

# Isolamento Temporal para Análise Temporal Probabilista Baseada em Medições em Processadores *Multi-Core* Aleatorizados

Luís Fernando Arcaro

luis.arcaro@posgrad.ufsc.br

Rômulo Silva de Oliveira

romulo.deoliveira@ufsc.br

Universidade Federal de Santa Catarina (UFSC)  
Departamento de Automação e Sistemas (DAS)

Florianópolis, SC, Brasil  
Outubro de 2016



1 / 34

## Conteúdo

- 1 Introdução
- 2 ATPBM com TVE
- 3 IT para ATPBM
- 4 Proposta de Tese
- 5 Perguntas



2 / 34

## Introdução

### Sistemas de Tempo Real (STRs) [1]:

- São sistemas computacionais que estão submetidos, além de a requisitos de natureza lógica, a requisitos de natureza temporal;
- Os resultados devem estar corretos não somente do ponto de vista lógico, mas também devem ser gerados no intervalo de tempo correto;
- Aplicações com requisitos de tempo real têm se tornado cada vez mais comuns e variam muito em tamanho, complexidade e criticalidade:
  - Simples controladores embutidos em utilidades domésticas (leitor de *CD*);
  - Complexos e críticos sistemas relacionados à aviação (controle de tráfego).
- São classificados de acordo com a criticalidade dos requisitos temporais:
  - STRs críticos ou *hard real-time*: Faltas temporais levam a consequências graves (econômicas, por exemplo) ou até mesmo catastróficas;
  - STRs não críticos ou *soft real-time*: Faltas temporais não têm consequências graves, mas reduzem a utilidade da aplicação.



3 / 34

## Introdução

### Sistemas de Tempo Real (STRs) [1]:

- Os requisitos temporais aos quais os STRs estão sujeitos são expressos em termos dos prazos (*deadlines*) nos quais os resultados devem ser gerados;
- Para STRs críticos é fundamental o oferecimento de garantias quanto ao atendimento desses prazos;
- Emprega-se testes de escalonabilidade que demonstram que, mesmo no pior caso, as tarefas cumprirão seus *deadlines*;
- Esses testes baseiam-se em parâmetros temporais que:
  - São impostos pelo contexto de operação (períodos  $T_i$  e *deadlines*  $D_i$ );
  - São derivados do *software* utilizado (*WCETs*  $C_i$  e *WCRTs*  $R_i$ ).
- Especialmente para STRs críticos, as estimativas de *WCETs* devem ser:
  - Seguras: O valor nunca deve ser subestimado;
  - Apertadas: A superestimação deve ser mínima.



4 / 34

## Introdução

### Sistemas de Tempo Real (STRs) [2, 3]:

A determinação de estimativas para *WCETs* pode ser realizada através de:

- Abordagens **estáticas**, a partir da análise do executável ou do código fonte da tarefa, levando em conta as características temporais do processador:
  - Permitem a obtenção de limites seguros para o *WCET*;
  - Requerem um modelo temporal abstrato do processador utilizado;
  - A ausência das informações abstraídas leva a superestimação;
  - Elementos de *hardware* modernos são inerentemente pouco previsíveis.
- Abordagens **baseadas em medições**, a partir da observação dos tempos de execução para um conjunto de entradas determinado com cuidado:
  - As condições para a ocorrência do *WCET* são de difícil previsão;
  - Medir todas as combinações de entradas possíveis é impraticável;
  - Em geral não oferecem **garantias** quanto à segurança do *WCET*;
  - Métodos estatísticos adequados **podem** prover confiabilidade suficiente, mas requerem que os tempos medidos sigam distribuições de probabilidade conhecidas e que as variáveis aleatórias associadas a essas medições sejam **independentes e identicamente distribuídas**.



5 / 34

## Introdução

### Sistemas de Tempo Real (STRs) [1, 2, 4]:

#### Pesquisas recentes:

- Abordagens **estáticas** são limitadas pelo uso de elementos especulativos modernos, podendo causar a negação da capacidade de até  $n - 1$  núcleos devido ao pessimismo exigido para oferecer garantias [5].
- Abordagens **baseadas em medição** permitem abstrair a complexidade do *hardware* [6], podendo viabilizar o uso de elementos complexos em STRs, porém carregam incertezas relacionadas ao ajuste dos dados ao modelo.

