronizadas facilita bastante o trabalho de divisão de tarefas, ou seja, uma distribuição de tarefas de forma equivalente para os processadores pode ser um bom início. O mesmo não se aplica de forma eficiente nos sistemas distribuídos, pois conhecidas as características das CPUs, é provável que aquela com maior poder computacional concluirá sua tarefa bem antes que as mais modestas, vindo a ficar ociosa. É fácil perceber que, neste caso, podemos não ter o resultado esperado, e pior, podemos até ter um desempenho inferior ao que teríamos se tivéssemos colocado todas as tarefas na máquina mais robusta.

Além de possuímos as tarefas a serem executadas, precisamos dividir da melhor maneira possível essas tarefas, levando em conta as características do sistema que está sendo utilizado. Para isso, alguns fatores importantes precisam ser analisados. Primeiro, procurar a melhor maneira de medir o "poder" de processamento das máquinas e o grau de ociosidade dos mesmos. Depois, tentar medir o "peso" (carga) das tarefas a serem distribuídas e estimar o tempo (ou a ordem) de processamento necessário. Com estes dados, é possível então providenciar uma correta distribuição destas tarefas, ou seja, é possível buscar o escalonamento ótimo.

2. Escalonamento de Processos em Sistemas Distribuídos

O escalonamento de processos pode ser considerado como a atividade responsável pela alocação dos processos aos processadores.

Os algoritmos de escalonamento são compostos por políticas e por mecanismos. Os mecanismos definem como o escalonamento será feito, e as políticas definem o que deve ser feito para que ocorra o escalonamento e podem ser divididas em:

- política de transferência, que determina se uma máquina está ou não pronta para receber ou enviar processos;
- política de seleção, que atua no momento em que uma máquina torna-se pronta para enviar uma tarefa, determinando qual tarefa será transferida;
- política de localização, que encontra uma máquina receptora ou transmissora para um processo, ou seja, para onde ele será transferido;
- política de informação, que decide quando, onde e quais informações devem ser coletadas.

Cada algoritmo de escalonamento possui seus objetivos. Anexadas a eles, devem estar presentes as métricas necessárias para que o escalonamento seja avaliado.

Mesmo que tenha um objetivo fixado, um algoritmo de escalonamento pode apresentar desempenhos distintos. Dessa forma, torna-se extremamente difícil afirmar que exista um único algoritmo capaz de produzir resultados excelentes em todas as combinações possíveis de hardware e software. As variações ocorridas no desempenho do algoritmo devem-se principalmente à influência direta de três fatores: a plataforma computacional (principalmente as distribuídas por serem heterogêneas), o algoritmo de escalonamento e as classes de aplicações que estão sendo 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.

O problema do escalonamento tem sido descrito de diferentes maneiras na literatura. Para escalonamento em processos distribuídos, temos uma visão geral da função de escalonamento como um recurso gerenciando outro recurso. Este gerenciamento de recursos é um mecanismo ou uma política para gerenciar de forma efetiva e eficaz o acesso e o uso de recursos pelos seus vários consumidores. Assim, o problema do escalonamento consiste de três componentes principais: o(s) consumidor(es), o(s) recurso(s) e a política adotada.

Para melhor entender o funcionamento de um escalonador, deve-se observar o efeito que ele causa no ambiente, ou seja, observar o comportamento do escalonador em termos de quanto a política por ele adotada afeta os recursos e consumidores. O relacionamento entre o escalonador, as políticas e os recursos é mostrado na figura abaixo:

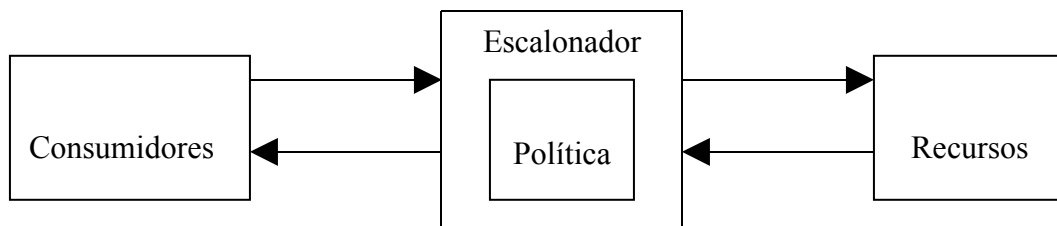


Figura1: Sistema de Escalonamento

Levando-se em conta a descrição da figura acima, os consumidores querem estar aptos a obter acesso rápido e eficiente ao recurso em questão, mas não desejam ser prejudicados pelos problemas de overhead associados ao uso da função de gerenciamento.

