



USP - Universidade  
de São Paulo



IME - Instituto de  
Matemática e Estatística

**MAC0461 e MAC5758**

**Introdução ao Escalonamento e Aplicações**

# **Economia de Energia em Sistemas Embarcados com Multiprocessadores Homogêneos ou Heterogêneos**

Victoriano Alfonso Phocco Diaz

[alfonso7@ime.usp.br](mailto:alfonso7@ime.usp.br)

Carlos Herrera Muñoz

[camunoz@ime.usp.br](mailto:camunoz@ime.usp.br)

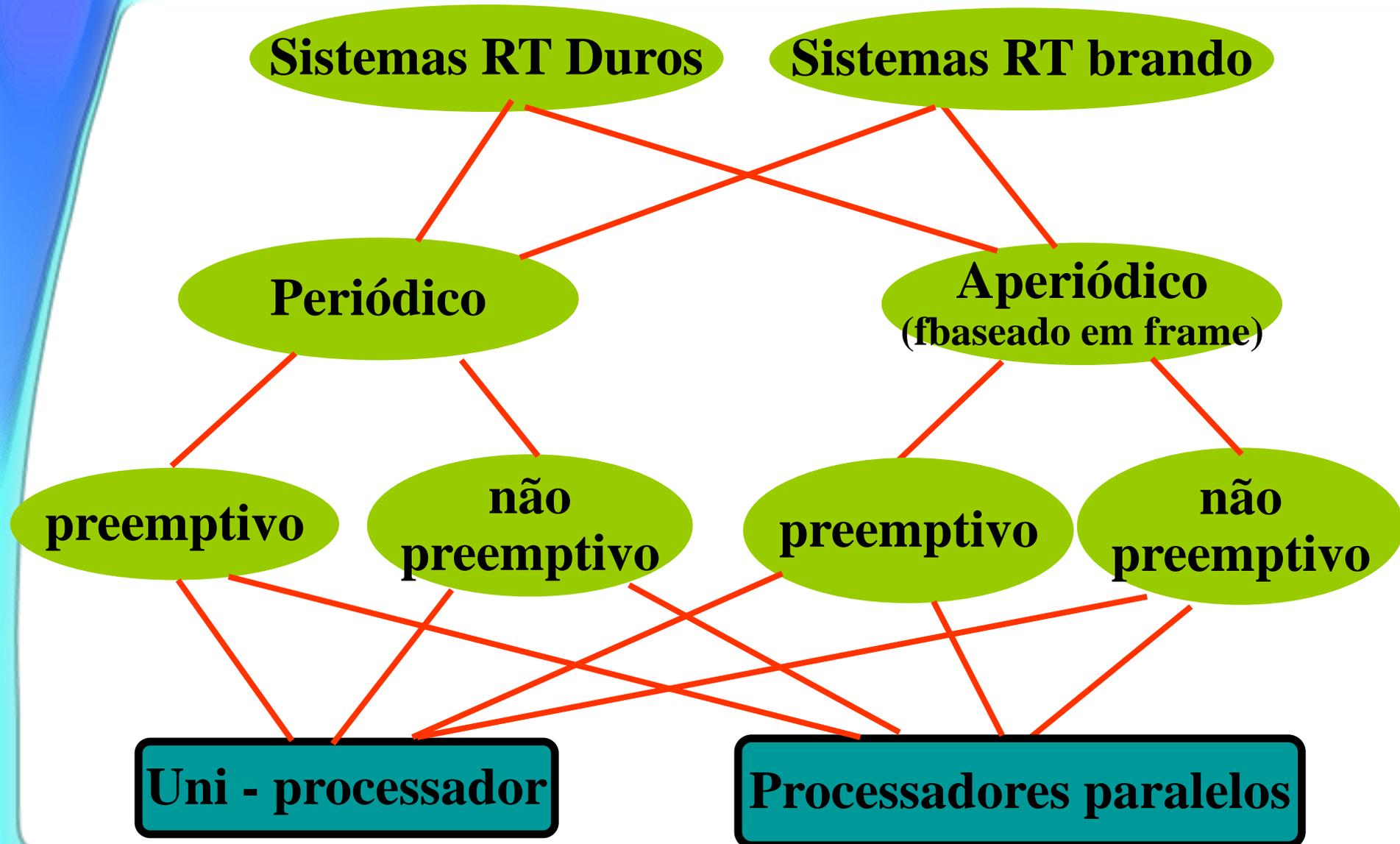


# Conteúdo

- **Conceitos**
- **Definição do problema**
- **Objetivos**
- **Preliminares**
  - modelo de sistema e aplicação
  - modelo de potência
- **Técnicas baseadas em gerenciamento da potência.**
- **Técnicas baseadas em escalonamento**
- **Descrição da técnica S&S, LAMP**
- **Descrição do CASPER**
- **Conclusões**



# Sistemas de Tempo Real (I)





# Sistemas de Tempo Real (II)

**Sistemas de TR duro**  **garantem os deadlines**

- Para garantir os deadlines, é preciso conhecer os tempos de execução no pior caso.
- Predição: É preciso conhecer se os deadlines podem ser descumprido.

