

---

# Estimação do WCET usando Medições

**Fundamentos dos Sistemas de Tempo Real**

Rômulo Silva de Oliveira

eBook Kindle, 2018

[www.romulosilvadeoliveira.eng.br/livrotemporeal](http://www.romulosilvadeoliveira.eng.br/livrotemporeal)

Outubro/2018

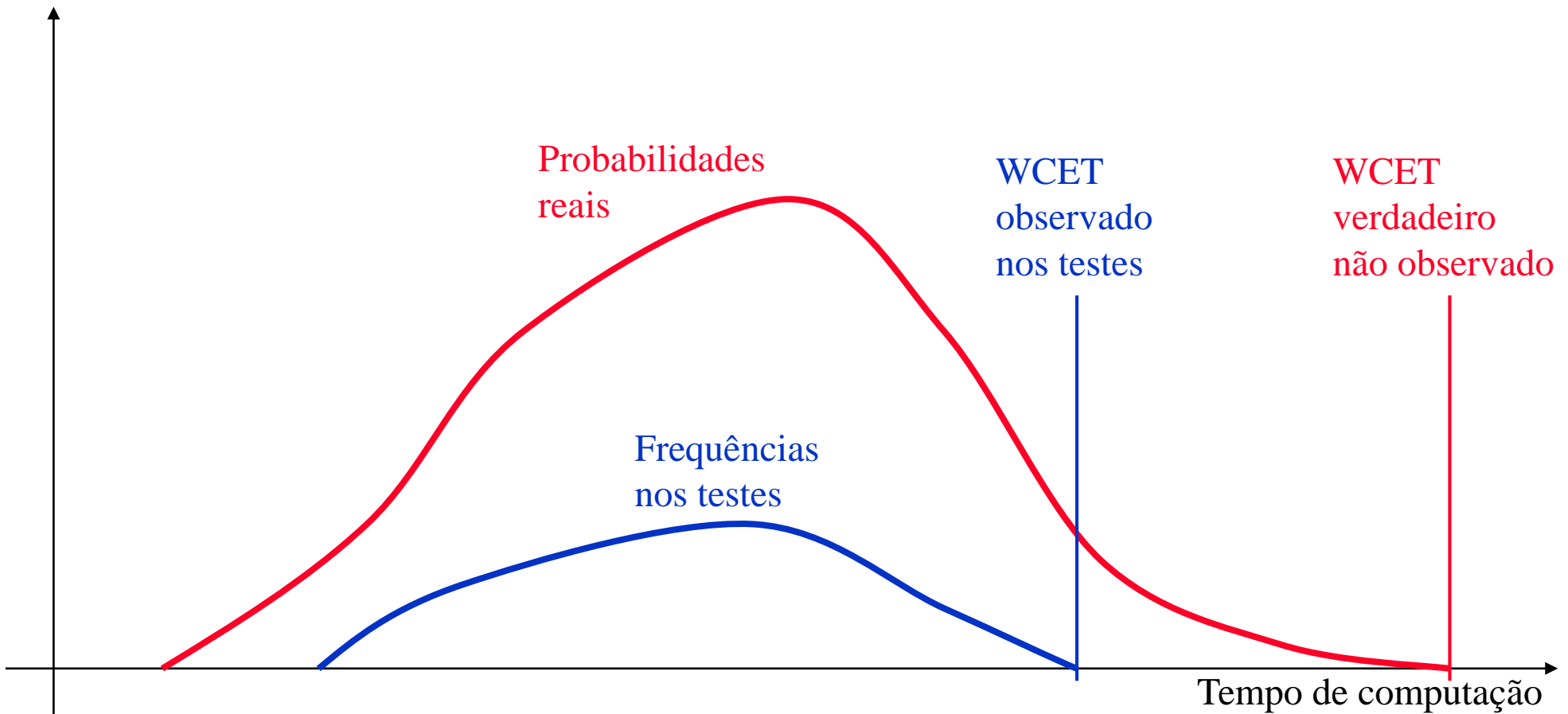


- Introdução
- Medição do Tempo de Execução
- Teste de Software Baseado em Busca
- Métodos Híbridos
- A Ferramenta RapiTime
- Métodos Estatísticos
- Métodos Estatísticos – Teoria dos Valores Extremos
- Métodos Estatísticos – TVE – Análise Experimental
- Considerações Gerais

