

GERÊNCIA DE RECURSOS Escalonamento Global

Sistemas Distribuídos

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

Sistemas Distribuídos

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

Sistemas Distribuídos

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

Sistemas Distribuídos

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

Sistemas Distribuídos

212

Gerência de recursos - escalonamento global

- **Característica desejáveis:**
 - Estável
 - evitar migrações desnecessárias de um processo
 - Escalável
 - Tolerante a falhas
 - Justo

Sistemas Distribuídos

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:
 - cada processo está dividido em tarefas
 - 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

Sistemas Distribuídos

Gerência de recursos - escalonamento global - atribuição

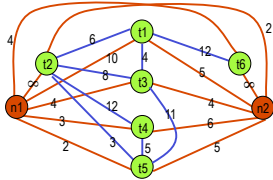
- Baseado nas considerações anteriores busca-se:
 - minimizar os custos de comunicação entre processos
 - melhorar performance por processo executado
 - aumentar paralelismo
 - utilizar de maneira mais eficiente os recursos do sistema

Sistemas Distribuídos

215

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

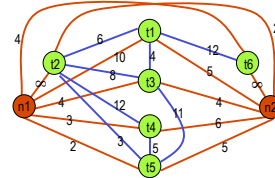


Sistemas Distribuídos

Gerência de recursos - escalonamento global - atribuição

- Algoritmo para encontrar solução ótima em dois processadores
 - problema: achar o **Corte de custo mínimo**
 - custo do corte é a soma dos custos das arestas cortadas

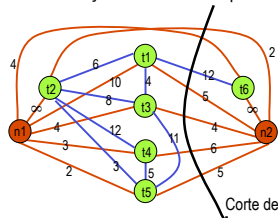
exercício: calcule o valor de alguns cortes



Sistemas Distribuídos

Gerência de recursos - escalonamento global - atribuição

- Algoritmo para encontrar solução ótima em dois processadores



Corte de custo mínimo

- E para $n > 2$?
 - para $n > 2$ (número de processadores), alg. é Não-Polinomial Difícil (NP Hard)
 - computacionalmente intratável

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

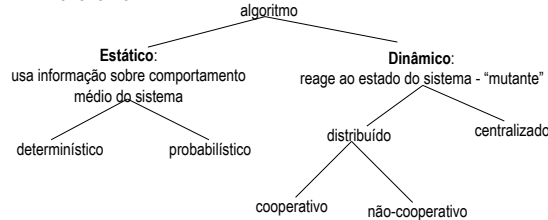
- **Balanceamento de carga**

- 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

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Taxonomia



Sistemas Distribuídos

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 nó ?
- Política de transferência de processos
 - quando um nó está sobrecarregado ?
- Política de seleção de novo nó
 - para qual nó será transferido o processo ?
- Política de troca de informações
 - como são trocadas as informações de estado entre os nós ?
- 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 ?

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de estimativa de carga

- **Como medir a carga de um nó ?**
 - número de processos no nó (vários algoritmos usam este)
 - demanda de recursos dos processos
 - arquitetura e velocidade dos processadores
- vários autores consideram que precisa de estimativa "para frente"
 - como estimar o futuro? - tempo de serviço restante do processo
 - *memoryless* - assume mesmo tempo esperado para término para todos processos
 - *past-repeats* - assume que tempo para término = tempo até aqui
 - *distribution* - usa distribuição de tempos de processos (taxas de incidências/probabilidade de tempos de processos)

Sistemas Distribuídos

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

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de transferência de processos

- *load-balancing*: transferir processos de nós carregados para nós leves
- **como decidir se nó está sobrecarregado ?**
- **usando limite**
 - processo novo é aceito localmente se nó está pouco carregado (abaixo do limite)
 - se nó acima do limite, tentativa de transferência do processo para outro nó
- limite pode ser
 - estático - pré-definido
 - dinâmico - valor limite calculado como média da carga do sistema troca de informações sobre estado

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de transferência de processos

- **usando limite (cont.)**
 - problema: nó abaixo do limite aceita processo e fica acima do limite ...
 - além disso: migração tem que valer a pena
- **três níveis (sobrecarregado, normal, subcarregado)**
 - estado do nó varia entre estes estados dinamicamente
 - decisão sobre transferir processo local ou aceitar processo remoto:
 - se sobrecarregado: processos locais novos são mandados para outros nós e pedidos de execução de processos remotos são negados
 - se normal: processos novos executados localmente e pedidos de execução de processos remotos são rejeitados
 - se subcarregado: aceita processos novos locais e pedidos de execução de processos remotos

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de seleção de um novo nó

- uma vez que se sabe que um processo deve ser transferido (política de transferência de processos) **para qual nó será transferido o processo ?**
- métodos
 - limite:
 - escolhe nó randomicamente e testa se transferência tomaria o receptor acima de um limite que o proíba de receber processos remotos
 - continua até exceder um número limite de testes, e executa localmente
 - resultados de simulação mostram que testar poucos ou muitos nós impacta muito pouco no desempenho geral do sistema
 - mais leve:
 - vários nós distintos são escolhidos randomicamente, cada um é perguntado sobre seu estado
 - processo é transferido ao nó de menor carga, ou executado no originador se nenhum pode receber
 - simulação não mostra significantes melhorias com relação a "limite"