**Sistemas de TR brando**  **tentam cumprir os deadlines**

- Se o *deadline* é descumprido, há uma penalidade.
- Fornece de garantias estatísticas (análise probabilística)
- É preciso conhecer a distribuição estatística dos tempos de execução.

Aplicações:

Sistemas de seguridade crítica, sistemas de controle e militares, robóticas, Comunicação, multimídia, etc.



# Sistemas Embarcados (I)

- É um sistema microprocessador no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla.
- Difere de outros sistemas computacionais, como os PCs e os supercomputadores.
- Faz tarefas específicas e predefinidas
- Devem ser confiáveis (confiabilidade, manutenção, seguridade), eficientes(em energia, peso, custo ), dedicados.
- A maioria deve cumprir com restrições de tempo real.



## Sistemas Embarcados (II)

- São sinônimos Embarcados e Tempo Real?
- A maioria dos sistemas embarcados são de tempo real
- A maioria dos sistemas de tempo real são embarcados





# Motivação

- Sistemas embarcados
  - Requerem alto desempenho.
  - Precisam muitos processadores.
- A Potência/Energia é limitada
  - **Potência dinâmica** : relação quadrática da voltagem.
  - **Potência estática**: relação exponencial da voltagem.
  - Então o gerenciamento da voltagem é o indicado.

$$f \propto V \quad P \propto f^3$$

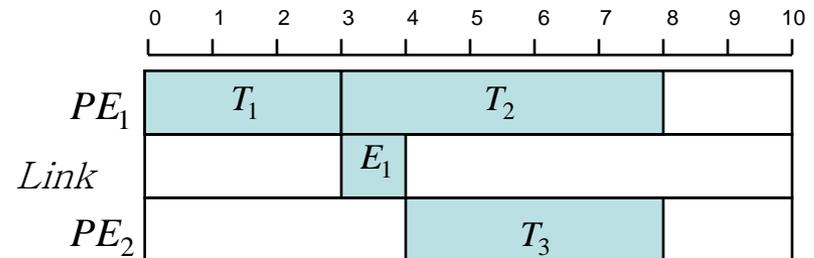
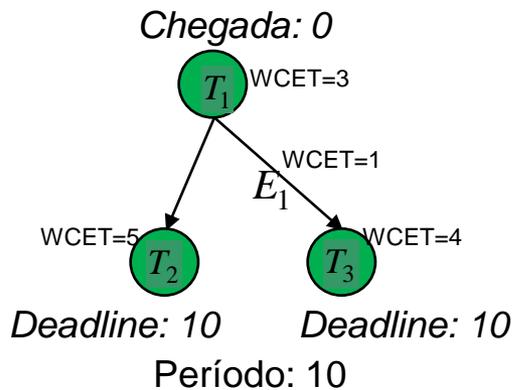
$$E_d = P \cdot t = P \cdot (\text{workload}/f) \Rightarrow E_d \propto f^2 (V^2)$$

Se  $f' = 1/2f$ ,  $E_d' = 1/4E_d$ . Energia poupada é  $3/4E_d$ .



# Definição do Problema

- Temos um sistema de tempo real, de vários processadores (ou um processador “*multicore*”), tarefas que precisam ser feitas e pouca potência.
- Dado :
  - Grafo de tarefas



- Problema:
  - **Como minimizar o consumo de Energia?**

WCET: worst case execution time



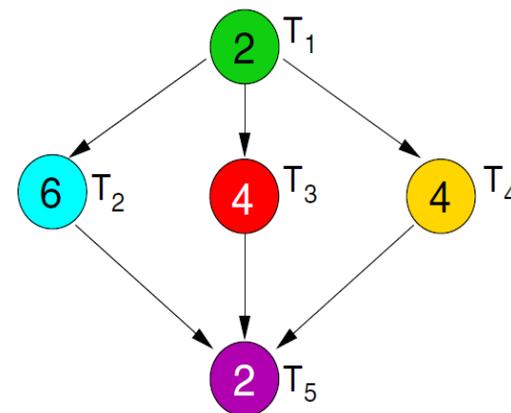
# Preliminares (I)

## ■ Modelo de Sistema e Aplicação.

Sistema que roda aplicações paralelamente onde o escalonamento é determinado estaticamente.

## ■ As aplicações são representadas por DAG (*Directed Acyclic Graphs*)

- Setas – dependências
- Pesos dos nós - tempo de processamento.





# Preliminares (II)

## Modelo de Potência

$$P = P_{AC} + P_{DC} + P_{on}$$

$P_{AC}$  = Consumo de potência dinâmica.

$P_{DC}$  = Consumo de potência estática.

$P_{on}$  = Potência mínima para manter ligado o processador.

$$P_{AC} = aC_{eff}V_{dd}^2f$$

$a$  = fator de atividade

$C_{eff}$  = capacidade de comutação efetiva

$V_{dd}$  = voltagem fornecido

$f$  = frequência de operação



# Preliminares (III)

## Modelo de Potência (cont.)

$$P_{DC} = V_{dd} I_{subn} + |V_{bs}| I_j,$$

$I_{subn}$  = sub-limiar da corrente de fuga.

$V_{bs}$  = voltagem aplicada entre o copo e a fonte.

