
GERÊNCIA DE RECURSOS

Escalonamento Global

Gerência de recursos - escalonamento global

⌘ Além de prover comunicação, recursos de acesso a rede, memória compartilhada, sistemas de arquivos distribuídos, um sistema operacional distribuído tem que poder executar um processo em um processador que se encontra em um nó remoto, ou migrar um processo de um nó para outro pois:

- ☒ pode ser necessário desligar o nó
- ☒ nó não possui algum recurso necessário ao processo
- ☒ deseja-se melhorar o tempo de execução geral do sistema

Gerência de recursos - escalonamento global

⌘ Os algoritmos de alocação do processador devem levar em conta requisitos como:

- ☒ tempo de resposta
- ☒ carga na rede
- ☒ *overhead*
- ☒ maximização do uso do processador

Gerência de recursos - escalonamento global

⌘ Três técnicas básicas para alocação do processador:

☒ atribuição de tarefas (*task assignment*)

- objetiva melhorar a performance de execução de um processo (composto por tarefas)

☒ balanceamento de carga (*load balancing*)

- objetiva distribuir a carga dos nós igualmente

☒ compartilhamento de carga (*load sharing*)

- garantir que nenhum nó fique ocioso enquanto algum processo espera para ser executado

Gerência de recursos - escalonamento global

⌘ Característica desejáveis:

- ☒ sem conhecimento prévio a respeito dos processos
 - usuários teriam de especificar perfil do processo ...
- ☒ Dinâmico
 - considera carga variável ao longo do tempo
- ☒ rápido na tomada de decisão
 - uso de heurísticas ao invés de soluções exaustivas
- ☒ performance equilibrada e pouco *overhead* no algoritmo de *scheduling*
 - dilema: muita informação para construir estado global - recolhimento e processamento da informação -> overhead
 - “validade” temporal da informação
 - baixa frequência de escalonamento pois custa muito caro
 - preferível: procedimento perto do ótimo, com tráfego mínimo para formação do estado global necessário

Gerência de recursos - escalonamento global

⌘ Característica desejáveis:

☑ estável

- evitar migrações desnecessárias de um processo

☑ escalável

☑ tolerante a falhas

☑ Justo

⌘ Lembra o que? ... Sistema Operacional

Gerência de recursos - escalonamento global - atribuição

⌘ Atribuição de tarefas

⌘ Cada processo é dividido em tarefas

⌘ Meta: encontrar a melhor forma de distribuir estas tarefas para um determinado processo

⌘ Considerações:

⏏ tempo de execução necessário para cada tarefa e velocidade dos processadores *são conhecidos*

⏏ custo para executar tarefa em um processador *é conhecido*

⏏ custo de comunicação entre processos *é conhecido*

- custo=0 se tarefas estão no mesmo processador

Gerência de recursos - escalonamento global - atribuição

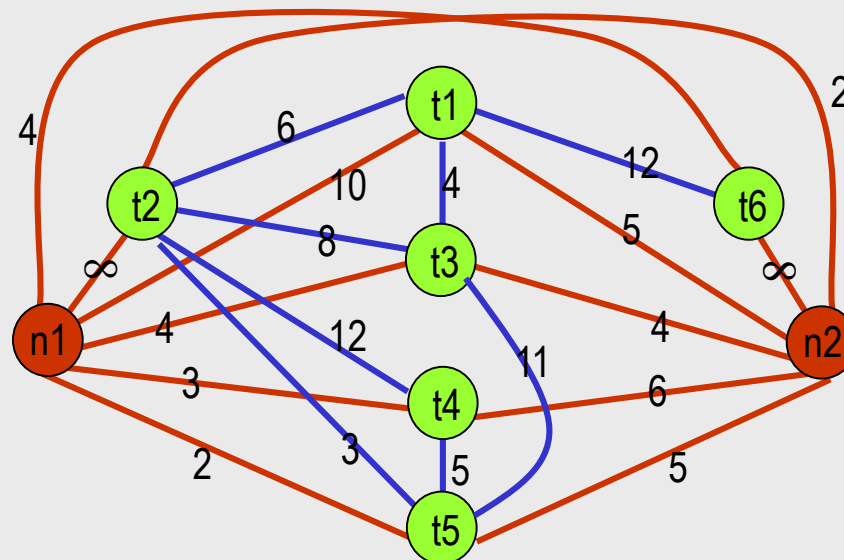
⌘ Baseado nas considerações anteriores busca-se:

- ☒ minimizar os custos de comunicação entre processos
- ☒ melhorar performance por processo executado
- ☒ aumentar paralelismo
- ☒ utilizar de maneira mais eficiente os recursos do sistema