3. Taxonomia de Escalonamento de Processos em Sistemas Distribuídos

Diversos autores têm sugerido taxonomias para a área de escalonamento de processos. Dentre elas, destaca-se a taxonomia de Casavant, por ser a mais abrangente e por ter melhor aceitação.

A meta da taxonomia de Casavant é fornecer um mecanismo que permita a comparação de trabalhos anteriores na área de escalonamento distribuído, de uma forma qualitativa. Além disso, se espera que as categorias e os relacionamentos entre elas sejam escolhidos de forma cuidadosa suficiente para ajudar em trabalhos de classificações futuras.

Na verdade, essa taxonomia pode ser empregada para classificar qualquer conjunto de recursos de gerenciamento de sistemas. No entanto, nossa atenção será dada ao gerenciamento de processos, ao passo que é nessa área que nós esperamos encontrar relações úteis para serem utilizadas em trabalhos futuros.

Essa estratégia de escalonamento é feita com base: no tipo de informações usadas para que as tarefas sejam escalonadas; onde as tarefas serão alocadas quando efetuado o re-escalonamento; e onde e de que forma são efetuadas as tomadas de decisão e a obtenção de informações.

3.1. Classificação Hierárquica

A estrutura da classificação hierárquica dessa taxonomia é mostrada na figura abaixo. Depois, segue uma discussão sobre as porções hierárquicas dessa classificação.

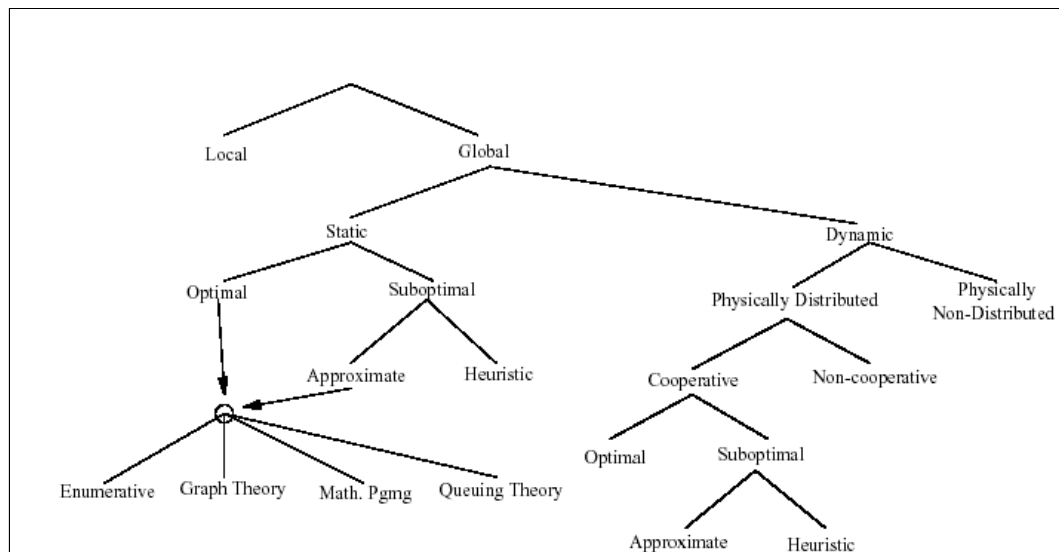


Figura 2: Classificação Hierárquica de Casavant

Local X Global

No nível mais alto, devemos diferenciar entre escalonamento global e local. O escalonamento local está envolvido com a associação de processos a “fatias” de tempo de um único processador. O escalonamento global deve decidir em qual elemento o processo deve ser executado, em um sistema multiprocessador, e o trabalho de escalonamento local é deixado para o sistema operacional do processador no qual o processo é, no final das contas, alocado. Isso permite que os processadores em um multiprocessador aumentem sua autonomia enquanto reduzem a responsabilidade, e, conseqüentemente o overhead do mecanismo de escalonamento global.