$I_j$  = união de polarização inversa atual.

$$I_{subn} = K_3 e^{K_4 V_{dd}} e^{K_5 V_{bs}},$$

$K_3$ ,  $K_4$ ,  $K_5$  são constantes



# Preliminares (IV)

## Modelo de Potência(cont.)

$$f = (V_{dd} - V_{th})^\alpha / L_d K_6,$$

$L_d$  = profundidade lógica

$$V_{th} = V_{th1} - K_1 V_{dd} - K_2 V_{bs},$$

$V_{th}$  = voltagem limiar

Constant	Value
$K_1$	0.063
$K_2$	0.153
$K_3$	$5.38 \cdot 10^{-7}$
$K_4$	1.83
$K_5$	4.19
$K_6$	$5.26 \cdot 10^{-12}$
$K_7$	-0.144
$V_{dd0}$	1.0
$V_{bs}$	-0.7
$\alpha$	1.5
$V_{th1}$	0.244
$I_j$	$4.8 \cdot 10^{-10}$
$C_{eff}$	$0.43 \cdot 10^{-9}$
$L_d$	37.0
$L_g$	$4.0 \cdot 10^6$



# Preliminares (V)

## Modelo de Potência (cont.)

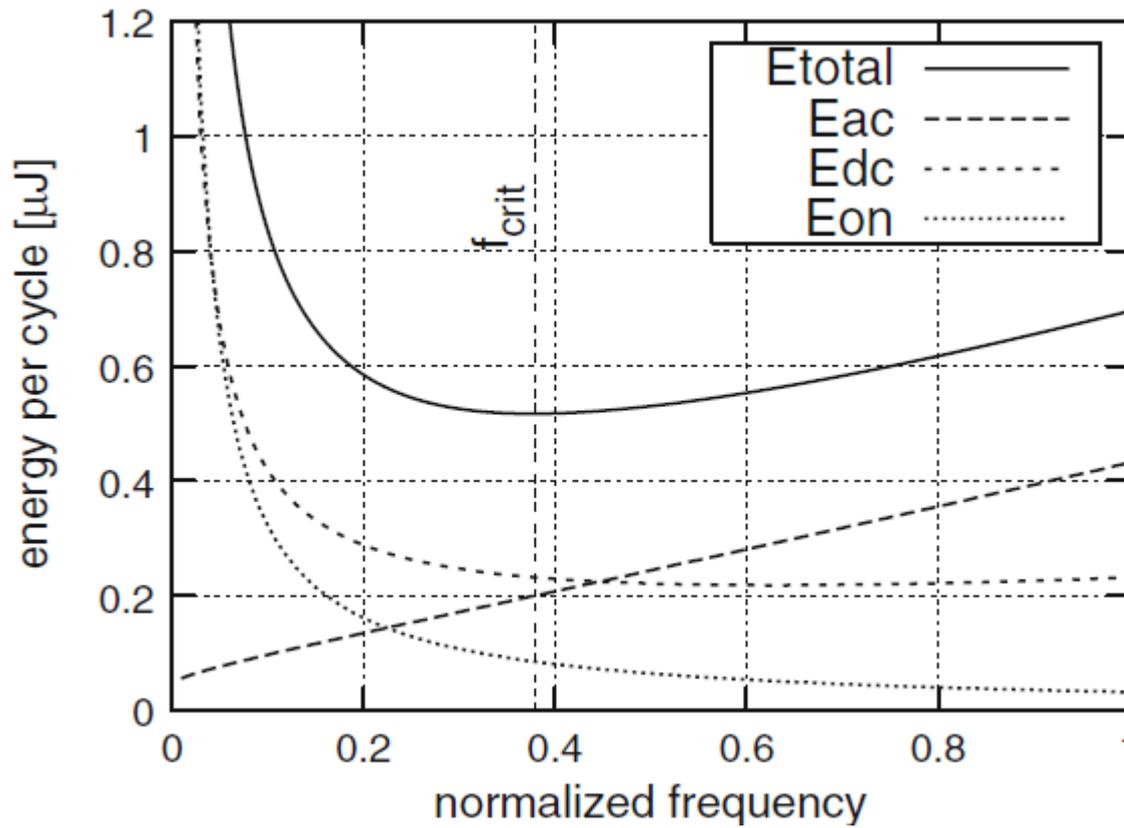


Figura: Energia Vs. frequência



# Técnicas baseadas em escalonamento de tarefas

## S&S

- Grafo é escalonado com EDF (“*Earliest deadline first*”)
- Tempo de *slack* é utilizado para decrementar o nível de voltagem frequência de todos os processadores.



