

# **Sistemas Computacionais Distribuídos**

**Prof. Marcos José Santana  
SSC-ICMC-USP**

**São Carlos, 2006**

# **Grupo de Sistemas Distribuídos e Programação Concorrente**

**Departamento de Sistemas  
de Computação - SSC**

# **Sistemas Computacionais Distribuídos**

**14a. Aula**

## **Escalonamento de Processos em Sistemas Distribuídos**

# Conteúdo

---

- Introdução
- Classificação dos Métodos de Escalonamento
  - Classificação Hierárquica
  - Classificação Plana
- Migração de Processos
  - Mecanismos e Políticas
  - Transparência, Custo e Arquiteturas Heterogêneas

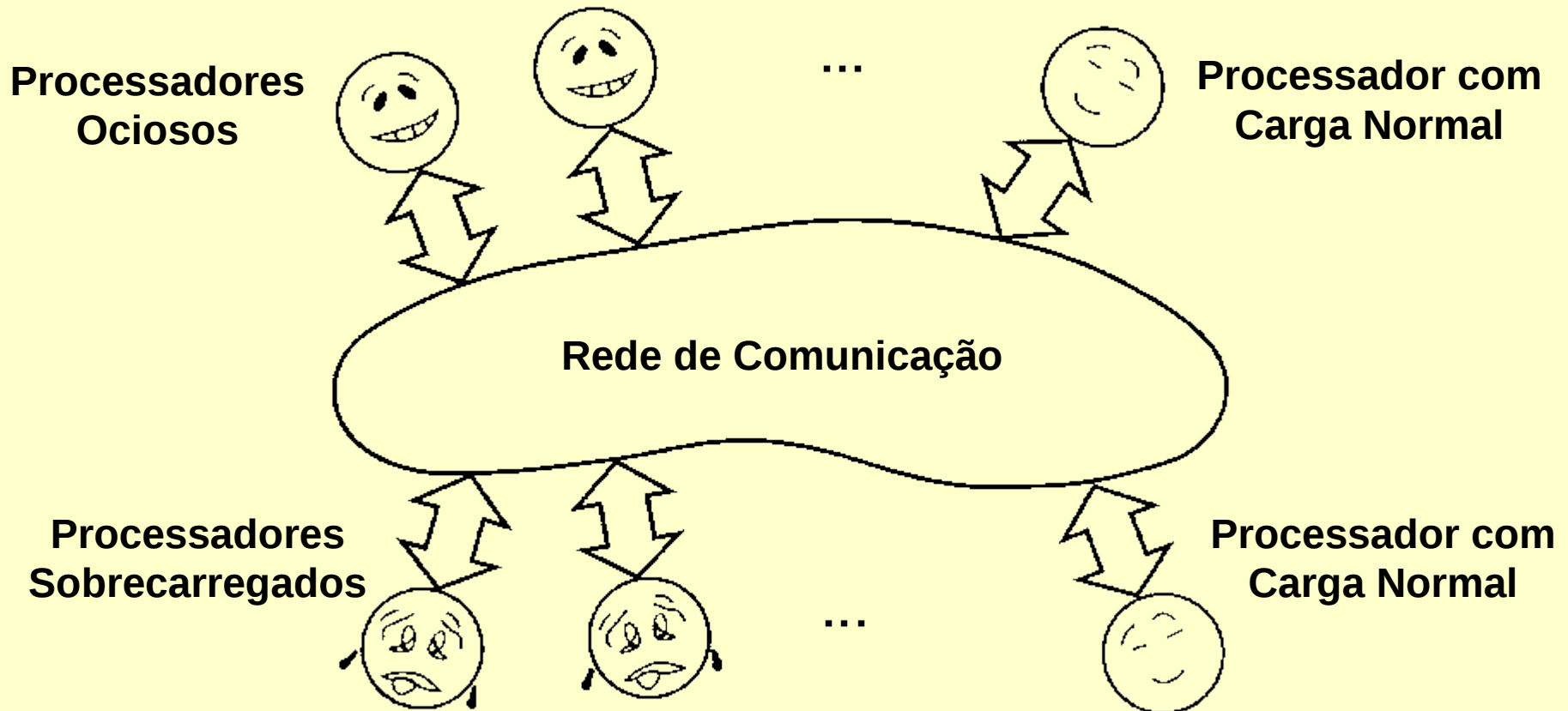
# Conteúdo

---

- Índices de Carga
- Algoritmos de Balanceamento de Carga
  - Classificação de Shivaratri
  - Classificação de Lüling
- Exemplos de Sistemas
  - *Sprite, Condor e Charlotte*
- Conclusões

