

---

## Sistemas de Tempo Real: Multiprocessadores 2

Rômulo Silva de Oliveira  
Departamento de Automação e Sistemas – DAS – UFSC

romulo@das.ufsc.br  
<http://www.das.ufsc.br/~romulo>

---

## Referências

- A Comparison of Global and Partitioned EDF Schedulability Tests for Multiprocessors
  - Theodore P. Baker
  - Technical Report TR-051101, Department of Computer Science, Florida State University, 2005

---

## Introdução

- Existe interesse em usar multi-core e multiprocessadores para sistemas embutidos de tempo real com alto desempenho
- A abordagem predominante para escalonar sistemas de tempo real críticos multiprocessados tem sido o particionamento
- Tarefas são alocadas estaticamente a um processador
- Escalonamento particionado permite que a escalonabilidade seja verificada através das bem conhecidas técnicas de análise para processadores únicos

---

## Introdução

- A alternativa ao escalonamento particionado é o escalonamento global
- Existe uma única fila de jobs
- Da qual jobs são despachados para qualquer processador disponível de acordo com um esquema de prioridades global
- Por exemplo
- A política de escalonamento EDF-US[1/2]
  - Na qual uma poucas tarefas de alta utilização são escalonadas com prioridade máxima
  - E as demais tarefas são escalonadas de acordo com os deadlines
- Garante a escalonabilidade no pior caso até o mesmo nível de utilização que o EDF particionado consegue

---

## Introdução

- Escalonamento global permanece controverso
- Existem aqueles que acreditam que
  - O overhead de sincronizar os escalonadores entre processadores
  - E a perda de desempenho devido a faltas (misses) no translation look-aside buffer e nas memórias caches após uma migração de tarefas entre processadores
  - Vai anular qualquer possível melhoria na eficiência do escalonamento
- Por outro lado
  - O conceito de escalonamento global é atraente
  - Especialmente em sistemas onde o tempo de resposta no caso médio também é importante, e não só no pior caso
- É bem sabido da teoria das filas que escalonamento com fila única produz melhores tempos médio de resposta do que escalonamento com uma fila por processador

---

## Introdução

- Este artigo tenta comparar o corrente estado da arte do escalonamento EDF global com o estado da arte do escalonamento EDF particionado
- O desempenho no pior caso de ambas abordagens foi mostrado equivalente
  - Ao menos quando o deadline é igual ao período
- Será feita uma comparação empírica do desempenho
- Quais são as chances de um conjunto de tarefas periódicas ou esporádicas escolhido aleatoriamente ser garantido por uma dada combinação de política de escalonamento e teste de escalonabilidade ?

### Escalonamento Particionado 1/3

- Andersson, Baruah, and Jonsson[1] mostraram que a utilização garantida para EDF ou para qualquer outro algoritmo de escalonamento usando prioridade fixa para multiprocessadores particionado ou global não pode ser maior que  $(m + 1)/2$  para uma plataforma com  $m$  processadores
- Fazer tal particionamento otimamente é reduzível ao problema do "bin packing" e outros problemas de particionamento inteiro, os quais são NP-completo
- Logo, a pesquisa tem focado na análise de heurísticas para a alocação de tarefas a processadores
  - E em como eles vão quando comparados ao algoritmo ótimo

### Escalonamento Particionado 2/3

- Lopez, Diaz, Garcia e Garcia[18] mostraram que é possível escalonar em  $m$  processadores qualquer conjunto de  $n$  tarefas periódicas independentes com a utilização individual máxima de  $U_{\max}$  e utilização total

$$U < \frac{m\beta_{EDF} + 1}{\beta_{EDF} + 1} \text{ where } \beta_{EDF} = \lceil 1/u_{\max} \rceil.$$

- Para o caso restrito onde  $U_{\max} = 1$  e  $\beta_{EDF} = 1$ , o limite de utilização garantida fica  $(m + 1)/2$