#### No futuro:

- O uso de elementos de *hardware* complexos será fundamental para suportar a demanda computacional das aplicações executadas em STRs, e portanto abordagens baseadas em medição precisarão ser empregadas.

#### Porém, para isso:

- É necessário o desenvolvimento de abordagens que garantam a aderência das medições obtidas a modelos que permitam a obtenção de estimativas com altos níveis de confiança (por exemplo *GEV*).



6 / 34

## Introdução

### Sistemas de Tempo Real (STRs) [1, 2, 4]:

- Os principais fatores que afetam o *WCET* de uma tarefa estão relacionados ao *hardware* do processador no qual ela é executada:
  - *Pipelines* são esvaziados e/ou atrasados para garantir coerência na execução;
  - Memórias *cache* apresentam latências diferentes para *hits* e *misses*;
  - *RAMs* dinâmicas têm latências diferentes para sequências de acesso diferentes;
  - Prever exatamente qual combinação desses eventos gera o *WCET* é difícil.
- Processadores *multi-core* usam recursos compartilhados entre núcleos para maximizar utilização, aumentar desempenho, e/ou reduzir custos:
  - Memórias *RAM*, memórias *cache*, barramentos, e pontes;
  - Os acessos a esses elementos precisam ser arbitrados de forma que apenas um núcleo tenha acesso em qualquer instante (serialização de acessos);
  - Elementos como memórias *cache* compartilhadas permitem que núcleos substituam dados inseridos pelos demais, aumentando a frequência de *misses*;
  - Essas **interferências temporais** induzidas via *hardware* comprometem o **isolamento temporal** entre tarefas executadas em diferentes núcleos.



7 / 34

## Introdução

### Sistemas de Tempo Real (STRs) [1, 2, 4]:

- A ausência de **isolamento temporal** compromete a confiabilidade das medições obtidas a partir de um determinado núcleo;
- A distribuição das medições realizadas em um determinado núcleo é alterada pelo comportamento das tarefas executadas nos demais:
  - As variáveis aleatórias associadas aos tempos do núcleo de interesse tornam-se **dependentes** daquelas associadas aos demais núcleos;
  - Os tempos medidos no núcleo de interesse tornam-se **diferentemente distribuídos** em função da interferência causada pelos demais.
- **Isolamento temporal** é uma característica fundamental para plataformas de *hardware* destinadas à análise de *WCET* baseada em medições;
- Caso contrário, os requisitos básicos da aplicação dos métodos estatísticos que baseiam as estimativas de *WCET* **não são atendidos**.



8 / 34

## Introdução

- **Isolamento Temporal (IT)**: Garantia de ausência de interferências temporais entre núcleos de um processador *multi-core*, permitindo a análise das tarefas executadas em cada núcleo separadamente.
- **Análise Temporal Probabilista Baseada em Medições (ATPBM)**: Análise temporal que aplica métodos estatísticos sobre medições de tempos de execução a fim de determinar limites confiáveis para *WCETs*.
- **Teoria dos Valores Extremos (TVE)**: Teoria estatística focada na análise dos valores extremos observados para um conjunto de amostras, a fim de projetar probabilidades de um determinado valor ser excedido.