# Técnicas baseadas em escalonamento

## S&S

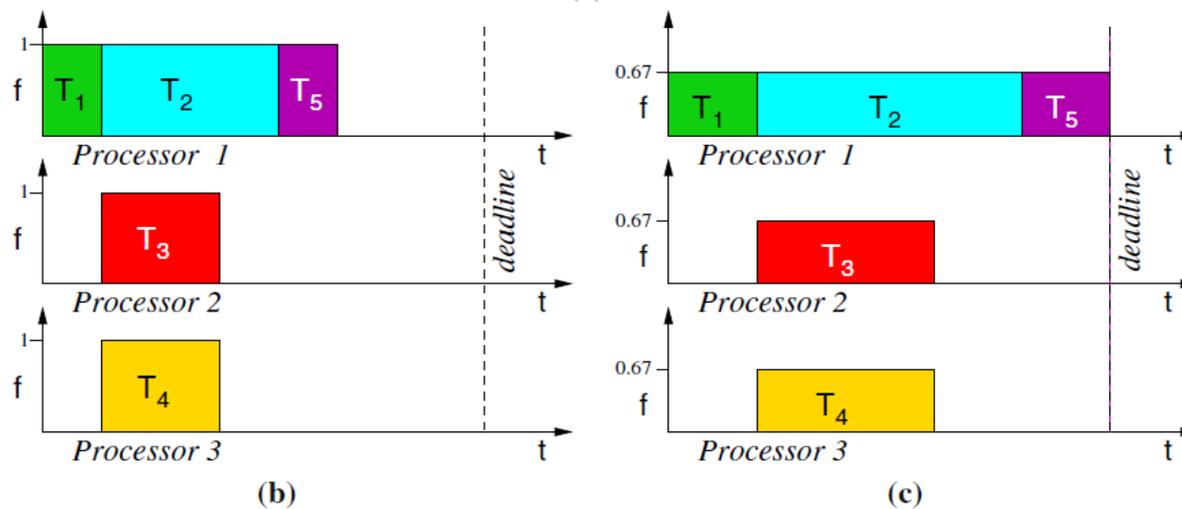
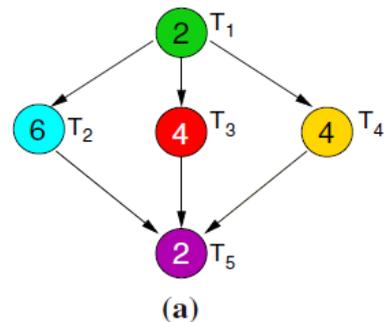


Figura: técnica S&S



# Técnicas baseadas em escalonamento

- **LAMPS**

- Primeiro determinamos o número mínimo de processadores requisitados para terminar as tarefas antes do deadline.
  - Lower bound = número mínimo de processadores para terminar as tarefas antes do deadline.
  - Upper bound = número de processadores que podem ser utilizado eficientemente.

$$N_{\text{lwb}} = \lceil \sum_{v \in V} w(v)/D \rceil, \quad N_{\text{upb}} = |V|.$$



# Técnicas baseadas em escalonamento

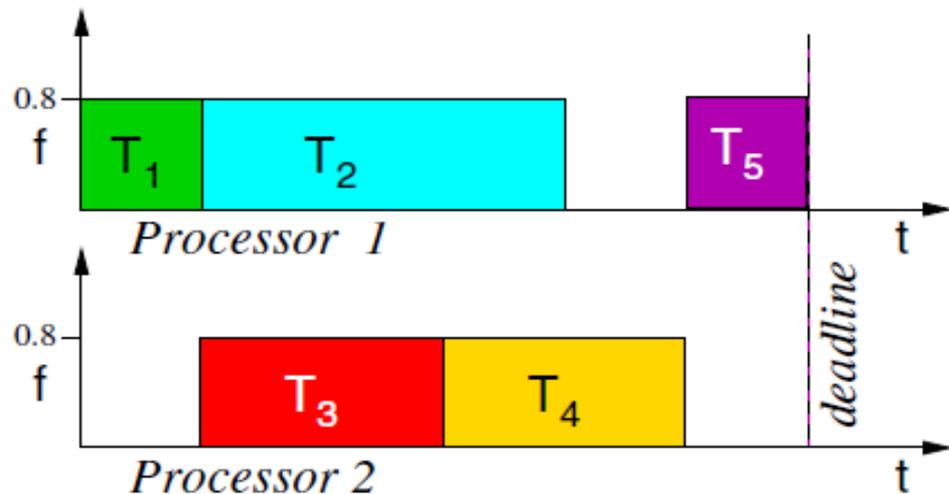
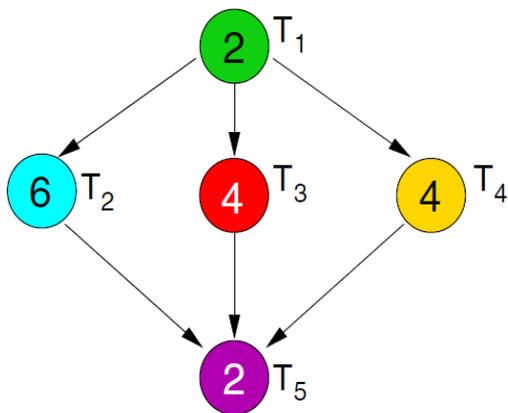
- **LAMPS (cont.)**

- Número de processadores que dissipam a menor quantidade de potência.
  - Calculamos a potência consumida para  **$N_{min}$**  processadores isto é feito decrementando a voltagem/freqüência deste jeito o grafo de tarefas é terminado tão perto possível do *deadline*
  - Isto é feito também para  **$N_{min}+1$   $N_{min}+2...$**  até o aumento de processadores não reduza o ***makespan*** do escalonamento.