### Escalonamento Particionado 3/3

- Baruah e Fisher[5] estudaram um algoritmo de particionamento que aloca tarefas a processadores na ordem First-Fit Deadline-Monotonic
- O teste de "fit" para processador único é baseado na análise da função limite de demanda (demand-bound function)

$$d_i - \sum_{j=1}^N DBF^*(j, d_i) \geq c_i$$

$$1 - \sum_{j=1}^N u_j \geq u_i$$

where  $u_i = c_i/T_i$  and

$$DBF^*(i, t) = \begin{cases} 0, & \text{if } t < d_i \\ c_i + (t - d_i)u_i, & \text{otherwise} \end{cases}$$

### Escalonamento Global 1/10

- Goossens, Funk, e Baruah [14] mostraram que um sistema de tarefas periódicas independentes pode ser escalonado
- Em  $m$  processadores por EDF se
- A utilização total for no máximo  $m(1 - u_{\max}) + u_{\max}$ 
  - Onde  $u_{\max}$  é a máxima utilização de qualquer tarefa individual

### Escalonamento Global 2/10

- Bertogna, Cirinei e Lipari[10] melhoram os testes de escalonabilidade para EDF global
- Primeiro, observaram que a prova para o teste por limiar de utilização em [14] estende-se naturalmente para deadlines menores que o período se a utilização  $u_i$  for substituída por  $c_i/D_i$
- A mesma prova estende-se para o caso de deadlines maiores que o período se  $c_i/D_i$  for substituído por  $c_i / \min\{D_i, T_i\}$

### Escalonamento Global 3/10

**Theorem 2 (GFB)** A set of sporadic tasks  $\tau_1, \dots, \tau_N$  is EDF schedulable on  $m$  identical processors if

$$\sum_{i=1}^N \lambda_i \leq m - \lambda_{\max}(m - 1)$$

where  $\lambda_{\max} = \max\{\lambda_i | i = 1, \dots, N\}$ .

### Escalonamento Global 4/10

- Srinivasan e Baruah[20] examinaram o escalonamento global por EDF de tarefas periódicas em multiprocessadores
- Mostraram que qualquer sistema de tarefas periódicas independentes no qual a utilização máxima de uma tarefa individual é  $m/(2m - 1)$  pode ser escalonado em  $m$  processadores se a utilização total for no máximo  $m^2/(2m - 1)$

### Escalonamento Global 5/10

- Srinivasan e Baruah[20] propuseram um método para lidar com algumas poucas tarefas pesadas, usando uma política de escalonamento híbrida
- Dá prioridade fixa mais alta para as tarefas com utilização maior que uma certa constante  $X$ 
  - Escalona as outras tarefas de acordo com EDF algorithm
  - Algoritmo é chamado de EDF-US[X]
- Algoritmo EDF-US[ $m/(2m - 2)$ ]
  - Escalona corretamente em  $m$  processadores qualquer conjunto de tarefas periódicas com utilização total  $U \leq m^2/(2m - 2)$
- Algoritmo RM-US[ $m/(2m - 2)$ ]
  - Escalona corretamente em  $m$  processadores qualquer conjunto de tarefas periódicas com utilização total  $U \leq m^2/(3m - 2)$

### Escalonamento Global 6/10

- Baker[2, 3] criou vários testes suficientes de escalonabilidade para escalonamento preemptivo EDF em  $m$  processadores
  - Conjuntos de tarefas periódicas e esporádicas com deadlines arbitrários
- Mostrou que o valor ótimo de  $X$  em EDF-US[X] com respeito à maximização da utilização garantida no pior caso é  $X = 1/2$ 
  - Neste caso o limiar de utilização é  $(m + 1)/2$