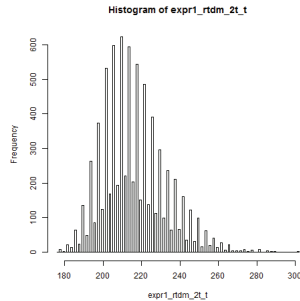


9 / 34

## ATPBM com TVE

### Análise Temporal Probabilista Baseada em Medições com TVE [6]:

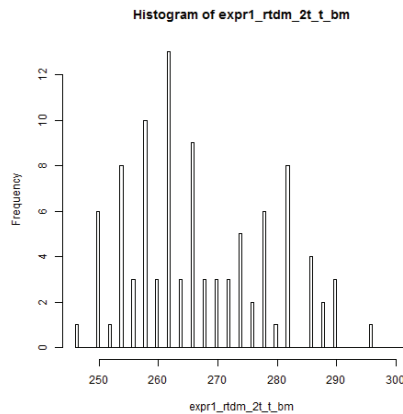
- 1) Coleta-se uma amostra composta de  $b$  blocos de tamanho  $n$  de tempos de execução da tarefa analisada:
  - Com dados de entrada que **acredita-se** gerar o maior tempo de execução, ou com entradas escolhidas de forma **aleatória**;
  - A decisão pelo número e tamanho dos blocos é crítica, porém ainda não foram estabelecidas regras gerais para definir esses parâmetros.



## ATPBM com TVE

### Análise Temporal Probabilista Baseada em Medições com TVE [6]:

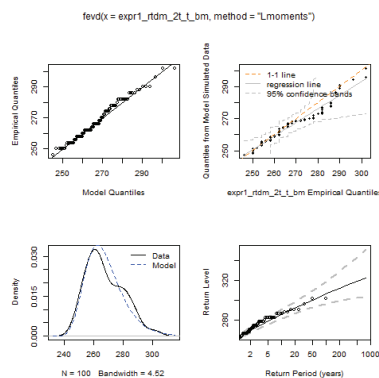
- 2) Obtém-se o valor máximo de cada bloco e descarta-se os demais valores, gerando então um conjunto reduzido de máximos de bloco:



## ATPBM com TVE

### Análise Temporal Probabilista Baseada em Medições com TVE [6]:

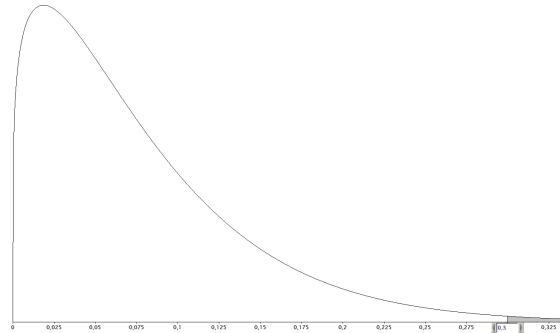
- 3) Ajusta-se uma distribuição Weibull, Gumbel, Fréchet, ou a família generalista dessas chamada *GEV*, aos máximos de bloco [7];
- 4) Aplica-se testes estatísticos a fim de evidenciar a aderência dos dados à distribuição e, assim, a validade dos resultados da análise.



## ATPBM com TVE

### Análise Temporal Probabilista Baseada em Medições com TVE [6]:

- 5) Obtém-se o *Probabilistic WCET* ( $pWCET$ ), ou seja, o valor do tempo de execução para o qual a área (integral) da cauda direita da curva de densidade de probabilidade ajustada é menor que a probabilidade máxima de perda de *deadline* pretendida (por exemplo  $10^{-16}$ ).



13 / 34

## ATPBM com TVE

### Análise Temporal Probabilista Baseada em Medições com TVE [6]:

- Parece poder ser aplicada desde que os tempos medidos possam ser modelados como distribuições de probabilidade conhecidas;
- Observar uma determinada distribuição de frequências **não prova** que os dados são gerados pela distribuição de probabilidades associada [8];
- Os testes estatísticos propostos até o momento são inexatos (necessários mas não suficientes), e também **não provam** a aderência à distribuição;
- Medições evidenciam a aderência dos tempos a uma distribuição específica, porém o sistema precisa de fato ter comportamento probabilístico;
- Plataformas de *hardware* comuns tendem a gerar tempos de execução que variam de forma não-suave e apresentam picos e vales patológicos.