Estático X Dinâmico

O próximo nível da hierarquia, continuando através da escolha de escalonamento global, é a escolha entre escalonamento estático e dinâmico, que indica o momento no qual o escalonamento é feito.

No caso de escalonamento estático, este é feito 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 conhecimento prévio sobre os tempos de execução das tarefas e sobre os recursos de processamento.

Assim, cada imagem executável no sistema tem uma associação estática para um processador em particular, e em cada momento que a imagem do processo é submetida a execução, ela é associada com aquele processador. Entretanto, métodos desse tipo não admitem preempção.

No caso de escalonamento dinâmico, a decisão de escalonamento é tomada em tempo de execução, para considerar o estado corrente do sistema e as necessidades dinâmicas específicas de uma aplicação.

Embora a técnica estática funcione quando pode se predizer facilmente os recursos necessários à aplicação, a execução dinâmica requer decisões de alocação *runtime* para uma execução eficiente. Políticas de alocação dinâmica alcançam essa velocidade, geralmente negligenciando o impacto de padrões de comunicação entre aplicações e restringindo sua

atenção para recursos de execução. Portanto, o objetivo da política de alocação dinâmica é o balanceamento da carga, ou seja, balancear a carga dos nós do sistema.

Ótimo X Sub-Ótimo

No caso de todas as informações estimarem o estado do sistema, e os recursos necessários para um processo forem conhecidos, uma associação ótima pode ser feita baseada em alguns critérios. Exemplos de medidas de otimização são: a minimização do tempo total de um processo e a maximização de utilização de recursos no sistema, ou maximização do throughput do sistema. Nos eventos em que estes problemas são computacionalmente impraticáveis, soluções sub-ótimas devem ser tentadas.

Com a escolha de soluções sub-ótimas para o problema do escalonamento, devem ser analisadas mais duas categorias de escalonamento : Aproximado e Heurístico.

Aproximado X Heurístico

O primeiro método utiliza o modelo computacional formal de algoritmo, mas ao invés de procurar em todo o espaço de soluções por uma solução ótima, se satisfaz ao encontrar um solução “boa”. Essas soluções são categorizadas como sub-ótimas aproximadas.

Em casos onde uma métrica está disponível para avaliar uma solução, esta técnica pode ser utilizada para diminuir o tempo necessário para encontrar uma solução aceitável.

Os fatores que determinam se essa aproximação é digna de registro incluem:

- A disponibilidade de uma função para avaliar a solução;
- O tempo requerido para avaliar uma solução;
- A habilidade para julgar de acordo como alguma métrica a validade de uma solução ótima;
- A disponibilidade de um mecanismo para encontrar o espaço de soluções inteligentemente.

O segundo método abaixo da categoria sub-ótima é chamado de heurístico, e representa a categoria de algoritmos estáticos que fazem as suposições mais realistas sobre conhecimentos relativos a processos e características de carga de sistema. Além disso,

representa as soluções para o problema de escalonamento estático que requerem a maior parcela de tempo e outros recursos do sistema para executar suas funções.

A característica mais importante de escalonadores heurísticos é que eles utilizam parâmetros que afetam o sistema através de meios indiretos. Por exemplo, o agrupamento de grupos de processadores que se comunicam muito entre si e a separação física de processos que se beneficiarão do paralelismo diminui diretamente o *overhead* envolvido na passagem de informações entre processadores, enquanto reduz a interferência entre processos que devem executar sem sincronização com outros. Esse resultado tem um impacto no serviço global que os usuários recebem, mas não pode ser diretamente relacionado à performance do sistema da maneira como o usuário o vê. Por consequência, intuitivamente acreditamos que em ações como a mencionada acima, o desempenho do sistema será melhorado quando possível. Porém, não podemos provar que uma relação de primeira ordem exista entre o mecanismo empregado e os resultados desejados.

Técnicas Ótimas e Sub-Ótimas Aproximadas

Sendo uma solução ótima ou sub-ótima aproximada, existem quatro categorias básicas de algoritmos de alocação de tarefas que podem ser usadas para uma associação entre processos e processadores.