- Para sistemas críticos é necessário conhecer o tempo de execução no pior caso de cada tarefa (WCET – Worst Case Execution Time)
  - Mesmo para sistemas não críticos é uma informação relevante
- O tempo de execução no pior caso supõe os piores comportamentos para o software e para o hardware
- Existe uma demanda crescente por capacidade de processamento
  - Também nos sistemas de tempo real
- Maior capacidade de processamento requer o uso de arquiteturas de computador modernas e complexas

- Arquiteturas de computador simples: tempo de execução de uma instrução depende apenas do tipo de instrução e operandos
  - Não existe pipeline, memória cache, etc
- Arquiteturas de computador complexas: apresentam efeitos temporais locais e efeitos temporais globais
- Elementos complexos de hardware melhoram o tempo médio de execução (Average-Case Execution Time – ACET) mas tornam a análise do WCET difícil
- Arquiteturas modernas:
  - Geram tempos de computação variáveis
  - Podem apresentar um comportamento patológico no pior caso
  - Pior caso tem probabilidade muito pequena, porém maior que zero

- Distribuição dos tempos de execução de uma tarefa
  - Probabilidades considerando todos os cenários e entradas possíveis  $P(c)$
  - Frequências observadas durante um certo número de testes  $F(c)$



- O método mais empregado na prática para estimar o tempo de execução no pior caso é simplesmente executar a tarefa
  - Mede o tempo de execução da tarefa sozinha
  - Para um certo número de cenários (os casos de teste)
- Com isto é obtido o máximo tempo de execução observado
  - Linha azul na figura
  - Também chamado de HWM (High Water Mark)
- Normalmente o máximo tempo de execução observado será menor que o verdadeiro (e desconhecido) WCET
  - Linha vermelha na figura
  - Na prática pode-se adicionar uma margem de segurança (20% por exemplo)

- Algumas abordagens medem o tempo de execução de diferentes partes do código, e não da tarefa inteira
- Estas medições do tempo de execução de segmentos são depois combinadas, da pior forma possível, para obter-se uma estimativa mais segura para o WCET
- Mesmo assim, não existe garantia de que a estimativa obtida é maior que o verdadeiro WCET

- As principais fontes de variância temporal são
  - O hardware do processador
  - O caminho de execução efetivamente medido
- Tanto os efeitos do hardware como o caminho de execução dependem diretamente ou indiretamente dos dados de entrada
- A qualidade das medições depende dos dados de entrada e do contexto de execução
- Na maioria dos casos o espaço de possibilidades é muito grande para ser completamente explorado, determinando assim o WCET
- Na maior parte da indústria o método convencional para estimar tempos de execução é medir o tempo de execução da tarefa para um subconjunto de todos os possíveis casos de execução



- Se os dados de entrada que levam ao pior caminho de execução e o pior estado inicial do hardware fossem conhecidos
  - Seria fácil
  - Bastaria medir o tempo de execução neste cenário
- Em geral, os dados de entrada e o estado inicial do processador no pior caso não são conhecidos

- Em arquiteturas de computador complexas
- Uma tarefa pode sofrer grandes interferências das outras atividades existentes no sistema (por exemplo, Linux)
- Isto torna ainda mais difícil determinar se um dado de entrada gera tempos de execução maiores que outro dado de entrada
- Existe uma interferência direta quando o sistema operacional suspende a tarefa em questão para executar outras tarefas de maior prioridade ou tratadores de interrupção
- Esta interferência é relativamente fácil de medir e não ser incluída no tempo de execução da tarefa

- Em arquiteturas complexas existem diversas interferências indiretas
  - Alteração da memória cache por outras tarefas
  - Alteração da TLB (Translation Lookaside Buffer) da MMU
  - Alteração da tabela com o histórico de saltos usada pelo branch predictor
- Estas alterações podem aumentar o tempo de execução de uma tarefa
  - Removem suas informações do hardware
  - Colocam informações relativas a outras tarefas
- Como são alterações difíceis de prever e detectar, podem mascarar os efeitos dos dados de entrada
- Dados de entrada que geram o pior tempo de execução são favorecidos em uma medição por, por acaso, pega poucas alterações decorrentes de outras atividades do SO