14 / 34

## ATPBM com TVE

### Análise Temporal Probabilista Baseada em Medições com TVE [6]:

- Plataformas de *hardware* com comportamento pseudo-aleatorizado apresentam-se vantajosas nesse contexto [9]:
  - Geradores de números pseudo-aleatórios são utilizados para a geração de números binários com distribuição uniforme;
  - A semente dos geradores precisa ser inicializada por uma fonte externa confiável de números aleatórios, como as usadas para fins de criptografia;
  - Partes da cadeia de *bits* podem ser utilizadas para finalidades diversas, mas não podem ser reutilizadas, ao custo de gerar correlação (dependência) [10];
  - A decisão pela execução ou não de ações tipicamente especulativas ou dependentes do histórico de execução é tornada pseudo-aleatória;
  - Eventos que levam a tempos de execução patológicos tornam-se extremamente raros, pois passam a depender de uma cadeia longa de eventos "aleatórios";
  - Duas execuções do mesmo programa com a mesma entrada passam a ter comportamentos temporais diferentes dependendo desses eventos.



15 / 34

## IT para ATPBM

### Isolamento Temporal para ATPBM

- Como obter isolamento temporal em *multi-cores* não aleatorizados?
  - Memórias *cache* compartilhadas: particionamento entre núcleos [11];
  - Barramentos compartilhados: políticas de arbitragem isoladas *non-work-conserving* independentes de comportamento (*TDM*) [12];
  - *Networks-on-Chip* (*NoCs*): políticas de encaminhamento de latência máxima previsível e isoladas possivelmente sobre canais virtuais [13];
  - Em geral, utiliza-se alguma forma de reserva de recursos para os núcleos, reduzindo a largura de banda disponível a todos deles (particionamento);
  - Os núcleos acessam recursos compartilhados de forma exclusiva em momentos conhecidos e, portanto, com latência máxima previsível para análise estática;
  - Associadas, essas técnicas podem reduzir o problema de análise estática de *WCET* em *multi-core* ao caso *single-core* [14], mas limitam o desempenho.



16 / 34

## IT para ATPBM

### Isolamento Temporal para ATPBM

- Como obter isolamento temporal em *multi-cores* aleatorizados?
  - As mesmas abordagens são possíveis, porém durante as medições o pior caso precisa ser induzido para obter segurança, causando grande pessimismo;
  - Porém não precisa ser tão rígido, uma vez que “apenas” as distribuições devem ser mantidas e interferências aleatórias são permitidas;
  - Interferências aleatórias podem inclusive melhorar o comportamento estatístico dos tempos de execução ao induzir variabilidade **aleatória**;
  - Abordagens aleatorizadas são factíveis e **possivelmente** oferecem **isolamento temporal** e características razoáveis de desempenho:
    - Barramentos: o próximo cliente a ser servido é escolhido aleatoriamente [15] ou uma escala fixa permutada é gerada [16];
    - Memórias *cache* compartilhadas aleatorizadas podem ter a frequência de substituições limitada àquela simulada durante as medições [17].



17 / 34

## IT para ATPBM

### Isolamento Temporal para ATPBM

#### Definição

No contexto de processadores *multi-core* tradicionais uma tarefa executada em um núcleo pode ser considerada **temporalmente isolada** se seus tempos de execução não variam em função do comportamento das demais tarefas executadas paralelamente em outros núcleos.

#### Definição

No contexto de processadores *multi-core* analisados através de métodos probabilísticos, uma tarefa executada em um núcleo pode ser considerada **temporalmente isolada** se a distribuição de seus tempos de execução não varia em função do comportamento das demais tarefas executadas paralelamente em outros núcleos.



18 / 34

## IT para ATPBM

### Como evidenciar IT para ATPBM?

- Elaborou-se uma tarefa curta e simples, que avalia a expressão:
  - $0xAFFFFFFF \text{ xor } 0xFFAFFFFFF \text{ xor } 0xFFFF55FF \text{ xor } 0xFFFFF55$
- Obteve-se uma amostra executando apenas a tarefa analisada:
  - Sobre um dual-core com dois barramentos de memória (*ROM* e *RAM*) compartilhados e arbitrados em políticas aleatorizadas;
  - A tarefa executa no núcleo 0 enquanto o núcleo 1 permanece em estado de espera (*reset*), não sofrendo portanto interferências.
- Obteve-se uma amostra executando a tarefa analisada nos dois núcleos:
  - A tarefa é disparada tanto no núcleo 0 quanto no 1, e ambos os tempos de execução são medidos assim que as tarefas são concluídas (nanossegundos);
  - A variabilidade dos tempos decorre **apenas** dos acessos ao barramento compartilhado que dá acesso à *ROM*, pois não há acesso à *RAM*.
- Comparou-se os resultados obtidos através de dois testes estatísticos:
  - Teste Kolmogorov-Smirnov de mesma distribuição para duas amostras, que tem como hipótese nula que as distribuições são iguais;
  - Teste de hipótese de comparação de duas médias, que tem como hipótese nula que as médias populacionais das amostras são iguais.
- As hipóteses nulas não podem ser rejeitadas se os valores  $p$  obtidos nos testes forem maiores que o nível de significância  $\alpha = 0.01 = 1\%$ .