### Escalonamento Global 7/10

- Bertogna, Cirinei e Lipari[10] também desenvolveram o teste abaixo

**Theorem 3 (BCL)** A set of sporadic tasks  $\tau_1, \dots, \tau_N$  (with constraint  $d_i \leq T_i$ ) is EDF schedulable on  $m$  identical processors if for each task  $\tau_k$  one of the following is true:

$$\sum_{i \neq k} \min\{\beta_i, 1 - \lambda_k\} < m(1 - \lambda_k) \quad (3)$$

$$\sum_{i \neq k} \min\{\beta_i, 1 - \lambda_k\} = m(1 - \lambda_k) \text{ and } \exists i \neq k : 0 < \beta_i \leq 1 - \lambda_k \quad (4)$$

where

$$\beta_i = \frac{N_i c_i + \min\{c_i, \max\{0, d_k - N_i T_i\}\}}{d_k}$$

and

$$N_i = \left\lceil \frac{d_k - d_i}{T_i} \right\rceil + 1$$

### Escalonamento Global 8/10

- Uma observação chave de BCL é que se uma tarefa  $k$  perde um deadline isto é devido à *interferência* de outras tarefas, e a máxima fração da carga de qualquer tarefa que pode contribuir para esta interferência é  $(1 - \lambda_k)$
- Baker[4] incorporou esta observação na análise apresentada em [3], para obter o seguinte teste:

### Escalonamento Global 9/10

**Theorem 4 (BAK2)** A set of sporadic tasks  $\tau_1, \dots, \tau_N$  is EDF schedulable on  $m$  identical processors if for each task  $\tau_k$  there exists  $\lambda > \frac{c_k}{d_k}$  such that one or more of the following is true:

$$\sum_{i=1}^N \min\{\beta_k^i(t), 1 - \lambda_k\} < m(1 - \lambda_k) \quad (5)$$

$$\sum_{i=1}^N \min\{1 - \lambda_k, \beta_k^i(t)\} = m(1 - \lambda_k) \text{ and } \exists i : 0 < \beta_k^i(t) < 1 - \lambda_k \quad (6)$$

$$\sum_{i=1}^N \min\{1, \beta_k^i(t)\} \leq m(1 - \lambda_k) + \lambda_k \quad (7)$$

where  $\lambda_k = \lambda \max\{1, \frac{d_k}{c_k}\}$  and

$$\beta_k^i(t) = \begin{cases} \max\{u_i, u_i(1 - \frac{t}{d_k}) + \frac{c_k}{d_k}\} & \text{if } \frac{t}{d_k} \leq \lambda \\ \frac{c_k}{d_k} & \text{if } \frac{t}{d_k} > \lambda \text{ and } \lambda \geq \frac{c_k}{d_k} \\ \frac{c_k}{d_k} + \frac{t - \lambda d_k}{d_k} & \text{if } \frac{t}{d_k} > \lambda \text{ and } \frac{c_k}{d_k} > \lambda \end{cases}$$

## Escalonamento Global 10/10

The above theorem can be used as a schedulability test by attempting to verify the three conditions for each value of  $k$ . The test is of complexity  $O(N^3)$  since the only values of  $\lambda$  that need be considered are the minimum and the points where  $\beta_k^*(i)$  is discontinuous, i.e.,

- $\lambda = u_i, i = 1, \dots, N$
- $\lambda = c_i/d_i, \text{ if } d_i > T_i$

## Comparações

- Para avaliar a eficácia do teste BAK2 em comparação os testes já existentes para EDF global
- Para comparar a eficácia dos escalonamentos globais versus escalonamento particionados
- Foi feita uma série de experimentos usando conjuntos de tarefas gerados pseudo-aleatoriamente

## Condições dos testes 1/2

- Diversos datasets
- Cada dataset composto por 1,000,000 conjuntos de tarefas
- Períodos gerados pseudo-aleatoriamente com distribuição uniforme entre 1 e 1000
- Utilização dos processadores (tempos de computação) escolhidos de acordo com as seguintes distribuições, truncadas para limitar a utilização entre 0.001 e 0.999
  - Distribuição uniforme entre 1/período e 1
  - Distribuição bimodal:
    - tarefas pesadas uniforme entre 0.5 and 1
    - Tarefas leves uniforme entre 1/período e 0.5
    - Probabilidade de ser pesada: 1/32
  - Distribuição exponencial com média 0.25
  - Distribuição exponencial com média 0.50
- Deadlines escolhidos de duas formas diferentes:
  - Constrained: distribuição uniforme entre tempo de computação e período
  - Unconstrained: distribuição uniforme entre tempo de execução e 4 vezes o período