# Técnicas baseadas em escalonamento

- LAMPS(cont.)



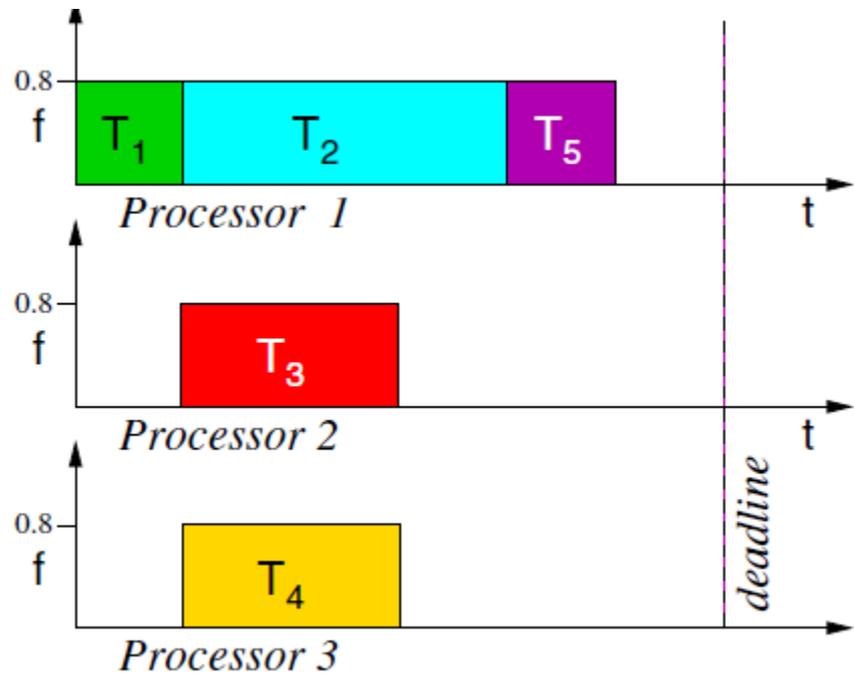
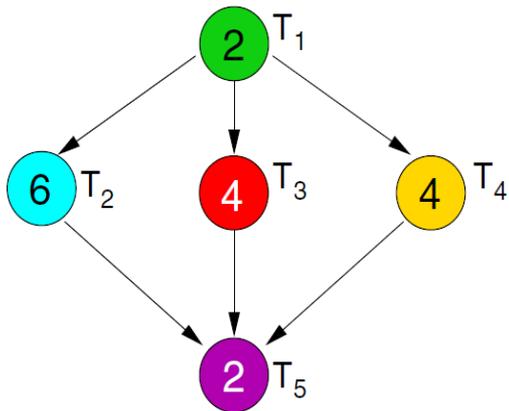


- **S&S+PS**

- No S&S+PS o tempo restante depois de completar a última tarefa é usado para desligar um processador se o período é suficientemente grande.



- S&S+PS (cont.)





## ■ LAMP+PS

- Determinamos o número de processadores para encontrar o deadline e o numero de processadores que executam o grafo de tarefas eficientemente.
- Para o intervalo determinamos o balanço entre DVS e PS variando a freqüência da máxima ate a mínima. requerida para encontrar ao deadline.
- Para cada freqüência usamos o tempo restante para desligar o processador



## ■ **LIMIT-SF**

- EDF nem sempre dá o escalonamento ótimo
- Processadores desocupados não consomem energia.
- O número de processadores é igual ao numero de tarefas (utilizar poucos processadores não reduz energia)
- A freqüência é baixada ate a freqüência ótima para encontrar o deadline ou tão perto for possível

## ■ **LIMIT-MF**

- Todas as tarefas são escalonadas na freqüência critica.
- É um *lower bound* incluso se os processadores pudessem rodas a múltiplas freqüências