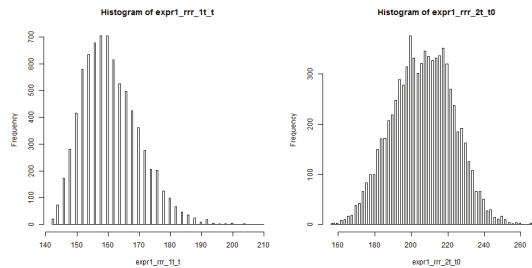


19 / 34

## IT para ATPBM

### Como evidenciar IT para ATPBM?

Barramento arbitrado em política *RRR* (*Random Round-Robin*):



$$\bar{x}_1 = 161.1074, s_1 = 9.213555, \bar{x}_2 = 206.88 \text{ e } s_2 = 16.62795$$

- KS: Obteve-se supremo  $D = 0.9104$  e valor  $p < 0.22^{-15}$ ;
- TH: Obteve-se valor  $p = 0$ .
- **Há evidências de diferença** entre as duas distribuições ao nível de significância de 1%, e portanto de que a política *RRR* **não oferece** isolamento temporal (o que era esperado).

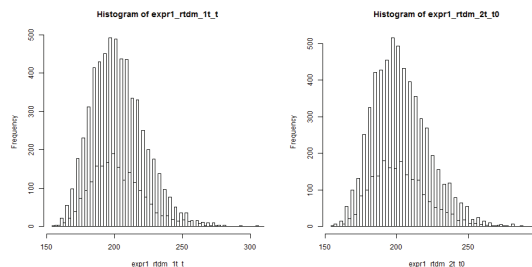


20 / 34

## IT para ATPBM

### Como evidenciar IT para ATPBM?

Barramento arbitrado em política *RTDM* (*Random TDM*):



$$\bar{x}_1 = 203.6785, s_1 = 19.83763, \bar{x}_2 = 203.6333 \text{ e } s_2 = 19.62019$$

- KS: Obteve-se supremo  $D = 0.0079$  e valor  $p = 0.9662$ ;
- TH: Obteve-se valor  $p = 0.8862$ .
- **Não há evidências** de diferença entre as duas distribuições ao nível de significância de 1%, e portanto de que a política *RTDM* **não oferece** isolamento temporal (o que também era esperado).







21 / 34




## Perguntas



## Referências I





-  J.-M. Farines, J. da Silva Fraga, and R. S. de Oliveira, *Sistemas de Tempo Real*. IME-USP, 2000, São Paulo (SP), 24-28 de Julho.
-  R. Wilhelm, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, P. Stenström, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, and R. Heckmann, "The Worst-Case Execution-Time Problem - Overview of Methods and Survey of Tools," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, pp. 36:1–36:53, 2008.
-  C. Cullmann, C. Ferdinand, G. Gebhard, D. Grund, C. Maiza, J. Reineke, B. Triquet, and R. Wilhelm, "Predictability Considerations in the Design of Multi-Core Embedded Systems," in *Embedded Real Time Software and Systems Conference 2010 (ERTS'10)*, vol. 807, 2010, pp. 36–42.
-  J. Engblom, "Processor pipelines and static worst-case execution time analysis," Ph.D. dissertation, Uppsala University, 2002.