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de seleção de um novo nó

- Métodos (cont.)
 - lance:
 - economia computacional - atores/abstrações: compradores e vendedores
 - *manager* - tem processo e precisa local para executar
 - *contractor* - representa nó pronto para aceitar processos remotos
 - procedimento:
 - *manager* broadcasts *request for bids* (pedido de lances)
 - *contractors* respondem lances (memória, CPU, etc.)
 - *manager* escolhe lance melhor e move processo
 - e se *contractor* ganha vários contratos simultaneamente:
 - *manager* manda mensagem ao ganhador (*contractor*)
 - *contractor* pode rejeitar contrato por ter mudado de estado
 - *manager* pode ter que recomençar o "leilão"

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de seleção de um novo nó

- Métodos (cont.)
 - pares:
 - outras políticas: reduzir discrepâncias de carga entre todos o nós do sistema
 - este: reduzir a diferença entre pares de nós
 - dois nós que, *temporariamente*, tem diferença grande de cargas são *pareados* e processos são migrados de um para outro
 - vários pares podem existir no sistema simultaneamente
 - nó inicia "pareamento" só se tem mais de dois processos (senão migração não vale a pena)
 - nós estão sempre dispostos a aceitar "pareamento", atendendo aos outros

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de seleção de um novo nó

- Métodos (cont.)
 - pares:
 - procedimento:
 - nó seleciona outro randomicamente para pareamento e manda *request*
 - enquanto não tem resposta, nega pedidos de pareamento que receba
 - se receber negação, seleciona outro nó para fazer *request*
 - se receber ok, forma par
 - durante pareamento, não aceita outros pedidos
 - transfere processos com base em estimativa de tempo de término de processo no par destino ser menor que tempo de término do processo no par origem + tempo de migração
 - migração dos próximos processos levam em conta processo já migrado/escolhido para migração

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de troca de informações

- Como obter a informação necessária de outros nós para efetuar balanceamento ?
- Dilema: sobrecarga da rede e processamento vs. melhoria no balanceamento
- broadcast periódico:
 - informa estado em intervalos de tempo t para todos nós da rede
- broadcast quando o estado muda:
 - informa quando muda estado -> subcarregado, normal, sobrecarregado

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de troca de informações

- broadcast quando é necessário (sob demanda)
 - nó precisa de informações de outros quando sobrecarregado (manda) ou subcarregado (aceita processos)
 - nó broadcasts *StateInformationRequest* e informa seu estado
 - outros nós respondem: respostas podem ser só de nós sobrecarregados ou só de nós subcarregados
- troca de informações através de *polling*
 - uso de broadcast - baixa escalabilidade
 - premissa: não há necessidade de trocar informações de estado com todos nós, mas somente com alguns
 - pode fazer *polling* a nós randomicamente escolhidos (similaridade com política de pares na escolha de nó)

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de prioridade

- que prioridades processos locais ou remotos assumem no procedimento de escalonamento local de um nó ?
- egoísta
 - maior prioridade para processos locais, prioridade muito baixa para remotos
 - pior resultado global
- altruísta
 - maior prioridade para processos remotos
 - melhor resultado global
- intermediário
 - processos locais tratados pouco melhor que remotos
 - desempenho intermediário, mas muito próximo do caso altruísta

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - balanceamento

➤ Política de migração

- quantas vezes um processo pode ser migrado ?
- não-controlada:
 - processo que migra para um nó é tratado como qualquer outro processo, podendo ser migrado novamente
 - pode levar a instabilidade
- controlada
 - contador de migrações de um processo
 - muitos autores defendem que não deve passar de 1 (!)
 - alguns autores defendem que processos longos podem ter que ser migrados mais de uma vez

Faculdade de Informática - PUCRS

Sistemas Distribuídos

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 nós
- novo objetivo: evitar que nós do sistema fiquem ociosos enquanto outros nós tem trabalho a espera

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - compartilhamento

➤ Evitar que algum nó fique ocioso enquanto outros nós tem mais trabalho a espera

- tipicamente: nó 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 nós do sistema mas garantir que um nó não está ocioso enquanto trabalho existe a espera em outro nó

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - compartilhamento

➤ Política de estimativa de carga

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

➤ Política de transferência

- uso de três níveis

➤ Política de seleção de nó

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

Faculdade de Informática - PUCRS

Sistemas Distribuídos

Gerência de recursos - escalonamento global - compartilhamento

➤ Política de troca de informações

- só age quando estado muda
- broadcast quando estado muda
- polling quando estado muda (baixa escalabilidade de broadcast)

Faculdade de Informática - PUCRS

Sistemas Distribuídos