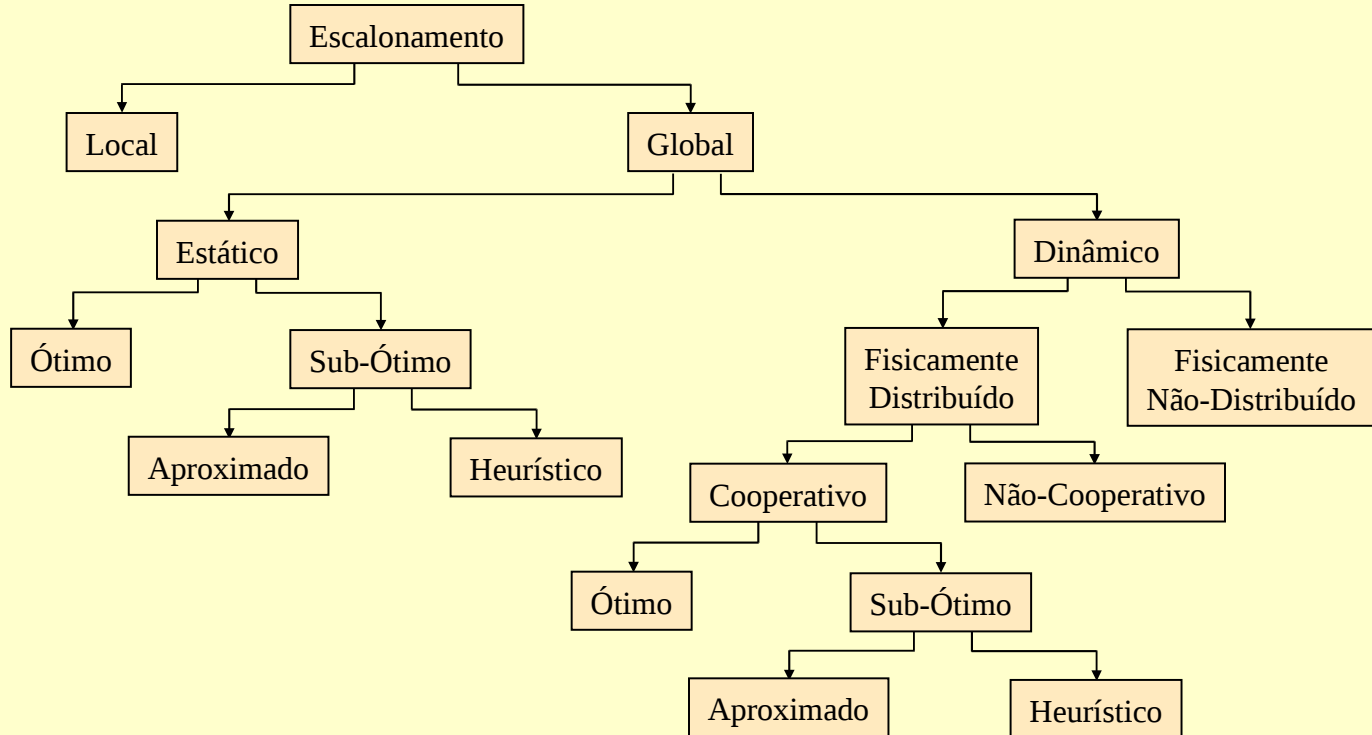
# Introdução

## ➤ Balanceamento de Carga



# Classificação dos Métodos de Escalonamento

## ➤ Classificação Hierárquica



# Classificação dos Métodos de Escalonamento

---

- Escalonamento Local
  - Distribuir processos em um único processador (*time-slice*)
- Escalonamento Global
  - Ambiente multiprocessado
    - Decidir em qual Elemento de Processamento (EP) um processo deve ser executado
      - Decisão centralizada
      - Decisão distribuída