## Condições dos testes 2/2

- Datasets foram gerados para sistemas com 2, 4, e 8 processadores
- Um conjunto inicial de  $m+1$  tarefas é gerado e testado
- Então outra tarefa é gerada e adicionada ao conjunto existente
- E todos os testes de escalonabilidade executados no novo conjunto
- Adição de tarefas é repetida até a utilização total exceder  $m$
  
- Todo o procedimento é repetido para um novo conjunto inicial de  $m+1$  tarefas
- Todos as experiências realizadas com todos os 24 datasets
- Apenas os resultados de algumas experiências são mostrados
  - Tendências nos vários experimentos são similares

## Experiências com EDF Global

- Para escolher um teste de escalonabilidade do tipo EDF global
- Experiências comparando desempenho dos vários testes de escalonabilidade
  - Escalonamento EDF puro e vários híbridos
- Experimentos para comparar os seguintes testes suficientes de escalonabilidade:
  - **BAK** Teste de Baker[2, 3]
  - **GFB** Teste de Goosens, Funk e Baruah, estendido para deadlines arbitrários por Bertogna, Cirinei e Lipari (teorema 2)
  - **BCL** Teste de Bertogna, Cirinei e Lipari (teorema 3)
  - **BAK2** Teste de Baker revisado (teorema 4)

## Experiências com EDF Global

- Três figuras com os resultados para três datasets
- Mostrado histograma com 100 buckets
  - Cada bucket corresponde a 1% da utilização total possível para a máquina em questão
- O único teste necessário e suficiente (exato) para escalonamento EDF Global de  $N$  tarefas em  $m$  processadores tem tempo de execução no pior caso na ordem de  $O(mN \cdot \prod_{i=1}^N(T_i, C_i))$ 
  - Impraticável para os datasets em questão
- Simulação do escalonamento até o MMC dos períodos não é um teste exato de escalonabilidade para
  - Conjuntos de tarefas esporádicas
  - Conjuntos de tarefas periódicas com offset inicial arbitrário
  - Conjuntos de tarefas periódicas e síncronas, mas com  $D > P$

### EDF Global, $D \leq P$ , bimodal, 4 CPUs

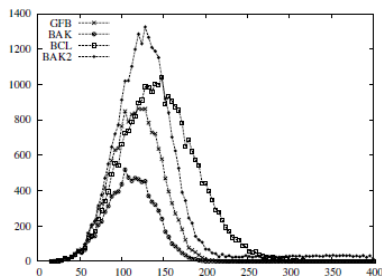


Figure 1. Constrained deadline, bimodal utilization distribution, 4 CPUs

Rômulo Silva de Oliveira, DAS-UFSC, novembro/2009 25

### EDF Global, $D \leq P$ , exponencial, 2 CPUs

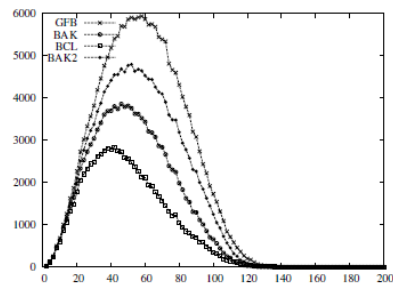


Figure 2. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

Rômulo Silva de Oliveira, DAS-UFSC, novembro/2009 26

### EDF Global, D qualquer, exponencial, 4 CPUs

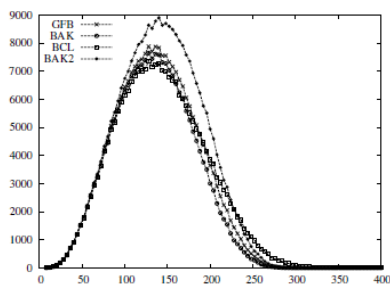


Figure 3. Unconstrained deadline, exponential utilization w/mean 0.25, 4 CPUs

Rômulo Silva de Oliveira, DAS-UFSC, novembro/2009 27

### Experiências com EDF Global

- Como cada um dos 3 testes de escalonabilidade mostram alguns casos onde eles são mais precisos em determinar a escalonabilidade
- Faz sentido aplicar eles combinados
- Começando pelo que requer menos esforço computacional
- 1. aplica teste GFB
- 2. Se teste GFB falhar e deadlines menores ou iguais ao período, aplica teste BCL
- 3. Se os outros testes falharem, aplica teste BAK2