- Índice do espaço de soluções e busca;
- Teoria dos Grafos;
- Programação Matemática
- Teoria do Enfileiramento

Soluções dinâmicas

Em problemas de escalonamento dinâmico, assume-se que muito pouco é conhecido dos recursos necessários para se executar os processos, bem como o ambiente o processo irá executar durante seu tempo de vida. Assim, nenhuma decisão é feita até que o processo inicie efetivamente a sua execução.

Distribuído X não-distribuído

A categoria dinâmica pode ser dividida em soluções onde a responsabilidade do escalonamento global dinâmico reside em uma única entidade, ou seja, fisicamente não distribuída, ou onde o trabalho envolvido no processo de decisão é fisicamente distribuído entre todas as entidades.

Cooperativos X não-cooperativos:

Dentre o grupo de técnicas de escalonamento global dinâmico e distribuído, pode-se distinguir entre os mecanismos que envolvem cooperação entre os componentes distribuídos (cooperativo) e aqueles em que entidades individuais fazem decisões independente das ações de outras entidades (não cooperativo).

A questão é qual o grau de autonomia que cada entidade tem para determinar se seus próprios recursos devem ser usados. No caso não-cooperativo, entidades individuais atuam sozinhas como entidades autônomas e chegam a decisões independente das decisões do restante do sistema. No caso cooperativo, cada elemento de processamento tem a responsabilidade de analisar sua própria porção da tarefa de escalonamento, mas todos as entidades trabalham em direção a um objetivo comum e global.

3.2. Outros Mecanismos Relacionados

Adaptável X não-adaptável:

Uma solução adaptável para o problema de escalonamento ocorre quando os algoritmos e os parâmetros usados para se implementar a política de escalonamento mudam dinamicamente de acordo com o comportamento atual e anterior do sistema, em resposta a decisões anteriores feitas pelo sistema de escalonamento. Em resposta ao comportamento do sistema, o escalonador pode começar a ignorar um parâmetro ou reduzir a importância daquele parâmetro se ele acreditar que o parâmetro está fornecendo uma informação que é não é consistente com o restante das entradas ou não está fornecendo nenhuma informação relevante para o estado do sistema.

O termos “adaptável” e “dinâmico” abrigam conceitos distintos..

Numa situação dinâmica, o escalonador leva em conta o estado atual à medida que tal é detectado no sistema. Isto é feito durante a operação normal do sistema sob uma carga dinâmica e imprevisível.

No sistema adaptável, a política de escalonamento reflete as mudanças no seu ambiente por si só. Apesar da solução dinâmica levar em conta entradas do ambiente para efetuar suas decisões, uma solução adaptável leva em conta essas mesmas variáveis para modificar a própria política de escalonamento.

Probabilístico

Em muitos problemas de associação, o número de permutações de tarefas, e o número de mapeamento a elementos de processamento é tão grande, que a análise analítica poderia requerer um montante proibitivo de tempo. Assim, a idéia de escolha aleatória repetitiva, de acordo com alguma distribuição conhecida, de um conjunto de possíveis escalonamentos, pode produzir um resultado satisfatório.

Preemptivo X não-preemptivo:

No escalonamento não preemptivo, as tarefas estarão disponíveis para re-alocação somente antes do início de execução. No escalonamento preemptivo, ao contrário, tarefas podem ser re-allocadas após o início de sua execução.

4. Balanceamento de Carga

O objetivo do balanceamento de carga é otimizar a saída (*throughput*) do sistema todo e assim, estabilizar o tempo de resposta. Para que isto seja possível, ele tem de coletar os estados do sistema, informações sobre o progresso do mesmo e possivelmente explorar previsões de cargas de tarefas.

Baseado nessas informações, é preciso também efetivar a designação de tarefas e decisões de migrações de tarefas - que ocorrem quando uma tarefa pode ser re-allocada a outro elemento de processamento mesmo após o início de sua execução -, que produzem uma carga igual no processamento de nós e minimizam a sincronização e atrasos na comunicação de dados entre tarefas inter-relacionadas.

Duas políticas citadas anteriormente são componentes importantes em um algoritmo de escalonamento global: a política de transferência e a política de localização.