## Referências II

-  N. Kim, B. C. Ward, M. Chisholm, C.-Y. Fu, J. H. Anderson, and F. D. Smith, "Attacking the One-Out-Of-m Multicore Problem by Combining Hardware Management with Mixed-Criticality Provisioning," in *Real-Time and Embedded Technology and Applications Symposium 2016 (RTAS'16)*. IEEE, 2016, p. 15.
-  L. Kosmidis, E. Quiñones, J. Abella, T. Vardanega, C. Hernandez, A. Gianarro, I. Broster, and F. J. Cazorla, "Fitting processor architectures for measurement-based probabilistic timing analysis," *Microprocessors and Microsystems (MICPRO)*, p. 16, 2016.
-  F. J. Cazorla, T. Vardanega, E. Quiñones, and J. Abella, "Upper-bounding Program Execution Time with Extreme Value Theory," in *International Workshop on Worst-Case Execution Time Analysis 2013 (WCET'13)*, vol. 30. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013, pp. 64–76.



## Referências III

-  S. Coles, *An Introduction to Statistical Modeling of Extreme Values*, ser. Springer Series in Statistics. Springer, 2001. [Online]. Available: <http://link.springer.com/10.1007/978-1-4471-3675-0>
-  E. Mezzetti, M. Ziccardi, T. Vardanega, J. Abella, E. Quiñones, and F. J. Cazorla, "Randomized Caches Can Be Pretty Useful to Hard Real-Time Systems," *Leibniz Transactions on Embedded Systems (LITES)*, vol. 2, pp. 01:1–01:10, 2015.
-  I. Agirre, M. Azkarate-askasua, C. Hernandez, J. Abella, J. Perez, T. Vardanega, and F. J. Cazorla, "IEC-61508 SIL 3 Compliant Pseudo-Random Number Generators for Probabilistic Timing Analysis," in *Euromicro Conference on Digital System Design 2015 (DSD'15)*. IEEE, 2015, pp. 677–684.
-  S. Altmeyer, R. Douma, W. Lunniss, and R. I. Davis, "Evaluation of Cache Partitioning for Hard Real-Time Systems," in *Euromicro Conference on Real-Time Systems 2014 (ECRTS'14)*. IEEE, 2014, pp. 15–26.




## Referências IV

-  C. Pitter and M. Schoeberl, "A real-time Java chip-multiprocessor," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 10, pp. 9:1–9:34, 2010.
-  K. Goossens and A. Hansson, "The Aethereal Network on Chip After Ten Years: Goals, Evolution, Lessons, and Future," in *Design Automation Conference 2010 (DAC'10)*. ACM, 2010, pp. 306–311.
-  R. Mancuso, R. Pellizzoni, M. Caccamo, L. Sha, and H. Yun, "WCET(m) Estimation in Multi-core Systems Using Single Core Equivalence," in *Euromicro Conference on Real-Time Systems 2015 (ECRTS'15)*. IEEE, 2015, pp. 174–183.
-  K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "LOTTERYBUS: A new high-performance communication architecture for system-on-chip designs," in *Design Automation Conference 2001 (DAC'01)*. IEEE, 2001, pp. 15–20.



## Referências V

-  J. Jalle, L. Kosmidis, J. Abella, E. Quiñones, and F. J. Cazorla, "Bus designs for time-probabilistic multicore processors," in *Design, Automation and Test in Europe Conference and Exhibition 2014 (DATE'14)*. IEEE, 2014, pp. 1–6.
-  M. Slijepcevic, L. Kosmidis, J. Abella, E. Quiñones, and F. J. Cazorla, "Time-Analysable Non-Partitioned Shared Caches for Real-Time Multicore Systems," in *Design Automation Conference 2014 (DAC'14)*. ACM, 2014, pp. 198:1–198:6.
-  L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quiñones, and F. J. Cazorla, "Measurement-Based Probabilistic Timing Analysis for Multi-path Programs," in *Euromicro Conference on Real-Time Systems 2012 (ECRTS'12)*. IEEE, 2012, pp. 91–101.
-  G. Lima, D. Dias, and E. Barros, "Extreme Value Theory for Estimating Task Execution Time Bounds: A Careful Look," in *Euromicro Conference on Real-Time Systems 2016 (ECRTS'16)*, 2016, p. 12.