# Classificação dos Métodos de Escalonamento

- Escalonamento Global (cont.)
  - Escalonamento Estático
    - Decisão de escalonamento em tempo de compilação
  - Vantagem
    - Minimiza o tempo na execução do programa e os atrasos na comunicação
  - Desvantagens
    - Estimar, durante a compilação, os tempos de execução de uma tarefa e os atrasos na comunicação
    - Os métodos ignoram a distribuição dos dados
    - Dificuldade em obter o perfil de execução de um programa em uma dada arquitetura

# Classificação dos Métodos de Escalonamento

- Escalonamento Global (cont.)
  - Escalonamento Dinâmico  
(Balanceamento de Carga Dinâmico)
    - Decisão de escalonamento em tempo de execução
    - Vantagem
      - Flexibilidade de adaptação às mudanças de carga
    - Desvantagem
      - Custo adicional de processamento
        - Escolha do processo e processador
        - Atrasos de comunicação na realocação da tarefa
  - Decisão de Balanceamento
    - Escalonamento Fisicamente Distribuído
    - Escalonamento Fisicamente Não-Distribuído

# Classificação dos Métodos de Escalonamento

---

- Classificação Plana
  - Sistemas Adaptativos
    - Algoritmos adaptam-se dinamicamente
  - Sistemas Não-Adaptativos
    - Mesmo comportamento

# Classificação dos Métodos de Escalonamento

- Sistemas Preemptivos (Migração de Processos)
  - Tarefas podem ser realocadas após terem iniciado sua execução
  - Difíceis de serem implementados
    - Suspende a tarefa
    - Reunir e transferir as informações sobre o estado da tarefa
    - Reiniciar a execução da tarefa no EP destino
- Sistemas Não-Preemptivos
  - Tarefas não podem ser realocadas após iniciarem sua execução

# Migração de Processos

*“Migração de Processos é a habilidade de mover a execução de um processo em qualquer momento, de uma determinada máquina (origem) para outra (destino)” [Chen96]*

*“Migração de Processos é a realocação de um programa em execução de uma máquina para outra” [Coulouris94]*

## ➤ Objetivos

- Melhorar o compartilhamento e a utilização dos recursos em sistemas distribuídos
- Facilitar a configuração e tolerância a falhas
- Reduzir o tráfego na rede

# Migração de Processos

- A migração de processos deve considerar
  - Para **onde** e **quando** migrar **quais** processos
  - Como transferir o estado do processo
  - Definir tolerância a falhas
  - Fornecer transparência e confiabilidade
- Sistemas distribuídos fracamente acoplados
  - Estações de trabalho/servidor
  - *Pool* de processadores (multiprocessadores)

# Migração de Processos

---

- Mecanismos de Migração
  - Podem estar implementados no *kernel* ou fora dele
  - Definem como a migração será efetuada
    - Coleta de informações sobre os processadores
    - Transferência de processos
  - Passos da migração
    - Suspende o processo
    - Transferir o estado do processo para o EP destino
    - Reiniciar a execução do processo no EP destino

# Migração de Processos

- Transferência de Memória Virtual
  - Transferência Total da Memória Virtual
    - Problemas
      - Processo não executa durante a transferência
      - Nem todas as regiões de memória são utilizadas
  - Transferência com Pré-Cópia
    - Processo continua executando na máquina origem à medida que as informações vão sendo transmitidas para a máquina destino
    - Problemas
      - Páginas “sujas”
      - Não impede a transferência desnecessária de informações



# Migração de Processos

---

- Transferência com Cópia por Referência
  - Processo e estados são transferidos
  - Páginas da memória virtual são transferidas à medida que são referenciadas
  - Problemas
    - Torna a execução do processo na máquina destino mais lenta
      - Falta de páginas de memória
    - Dependências residuais
      - Problemas na máquina origem afetam a execução do processo

# Migração de Processos

---

- Transferência com Cópia por Referência Melhorada
  - Versão melhorada da cópia por referência
  - Páginas da memória virtual são transferidas para o servidor de arquivos
  - Problema
    - Páginas de memória precisam ser copiadas duas vezes: servidor e máquina destino