Rômulo Silva de Oliveira, DAS-UFSC, novembro/2009 28

### Experiências com EDF Global

- Figuras 4-5 mostram resultados deste teste de 3 estágios nos mesmos datasets das figuras 1-3
  - Resultados típicos para os 24 datasets
- Está claro que a combinação de todos os testes disponíveis é o vencedor entre os testes de escalonabilidade para escalonamento EDF puro
- O teste combinado será chamado de teste “GBB”

Rômulo Silva de Oliveira, DAS-UFSC, novembro/2009 29

### EDF Global, $D \leq P$ , bimodal, 4 CPUs

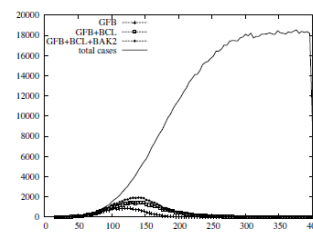


Figure 4. Constrained deadline, bimodal utilizations, 4 CPUs

Rômulo Silva de Oliveira, DAS-UFSC, novembro/2009 30

### EDF Global, $D \leq P$ , exponencial, 2 CPUs

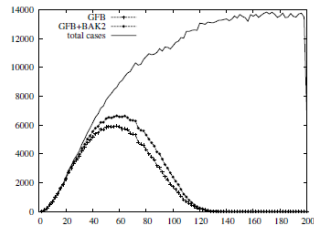


Figure 5. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

### Híbrido Global

- Foi analisado o desempenho de soluções híbridas de escalonamento EDF e highest-utilization-first
- 1. EDF-US[1/2]:
  - Prioridade especial para as tarefas de utilização maior que 1/2, valor que garante a utilização mais alta no pior caso quando  $D=P$
- 2. EDF-UM:
  - Prioridade especial para as  $k$  tarefas com utilização mais alta, onde  $k$  é o menor valor entre 0 e  $m$  para o qual o sistema pode ser verificado como escalonável de acordo com o teste GBB
- 3. EDF-LM:
  - Prioridade especial para as  $k$  tarefas com o valor maior de  $c_i / \max\{T_i, d_i\}$
  - $K$  é o menor valor entre 0 e  $m$  para o qual o sistema pode ser verificado como escalonável de acordo com o teste GBB

### Híbrido Global

- Figuras 6-7 mostram os resultados de aplicar estas 3 políticas híbridas de escalonamento
  - com o teste GBB test
  - Nos mesmos datasets das figuras 1-3
- O teste GBB foi aplicado às  $N - k$  tarefas nos  $m - k$  processadores, após as  $k$  tarefas especiais escolhidas receberem a prioridade máxima

### Híbrido Global, $D \leq P$ , bimodal, 4 CPUs

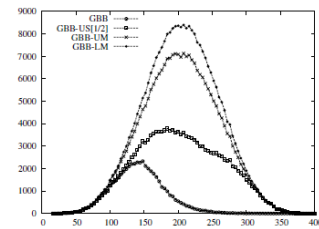


Figure 6. Constrained deadline, bimodal utilization, 4 CPUs

### Híbrido Global, $D \leq P$ , exponencial, 2 CPUs

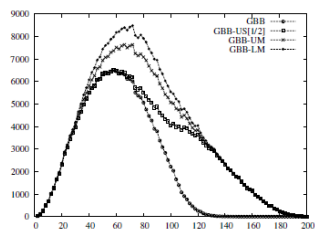


Figure 7. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

### EDF Particionado

- Para selecionar um representante do escalonamento particionado diversos esquemas de particionamento baseados em EDF foram avaliados
- Em cada caso as tarefas foram alocadas aos processadores de acordo com a heurística first-fit na ordem de alguma métrica tal como o deadline relativo ( $d_i$ )
- Dois testes de “fit” foram usados:
- (GF) o teste suficiente de [5]
- (BHR) o teste necessário e suficiente de Baruah et al.[9]