Gerência de recursos - escalonamento global - atribuição

⌘ Algoritmo para encontrar solução ótima em dois processadores

- ⏏ usa grafo
- ⏏ nós representam tarefas e processadores
- ⏏ arco entre tarefas representam custo de comunicação entre tarefas
- ⏏ arco entre tarefa e processador representa o custo de execução da tarefa no processador - infinito significa impossibilidade de executar tarefa no nodo



Problemas deste algoritmo

- ⌘ Não é escalável (NP Hard)
- ⌘ Considerou as duas grandezas equivalentes em custo
 - ⌘ Comunicação e execução, pode não ser
 - Comunicação é rede, execução processador
 - ⌘ Como combinar?
- ⌘ Assume que tenho todas as informações antes de rodar
 - ⌘ Muitas vezes não tenho e não consigo estimar
- ⌘ O que fazer?
 - ⌘ Diminuir exigência

Gerência de recursos - escalonamento global - balanceamento

⌘ Balanceamento de carga

- ⌘ Baseado na intuição de que para melhor utilizar os recursos do sistema, a carga deve estar distribuída igualmente no sistema
- ⌘ Meta: ter a “mesma” carga em todos os nós do sistema, transferindo processos de um nó sobrecarregado para um nó subcarregado

Gerência de recursos - escalonamento global - balanceamento

⌘ Considerações em algoritmos de balanceamento de carga

⏏ Política de estimativa de carga:

- como se mede a carga de um nodo ?

⏏ Política de transferência de processos

- quando um nodo está sobrecarregado ?

⏏ Política de seleção de novo nó

- para qual nodo será transferido o processo ?

⏏ Política de troca de informações

- como são trocadas as informações de estado entre os nodos ?

⏏ Política de prioridades

- que prioridade processos locais e remotos assumem para o escalonamento local ?

⏏ Política de limitação de migrações

- quantas vezes um processo pode ser migrado ?

Gerência de recursos - escalonamento global - balanceamento

⌘ Política de estimativa de carga

☒ Em sistemas operacionais distribuídos modernos

- muitos daemons rodam
- acordam periodicamente para executar



- => *número de processos não é significativo*
- => *tempo para término não é válido para processos permanentes*
- *estimativas anteriores não são boas!!!*

- Medida aceitável: utilização da CPU, mas também outros recursos memória, rede ...
- Posso usar técnicas de regressão linear e Machine Learning

Gerência de recursos - escalonamento global - balanceamento

⌘ Política de troca de informações

- ⏏ **como obter a informação necessária de outros nodos para efetuar balanceamento ?**
- ⏏ **Dilema:** sobrecarga da rede e processamento vs. Melhoria no balanceamento
- ⏏ *broadcast* periódico:
 - informa estado em intervalos de tempo t para todos nodos da rede
 - Técnica de heart-beat (batida de coração)
- ⏏ *broadcast* só quando o estado muda:
 - informa quando muda estado -> subcarregado, normal, sobrecarregado

Gerência de recursos - escalonamento global - compartilhamento

⌘ Compartilhamento de Carga

- ⏏ muitos pesquisadores acreditam que balanceamento de carga não tem objetivo apropriado
 - overhead na coleta de informações sobre estado do sistema é normalmente alto, crescendo com tamanho do sistema distribuído
 - balanceamento estrito não é alcançável pois diferenças momentâneas de carga sempre existem entre nodos
- ⏏ novo objetivo: evitar que nodos do sistema fiquem ociosos enquanto outros nodos tem trabalho a espera
- ⏏ Mais uma diminuição de exigência em prol da redução de custo!

Gerência de recursos - escalonamento global - compartilhamento

- ⌘ Evitar que algum nodo fique ocioso enquanto outros nodos tem mais trabalho a espera
 - ☒ tipicamente: nodo com mais de dois processos
 - ☒ necessita mesmas políticas de balanceamento de carga
 - ☒ no entanto, é muito mais simples decidir que política usar para cada caso pois a tentativa não é manter uma carga média em todos nodos do sistema mas garantir que um nodo não está ocioso enquanto trabalho existe a espera em outro nodo

Gerência de recursos - escalonamento global - compartilhamento

⌘ Política de estimativa de carga

- é suficiente saber se nodo está ocupado ou livre
- medida de utilização de CPU

⌘ Política de transferência

- uso de três níveis

⌘ Política de seleção de nó

⏏ Quem tem a iniciativa?

- sender initiated: nó sobrecarregado procura nó livre
- receiver initiated: nó livre procura sobrecarregado
- em ambos: uso de broadcast ou prova randômica

⏏ Melhor caso vai depender da carga geral do sistema?

Gerência de recursos - escalonamento global - compartilhamento

⌘ Política de troca de informações

⏏ só age quando estado muda

- broadcast quando estado muda
- polling quando estado muda, para alguns nós escolhidos randomicamente
 - por causa baixa escalabilidade de broadcast