# Migração de Processos

- Políticas de Migração
  - Determinam como a migração é conduzida
    - Qual processo deve ser migrado
    - Para onde mover o processo
    - Quando mover o processo
    - Quem é responsável pelas decisões
- Política de Transferência
  - Determina se um EP é um transmissor ou um receptor
  - Decisão tomada em função de valores-limites
    - carga superior a um determinado limite  $\Rightarrow$  transmissor
    - carga inferior a um determinado limite  $\Rightarrow$  receptor

# Migração de Processos

---

- Política de Seleção

- Responsável por selecionar as tarefas que devem migrar

- Fatores a serem considerados

- Sobrecarga causada na transferência deve ser mínima

- Tempo de processamento deve compensar a sobrecarga de transferência

- Número de chamadas do sistema dependentes da localização deve ser mínimo

# Migração de Processos

---

- Política de Localização
  - Responsável por identificar para qual EP a tarefa deve migrar
  - Baseada em duas políticas gerais
    - Carga mínima: menor carga corrente
    - Carga baixa: carga abaixo de um valor limite
  - Descentralizada
    - Base aleatória, histórica ou vizinhos mais próximos
  - Centralizada
    - EP coordenador

# Migração de Processos

---

- Política de Informação
  - Responsável por decidir
    - Quando coletar as informações
    - A partir de onde coletá-las
    - Quais informações coletar
  - Pode ser de três tipos
    - Dirigida por demanda
    - Periódica
    - Dirigida pela mudança de estado

# Migração de Processos

---

- Transparência de Migração
  - Distribuir as tarefas sem afetar as operações dos usuários nem as demais aplicações
- Custo da Migração
  - Custo de selecionar EP destino + custo de transferir o estado do processo + custo de atualizar o ambiente no EP destino
  - Varia em função das características do sistema operacional distribuído

# Migração de Processos

---

- Arquiteturas Heterogêneas
  - Processos devem reiniciar a partir do ponto em que foram interrompidos
  - Impõem custos adicionais à migração
    - Estados dos processos são diferentes
    - Requer a conversão dos dados
- Theimer
  - Método por Recompilação
    - Programa de migração independente da máquina



# Índices de Carga

- Quantificar o conceito de carga
- Principais características
  - Além das necessidades de CPU, considerar operações de E/S e memória dos processos
  - Deve refletir quantitativamente as estimativas qualitativas da carga no *host*
  - Deve ser capaz de prever a carga dos *hosts* em um futuro próximo
  - Deve ser relativamente estável
  - Deve haver um relacionamento entre o índice de carga e a performance do sistema

# Índices de Carga

---

- Possíveis índices de carga
  - Tamanho da fila da CPU
  - Utilização da CPU
  - Tempo de resposta ou tempo de processamento das tarefas
  - Funções agregadas
- Ferrari
  - Estudo comparativo dos índices de carga
    - Melhores índices são os que se baseiam no tamanho da fila da CPU