### EDF Particionado

- A força do teste GF está em sua simplicidade
- Em contraste, o limite superior no pior caso para a complexidade do teste BHR é o MMC dos períodos das tarefas
- Para evitar tempos de execução longos, o teste BHR somente foi aplicado em conjuntos de tarefas que não podiam ser revolvidos usando estes dois testes simples

1. A task set is feasible if  $\sum_{i=1}^N \frac{c_i}{\min\{d_i, T_i\}} < 1$
2. A task set is not feasible if  $\sum_{i=1}^N u_i > 1$

- Com isto, o teste BHR convergiu rapidamente para todos os conjuntos de tarefas testados
- O teste BHR pode ser impraticável devido a overflow de inteiros ou tempo de computação muito longo

### EDF Particionado

- Foram então usadas 3 heurísticas de ordenamento:

- (FFD-U) Utilização individual decrescente
- (FFD-L) Densidade  $c_i / \min\{d_i, T_i\}$  decrescente
- (FFD-D) Deadline relativo crescente

### EDF Particionado, $D \leq P$ , bimodal, 4 CPUs

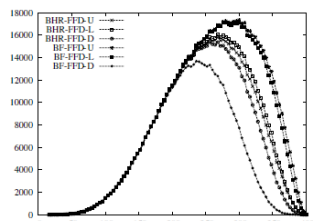


Figure 8. Constrained deadline, bimodal utilization, 4 CPUs

### EDF Particionado, $D \leq P$ , exponencial, 2 CPUs

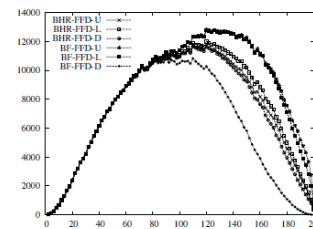


Figure 9. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

### EDF Particionado, $D$ qualquer, exponencial, 4 CPUs

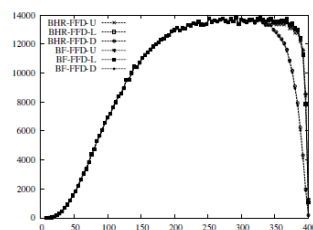


Figure 10. Unconstrained deadline, exponential utilization w/mean 0.25, 4 CPUs

### EDF Particionado

- Para todos os datasets GF-FFD-U and GF-FFD-L deram os melhores resultados
  - Com cada um indo melhor em diferentes datasets
- O esquema GF FFD-L foi escolhido para comparação com o escalonamento global

### Particionado versus Global

- Figuras 11-16 mostram o desempenho dos esquemas GBB e GBB-LM híbrido contra o esquema GF particionado
- O padrão exibido nestes exemplos persiste para todos os 24 datasets
- Em todos os casos o esquema de escalonamento híbrido melhorou a taxa de sucesso significativamente com respeito ao EDF puro
- Mas ainda teve uma taxa de sucesso pior que com escalonamento particionado