# Experiências LAMPS S&S

- Consumo de energia para tarefas de grão grosso com 1.5x do caminho crítico

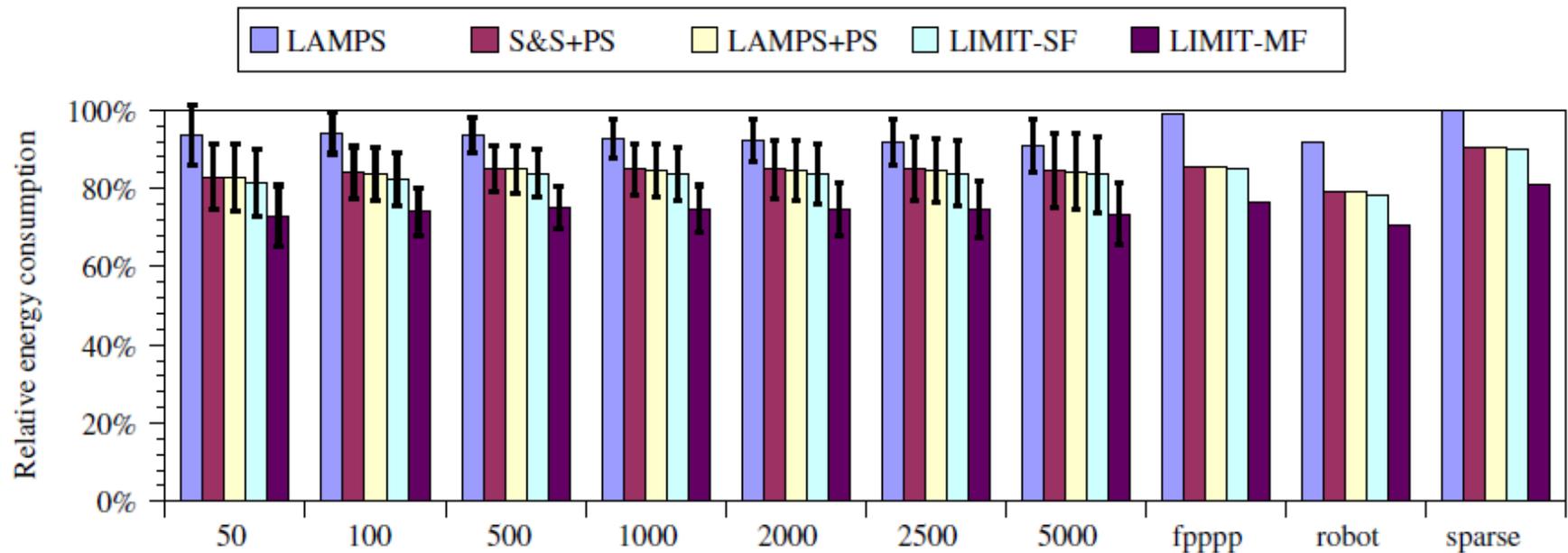


Figura: Gráfico do Consumo de Energia de grão grosso



# Experiências LAMPS S&S

- Consumo de energia para tarefas de grão fino com 8x do caminho crítico

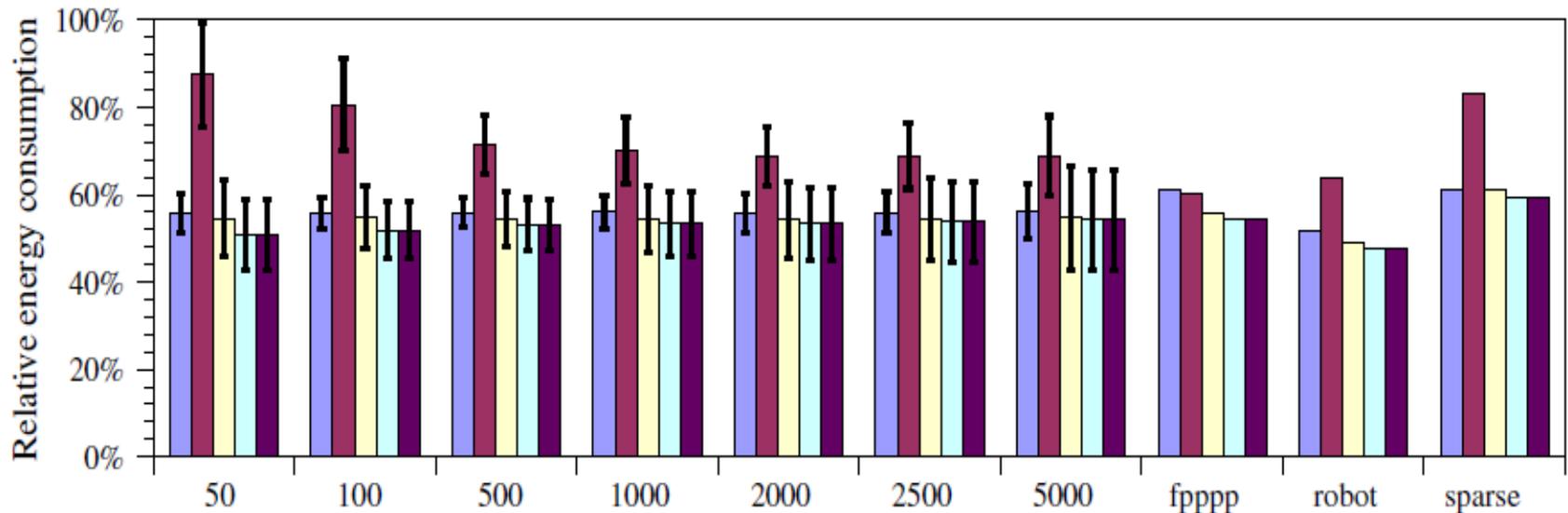


Figura: Gráfico do Consumo de Energia de grão grosso



# CASPER

- **CASPER** ( Combined Assignment, Scheduling, and Power Management )
- É um framework que integra alocação, ordenação e escalonamento de tarefas e gerenciamento de potência ou energia estática.
- Modelo de Potência usado:

$$p_d = C_{ef} \cdot V_{dd}^2 \cdot f, \quad (1)$$

$$f = k \cdot (V_{dd} - V_t)^2 / V_{dd}, \quad (2)$$

- O coração do CASPER é um algoritmo genético de lista de escalonamentos que codifica a alocação e ordenamento de tarefas em um individuo de um só cromossoma.