# Algoritmos de Escalonamento (Balanceamento de Carga)

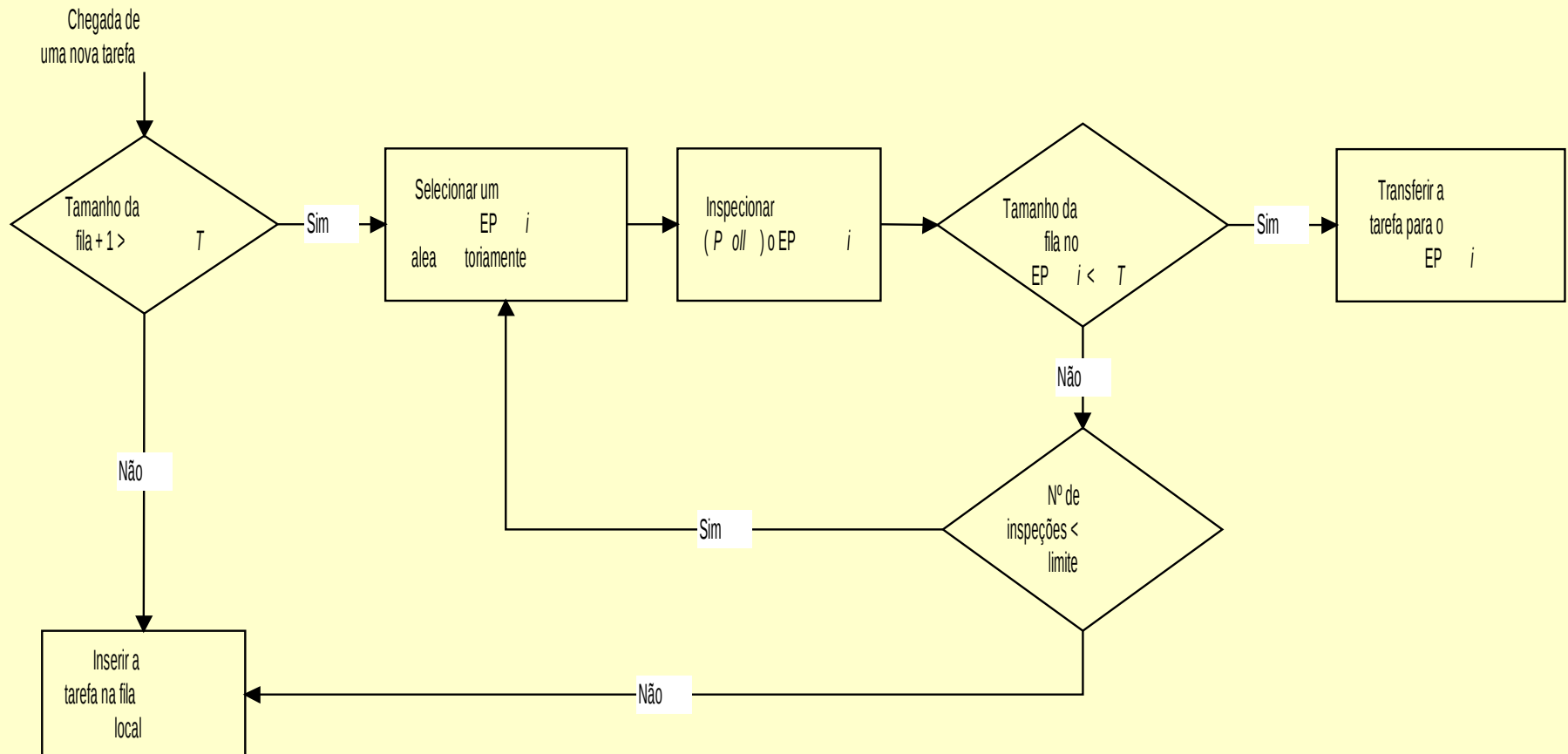
- Classificação de Shivaratri
  - Algoritmos baseados nas políticas
    - Transferência, Seleção, Localização e Informação
- Algoritmos Estáveis
  - Teoria das Filas
    - Instabilidade quando a taxa de chegada de tarefas é maior que a taxa de conclusão
  - Algoritmos de Distribuição
    - Instabilidade quando o algoritmo realiza operações improdutivas indefinidamente
- Algoritmos Eficientes

# Algoritmos de Escalonamento (Balanceamento de Carga)

- Algoritmos Iniciados pelo Transmissor
  - Política de Transferência
    - Baseada no tamanho da fila da CPU
  - Política de Seleção
    - Somente tarefas recentes
  - Política de Localização
    - Aleatória, Limite, Menor Limite
  - Política de Informação
    - Orientada à demanda
- Instáveis em altas cargas