# Medição do Tempo de Execução 1/12

---

- Medições de tempo de execução
  - Execução de uma tarefa, ou partes dela, para um conjunto de entradas, em um hardware específico ou em um simulador
- Os tempos de operação dos diferentes componentes e procedimentos definidos em cada tarefa são avaliados
  - Só é possível medir uma rota de execução por vez
  - Cabe ao projetista encontrar as entradas com alta probabilidade de situações com o pior tempo de execução
- Apesar de sua utilização na indústria ser frequente
  - Perde em exaustividade
  - O maior tempo medido pode não ser o pior tempo de execução
- Podem gerar limites otimistas para o pior tempo de execução
- É possível ser aplicado a qualquer arquitetura de processador

## Medição do Tempo de Execução 2/12

---

- Os tempos de execução medidos são combinados e analisados
  - Normalmente por alguma forma de cálculo de limite
  - Para produzir estimativas do WCET
- As abordagens seguem uma metodologia em comum:
  - Determinar a entrada de pior caso
  - Executar e medir
  - Adicionar uma margem de segurança
  - Para que os dados de entrada selecionados sejam representativos, utiliza-se otimização combinatória, mas especificamente, métodos heurísticos

# Medição do Tempo de Execução 3/12

---

- Métodos baseados em medição
  - Executam a tarefa em um dado hardware ou simulador
  - Fazem isto para um conjunto de diferentes dados de entrada
  - Medem o tempo de execução da task (ou de partes dela)
- As medições do tempo de execução da tarefa
  - Produz uma distribuição das frequências dos tempos observados
  - Mas não produz nenhum limite superior para o tempo de execução
  - Não é garantido que o pior caso será observado nos testes
- Se as condições que geram o pior caso fossem conhecidas, bastaria uma execução

## Medição do Tempo de Execução 4/12

---

- Sempre é possível executar mais testes para medir o tempo de execução
  - Com mais variações de dados de entrada
  - Com mais variações de contexto de execução (estado inicial da cache, etc)
- Isto diminui o risco de resultados muito otimistas, mas não elimina este risco
- Testes para medir tempo de execução representam um custo adicional no processo de desenvolvimento
  - Amortecido se os testes foram aproveitados para outros propósitos também
- Teste exaustivo de todos os caminhos de execução possíveis para a tarefa é geralmente impossível

## Medição do Tempo de Execução 5/12

---

- Existem muitas formas de realizar as medições
- A abordagem mais simples é adicionar código à tarefa (instrumentar a tarefa) para obter e salvar o instante de tempo (timestamp) no qual a execução passa por determinado ponto da tarefa
  - Pode ser usado um relógio de tempo real
  - Pode ser usado um contador de ciclos do processador
- Existem técnicas que usam soluções combinadas de software e hardware
  - O software da tarefa é instrumentado
  - Mas a coleta das informações é feita por um hardware externo especialmente projetado para isto, para minimizar o impacto sobre a tarefa
- Mecanismos de medição totalmente transparentes (nonintrusive) são possíveis com instrumentos de hardware
  - Por exemplo analisadores lógicos



## Medição do Tempo de Execução 6/12

---

- Um método para teste de tempo de execução deve prover resposta para as seguintes questões
  - Quantos caminhos testar?
  - Quais caminhos testar?
  - Quantas vezes testar cada caminho individual?
  - Como gerar as entradas para forçar o caminho a ser testado?
  - Como estimar o grau de confiança sobre o valor WCET obtido?
  - Como orientar o desenvolvedor sobre a testabilidade do seu código e o que pode ser feito a este respeito?

## Medição do Tempo de Execução 7/12

---

- A determinação de quantos caminhos testar depende da tarefa
- A quantidade de caminhos que existem pode ser um número tratável ou intratável explicitamente
  
- Tarefa **sem** desvio e **sem** laço: existe apenas um caminho
- Tarefa **com** desvio mas **sem** laços: depende da combinação dos desvios, mas um número de caminhos tratável explicitamente
- Tarefa **sem** desvio mas **com** laço: depende da combinação dos laços, mas um número de caminhos tratável explicitamente
- Tarefa **com** desvio e **com** laço: um imenso número de caminhos, sendo intratável explicitamente

# Medição do Tempo de Execução 8/12

---

- Quantos caminhos testar?
  - Testes exaustivos de todos os caminhos de execução é geralmente impossível
- Análise de cobertura dos testes
  - Procura medir o quanto do código fonte é executado durante um conjunto de testes
  - Uma maior cobertura dos testes implica em uma maior probabilidade de observar tempos de execução mais próximos do pior caso
- Principais tipos de cobertura:
- Cobertura de funções
  - Todas as funções (subrotinas) foram executadas ?
- Cobertura de comandos
  - Todos os comandos (linhas) foram executados ?
- Cobertura de desvios
  - Todos os caminhos possíveis a partir de cada situação de desvio condicional (if, while, switch, etc) foram executados ?
- Cobertura de caminhos
  - Todos os caminhos possíveis ao longo da tarefa foram executados ?