# CASPER (II)

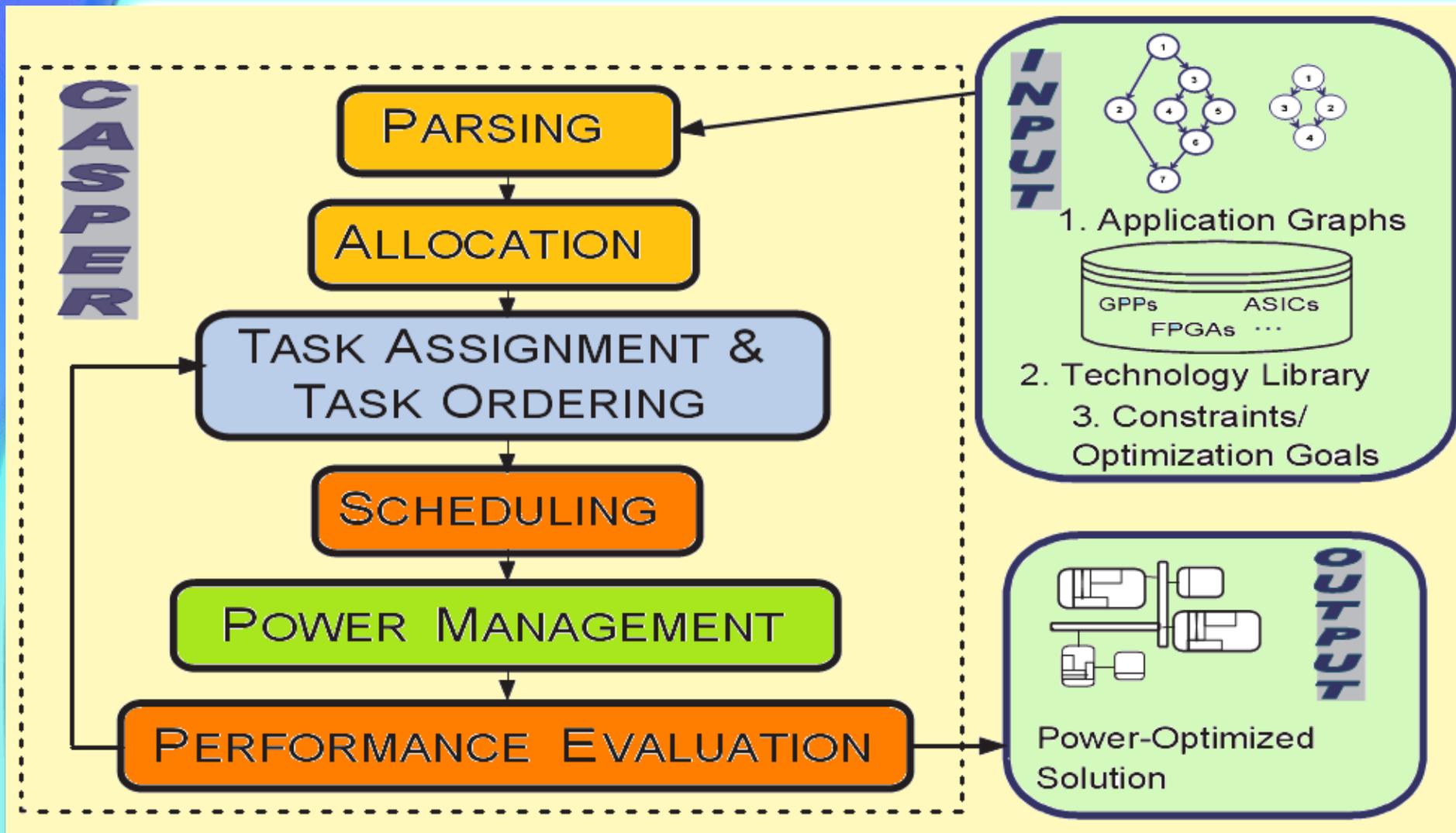
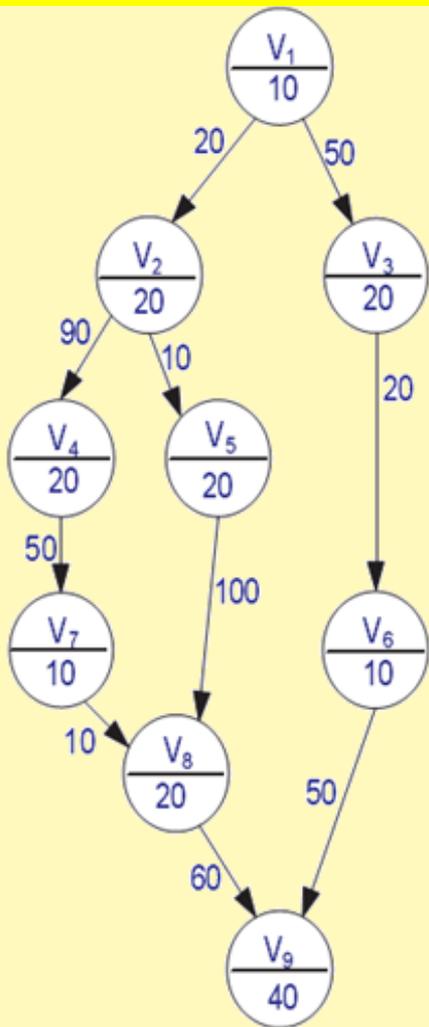