Como já foi dito, a política de transferência em um nó determina se uma tarefa deve ser transferida de ou para este nó, identificando o nó como um remetente (transmissor) apropriado, um receptor apropriado, ou nenhum dos dois.

O tamanho da tarefa pode ser quantificado como o tempo de execução da tarefa, possuindo os seguintes parâmetros:

- Carga do host: Um processo pode ser transferido somente se a carga do host é maior que um certo limiar.
- Tamanho do processo: Após feita a checagem se um processo é maior que um certo limiar, verifica-se se é vantajoso a transferência, levando em conta o seu *overhead*.
- Necessidades de recursos do processo: uma vez que um processo pode não ser executado em todas as máquinas possíveis ou o seu tempo de

execução possa variar, torna-se necessário especificar claramente este requisito.

A política de localização determina o elemento de processamento para o qual uma tarefa deve se associar, e a seleção é baseada em uma carga mínima do elemento de processamento ou no primeiro elemento cuja a carga seja menor que um certo limiar. Uma medida da carga de um elemento de processamento pode se basear no tamanho da fila de tarefas prontas deste elemento.

As políticas de localização podem ser caracterizadas como *envio-iniciadas*, na qual os remetentes procuram por receptores apropriados; *recepção-iniciadas*, na qual receptores potenciais procuram por remetentes; e *simetricamente-iniciadas*, na qual ambos remetentes e receptores procuram por nós que lhes sejam complementares.

Para cargas baixas até as moderadas, as buscas envio-iniciadas produzem um desempenho melhor que as buscas recepção-iniciadas, desde que para tais cargas, remetentes em potencial possam encontrar nós sem muita carga, para onde possam transferir suas tarefas antes que fiquem sobrecarregados. Na busca recepção-iniciada, o nó sobrecarregado deve esperar por um contato de um nó não carregado, e isso pode levar a uma perda de tempo significativa. No entanto, quando as cargas das tarefas que chegam são altas, a busca recepção-iniciada apresenta um desempenho melhor do que as envio-iniciadas, desde que nós menos sobrecarregados sejam capazes de quase que imediatamente encontrar nós sobrecarregados dos quais deverão obter tarefas.

Algoritmos de balanceamento de carga podem ainda ser classificados como *Estáticos ou Dinâmicos*. No primeiro, as políticas de transferência e localização baseiam-se em informações médias do comportamento do sistema e são independentes do estado corrente do sistema no momento da tomada de decisão. Políticas de localização aleatórias e *round-robin* são exemplos da categoria estática.

No segundo, reagem às mudanças do estado do sistema. A complexidade desses métodos é um ponto negativo no sentido de exigirem informação em tempo real da carga do sistema e do estados das tarefas associadas. O escalonamento baseado em limiar é um exemplo desta segunda categoria.

4. Conclusão

Através desse trabalho, foi possível notar a importância do uso do escalonamento de processos em sistemas distribuídos, que mostram claramente a vantagem de se utilizar sistemas computacionais distribuídos e máquinas paralelas sobre sistemas monoprocessadores. Para que essas vantagens sejam utilizadas, algoritmos foram propostos para controlar da melhor forma possível o compartilhamento dos recursos existentes nesses tipos de sistemas.

A taxonomia de Casavant possibilita, de uma forma qualitativa, a comparação de trabalhos anteriores na área de gerenciamento de recursos, e foi estudada por ser a mais abrangente e mais aceita.

Os algoritmos de balanceamento de carga possibilitam um ganho no desempenho do sistema, pois a otimização e a eficácia do sistema estão diretamente relacionadas com a melhor utilização da capacidade de processamento existente. Assim, 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.

5. Referências

T. L. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems", IEEE Trans. Software Eng., vol. 14, no. 2, pp 141-154, Feb.1988.

COULOURIS, G., DOLLIMORE, J., KINDBERG, T. Distributed Systems Concepts and Design. Segunda Edição, Addison-Wesley, 1994.

TANENBAUM, A.S. *Modern Operating Systems*. New Jersey, Prentice Hall International, Inc., 1992.

<http://www.ic.unicamp.br/~ra951407/taskschd.pdf>

http://lasdpc.icmc.sc.usp.br/pesquisa/kalinka/quali_kalinka.doc