# Algoritmos de Escalonamento (Balanceamento de Carga)

## ➤ Algoritmo Iniciado pelo Transmissor

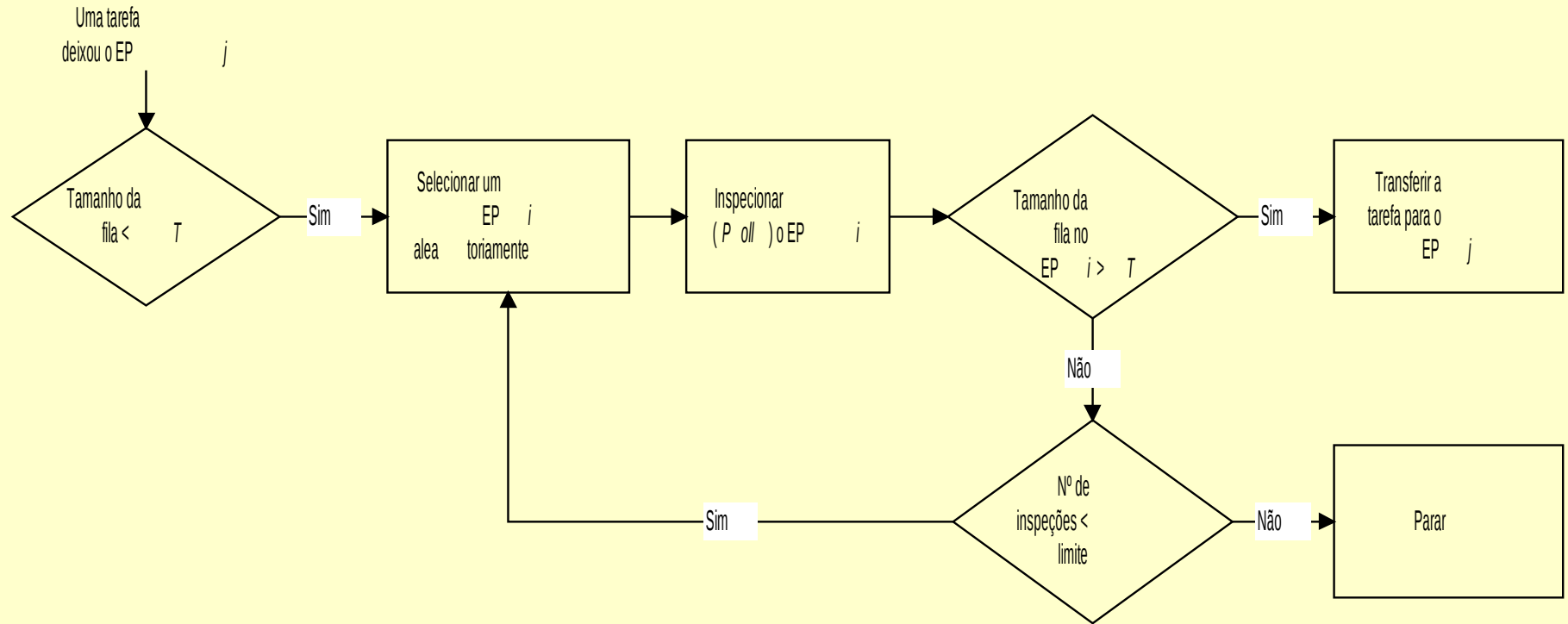


# Algoritmos de Escalonamento (Balanceamento de Carga)

- Algoritmos Iniciados pelo Receptor
  - Política de Transferência
    - Baseada no tamanho da fila da CPU
  - Política de Seleção
    - Considera todas as tarefas
  - Política de Localização
    - EP selecionado aleatoriamente e inspecionado
  - Política de Informação
    - Orientada à demanda

# Algoritmos de Escalonamento (Balanceamento de Carga)

## ➤ Algoritmo Iniciado pelo Receptor



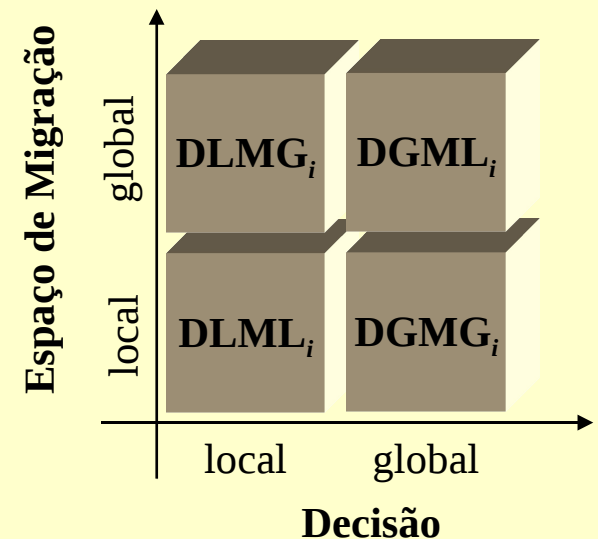
# Algoritmos de Escalonamento (Balanceamento de Carga)