Figura: Fluxo do CASPER framework



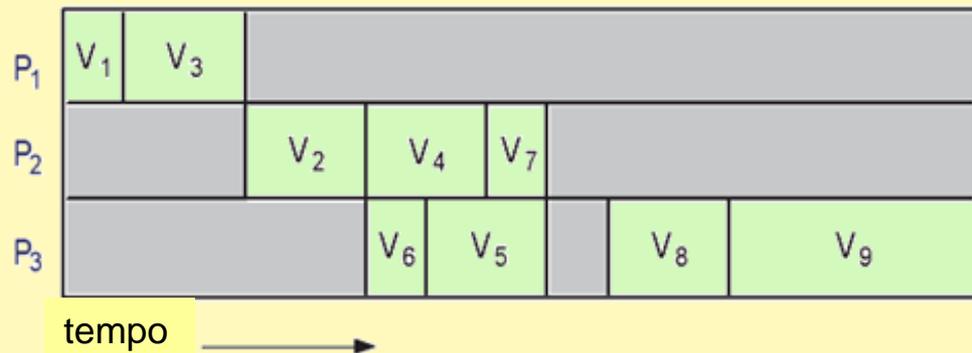
# CASPER (III)

## Grafo de tarefas



Tasks	$h_{init}$	height
V <sub>1</sub>	0	0
V <sub>2</sub>	1	1
V <sub>3</sub>	1	1
V <sub>4</sub>	2	2
V <sub>5</sub>	2	2,3
V <sub>6</sub>	2	2,3,4
V <sub>7</sub>	3	3
V <sub>8</sub>	4	4
V <sub>9</sub>	5	5

## Escalonamento



## Representação em String

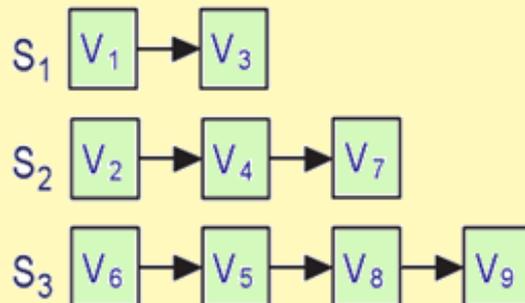


Figura: Representação em String de um escalonamento, que forma o cromossoma do individuo



# CASPER (IV)

## Alocação e Escalonamento integrados (usando o algoritmo genético):

- **Cromossoma:** lista de  $n_p$  strings, cada mantém a alocação e ordem de execução das tarefas em um PE( elemento de processamento).

- **Fitness:** 
$$F_i(I_i, P_t) = \begin{cases} \frac{\tau_c(I_i, P_t)}{2} & \Delta_c(I_i, P_t) > 0, \\ \frac{1 + E_{opt}(I_i, P_t)}{2} & \Delta_c(I_i, P_t) \leq 0, \end{cases}$$

- **Seleção:** Roleta
- **Cruzamento:** partindo e alocando cada parte dos pais nos filhos.
- **Mutação:** dado uma tarefa  $v_i$  logo escolher aleatoriamente outra  $v_j$  com o mesmo tamanho para trocar a posição das duas tarefas.



# CASPER (V)

## Técnicas para o gerenciamento de Potência

- Sistemas de Multiprocessadores **Homogêneos**.
  - PDP-SPM (Static Power Management with Proportional Distribution and Parallelism Algorithm)
  
- Sistemas de Multiprocessadores **Heterogêneos**.
  - PV-DVS( Power Variation DVS algortihm)



## Conclusões

- Dependendo da quantidade de tempo restante (slack ) que continua antes do deadline, a quantidade de paralelismo, granularidade da aplicação, variação da voltagem e desligamento de processadores pode ser usado para reduzir significativamente a energia consumida.
- LAMPS-PS diminui o consumo de energia para uma implementação do MPEG-1 por não menos de 40% comparado com o S&S.
- CASPER mostra que integrando o mapeamento, ordenamento e escalonamento de tarefas numa só iteração sem ter outra interna melhora a poupança de energia num 8% de média comparado com outro que não tem esta abordagem integrado.
- O framework CASPER é usado para escalonar tarefas e economizar energia para sistemas multiprocessadores homogêneos e heterogêneos usando na otimização de energia o PDP-SPM para o caso homogêneo e o PV-DVS para o caso heterogêneo, no entanto é flexível e pode trabalhar com outras técnicas.



## Bibliografia

- *“Leakage-Aware Multiprocessor Scheduling”*, Pepijn de Langen · Ben Juurlink
- *“CASPER: An Integrated Energy-Driven Approach for Task Graph Scheduling on Distributed Embedded Systems”*, Vida Kianzad, Shuvra S. Bhattacharyya and Gang Qu, *ECE Department and Institute for Advanced Computer Studies*
- *“Um breve survey: Escalonamento em sistemas de tempo real com otimizacao em consumo de energia”*, L. Bertini, Universidade Federal de Fluminense.



# Perguntas ?