### Particionado versus Global

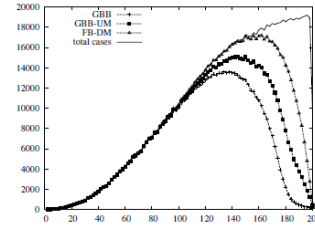


Figure 11. Unconstrained deadline, bimodal utilization, 2 CPUs

### Particionado versus Global

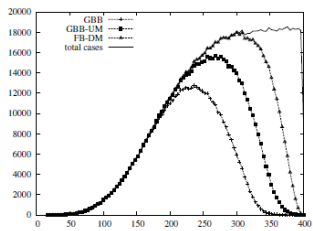


Figure 12. Unconstrained deadline, bimodal utilization, 4 CPUs

### Particionado versus Global

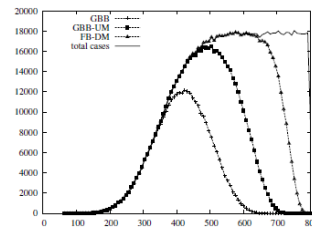


Figure 13. Unconstrained deadline, bimodal utilization, 8 CPUs

### Particionado versus Global

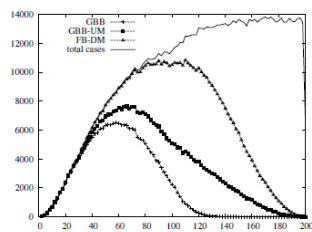


Figure 14. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

### Particionado versus Global

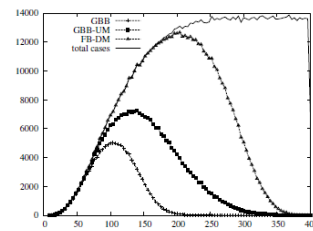


Figure 15. Unconstrained deadline, exponential utilization w/mean 0.25, 4 CPUs



## Particionado versus Global

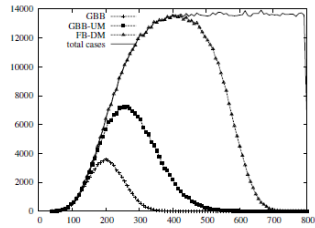


Figure 16. Unconstrained deadline, exponential utilization w/mean 0.25, 8 CPUs

## Conclusões 1/4

- Os testes de escalabilidade disponíveis para EDF global tem melhorado significativamente
- Entretanto, a abordagem global ainda não mostrou ser a melhor
- Escalonamento particionado ainda aparece como mais vantajoso do que o melhor teste de escalabilidade para escalonamento global
- Isto com respeito a chance estatística de ser capaz de escalonar um conjunto de tarefas arbitrário
- Além de tudo, alocação estática de tarefas tem overhead menor em tempo de execução

## Conclusões 2/4

- Este estudo não termina a questão Escalonamento Global versus Escalonamento Particionado
- Progresso adicional na análise do escalonamento EDF Global parece possível
- Mesmo que EDF Global não mostre-se competitivo contra o escalonamento EDF particionado
- Existem outros esquemas de escalonamento global a serem considerados
- Alguns desses esquemas podem garantir escalabilidade no pior caso a níveis de utilização do processador maiores do que a utilização  $(m + 1)/2$  que é o limite no pior caso para prioridade fixa por job

## Conclusões 3/4

- Existem muitas variações do conceito PFAIR
- Baruah, Cohen, Plaxton e Varvel[7, 8] mostraram que o escalonamento PFAIR
  - é ótimo para escalonar tarefas periódicas em um multiprocessador
  - Tem um teste de escalabilidade que é de tempo linear, além de ser exato (necessário e suficiente)
  - E para tamanhos de quantum suficientemente pequenos pode garantir escalabilidade em níveis de utilização arbitrariamente perto de  $m$
- O maior problema com o escalonamento PFAIR é a necessidade de dividir o tempo em pequenos quanta
- E o conseqüente grande overhead na implementação

## Conclusões 4/4

- Escalonamento global tem overhead maior em pelo menos 2 aspectos
- O atraso devido a contenção e sincronização em torno de uma única fila de aptos é maior do que quando existe uma fila por processador
- O custo de retomar a execução de uma tarefa pode ser maior se ocorrer em um outro processador
  - Tratamento de interrupções entre processadores
  - Repovoamento da cache
- Apenas experimentação com implementações reais pode gerar uma conclusão definitiva sobre a seriedade desses overheads
- E como eles pesam contra qualquer vantagem que o escalonamento global pode oferecer para o cumprimento de deadlines em aplicações reais