- Algoritmos Iniciados Simetricamente
  - Cargas baixas
    - Transmissor procura um receptor
  - Cargas altas
    - Receptor procura por um transmissor
- Herda as vantagens e desvantagens de ambas as estratégias



# Algoritmos de Escalonamento (Balanceamento de Carga)

- Classificação de Lüling
  - Balanceamento em duas fases
    - Decisão
      - Local (vizinhos mais próximos)
      - Global (EPs de toda a rede)
    - Migração
      - Local (processadores vizinhos)
      - Global (qualquer processador)
  - Algoritmos iniciados pelo transmissor, receptor ou simétricos



# Exemplos de Sistemas

---

## ➤ *Sprite*

- Ambiente de estações de trabalho
  - Disponibilidade de recursos para seus donos
- Política de informação centralizada e dirigida pela mudança de estado
- Política de localização centralizada
- Política de seleção basicamente manual
  - O usuário escolhe as tarefas para serem executadas remotamente
- Política de transferência semi-automática
  - Tarefas externas transferidas automaticamente

# Exemplos de Sistemas

---

## ➤ *Condor*

- Ambiente de estações de trabalho
  - Disponibilidade de recursos para seus donos
- Política de seleção e transferência similares às do *Sprite*
  - Transferências iniciadas manualmente pelo usuário
- Política de informação periódica
- Política de localização centralizada

# Exemplos de Sistemas

---

## ➤ *Charlotte*

- Política de migração determinada por um processo iniciador (*starter*)
  - Coleta de informações
  - Transferência de processos
- Iniciador controla um subconjunto de EPs
  - Migração de processos
  - Atualização de informações de carga
- Fases da transferência de processos
  - Negociação
  - Transferência
  - Estabelecimento do processo migrado

# Exemplos de Sistemas

- **AMIGO:** dyn**AM**ical flex**I**ble schedul**I**ng Envir**ON**ment (AMIGO)



- Se baseia em um escalonador mestre e vários processos daemons(2 camadas)
- Aplicações paralelas/distribuídas não precisam ser modificadas ou adaptadas
- Opera conjuntamente com PVM, MPI ou CORBA

## Interface com o Usuário

### Classes de Software

Cad. de Classes  
Cad. de Aplicações

### Plataformas

Cad. de Hardware  
Detecta Hardware  
Cad. de Benchmarks  
Executa Benchmarks

### Políticas de Escalº.

Cad. de Políticas  
Rel. Classes & Políticas  
Configura Uso Políticas  
Seleciona Políticas  
Livre, Assistida, Padrão

### Monitoramento

Cad. de Métricas  
Início/Término

### Simulação

Aplicações  
Plataformas  
Políticas  
Simula

Arquivos de  
Configuração  
e de Dados

*Policy.cfg*

[config]

...

[policies]

...

[applications]

...

*"Benchmarks".cfg*

1P proc\_name result

1P proc\_name result

1P proc\_name result

...

Arquivos .dad e .cfg

classoft.dad

hardware.dad

bench.dad

....

....

Camada Superior

Camada Inferior

Mechanisms.lib

Pol\_1

Pol\_2

...

Pol\_n

AMIGO Daemon  
(AMIGOD)

Ambiente de  
Passagem de  
Mensagem

Aplicação  
Paralela

Rede de Conexão

# Conclusões

---

- Escalonamento e Balanceamento de Carga
  - Caros e difíceis de serem implementados
  - Ferramentas eficientes e capazes de tornar o balanceamento fácil e adaptável às diversas plataformas
- Migração de Processos
  - Melhorar o desempenho
  - Tolerância a falhas
  - Compartilhamento de recursos

# Conclusões

---

## ➤ Tendências

- Definir medidas efetivas de índices de carga
- Estabelecer uma organização hierárquica de sistemas
  - Distribuição de carga local
  - Políticas de balanceamento de carga local
- Incorporar ferramentas aos sistemas distribuídos
  - Diferentes políticas de balanceamento
  - Adaptáveis a