## Medição do Tempo de Execução 9/12

---

- Quais caminhos testar?
  - Na literatura de engenharia de software a questão de quais caminhos testar está relacionada com a cobertura dos caminhos
- Uma grande dificuldade é encontrar casos de teste adequados, que garantam uma cobertura significativa
- No caso de teste temporal, o desafio está em encontrar os piores caminhos e garantir que estes tenham sido testados
- A seleção de caminhos não só é vital para encontrar dados de teste dado um caminho, mas para encontrar dados de teste bons

# Medição do Tempo de Execução 10/12

---

- Como gerar as entradas para forçar um caminho a ser testado?
- O que define qual caminho é executado são:
  - As variáveis de entrada do programa
  - As variáveis permanentes (globais) alteradas em execuções anteriores
- Alguns dos caminhos são semanticamente impossíveis pela semântica do programa ou pela semântica do ambiente
- Geralmente há uma diferença significativa do maior tempo de execução obtido considerando a entrada de dados no pior caso em relação à entrada aleatória
  - Existe a necessidade da geração de dados de entrada que executem os piores caminhos

## Medição do Tempo de Execução 11/12

---

- Quantas vezes testar cada caminho individual?
- Depende da tarefa em análise, e do estabelecimento de trade-offs entre confiabilidade e custo do teste
- Em qualquer projeto de software a etapa de testes é cara
- Os testes a serem aplicados não podem inviabilizar economicamente o desenvolvimento do software, mas os resultados devem ter uma certa confiabilidade
- Buscando extremos, um teste barato porém de baixa confiabilidade é executar a aplicação uma única vez, com dados de entrada aleatórios
- No outro extremo, temos os métodos analíticos de STR crítico, capazes de prover uma estimativa segura para o WCET

## Medição do Tempo de Execução 12/12

---

- Como estimar o grau de confiança sobre o valor WCET obtido?
- Estimar o grau de confiança sobre a estimação do WCET obtida é difícil na prática
- Está ligada ao conceito de cobertura dos testes
- Métodos estatísticos (visto mais adiante) podem fazer boas inferências, no entanto, existem algumas restrições para sua aplicabilidade

## Teste de Software Baseado em Busca 1/3

---

- Teste de software baseado em busca
  - SBST – Search-Based Software Testing
- A estimação do WCET de uma determinada tarefa pode ser entendida como um problema de busca complexo
- O resultado depende da arquitetura do computador onde a tarefa executa e de seus dados de entrada
- Este problema pode ser modelado como um problema de busca
  - Uma otimização ao contrário onde o objetivo é maximizar o tempo de execução
- Podem ser aplicadas soluções heurísticas para maximizar o tempo de execução e assim estimar o WCET



## Teste de Software Baseado em Busca 2/3

---

- Em um problema de otimização
  - Existe uma função objetivo
  - Existe um conjunto de restrições relacionados às variáveis de decisão
- O problema pode ser de minimização ou maximização da função objetivo (neste caso, maximização)
- A resposta para o problema de otimização é o ótimo global
  - Será o maior valor possível para a função objetivo desde que o valor atribuído às variáveis não viole nenhuma restrição
- Em alguns casos, chegamos a valores cuja alteração discreta não conduz a resultados melhores, mas que não são também o ótimo global (a essas soluções chamamos de ótimo local)

# Teste de Software Baseado em Busca 3/3

---

- Existem métodos heurísticos de diversos tipos
- Heurísticas de Construção
  - Uma ou mais soluções são construídas elemento a elemento, seguindo algum critério heurístico de otimização, até que se tenha uma solução viável
- Heurísticas de Busca em Vizinhança
  - Partem de uma solução inicial viável, tentando melhorar esta solução através de operações de troca, remoção ou inserção
- Heurísticas Sistemáticas
  - Árvore de espaço de soluções é percorrida utilizando critérios de ramificação e corte
- Meta-heurísticas
  - Heurísticas genéricas mais sofisticadas, uma heurística mais simples é gerenciada por um procedimento que visa explorar a instância do problema e o seu espaço de soluções
- Heurísticas Híbridas
  - Resultantes da combinação de duas ou mais heurísticas com estratégias diferentes

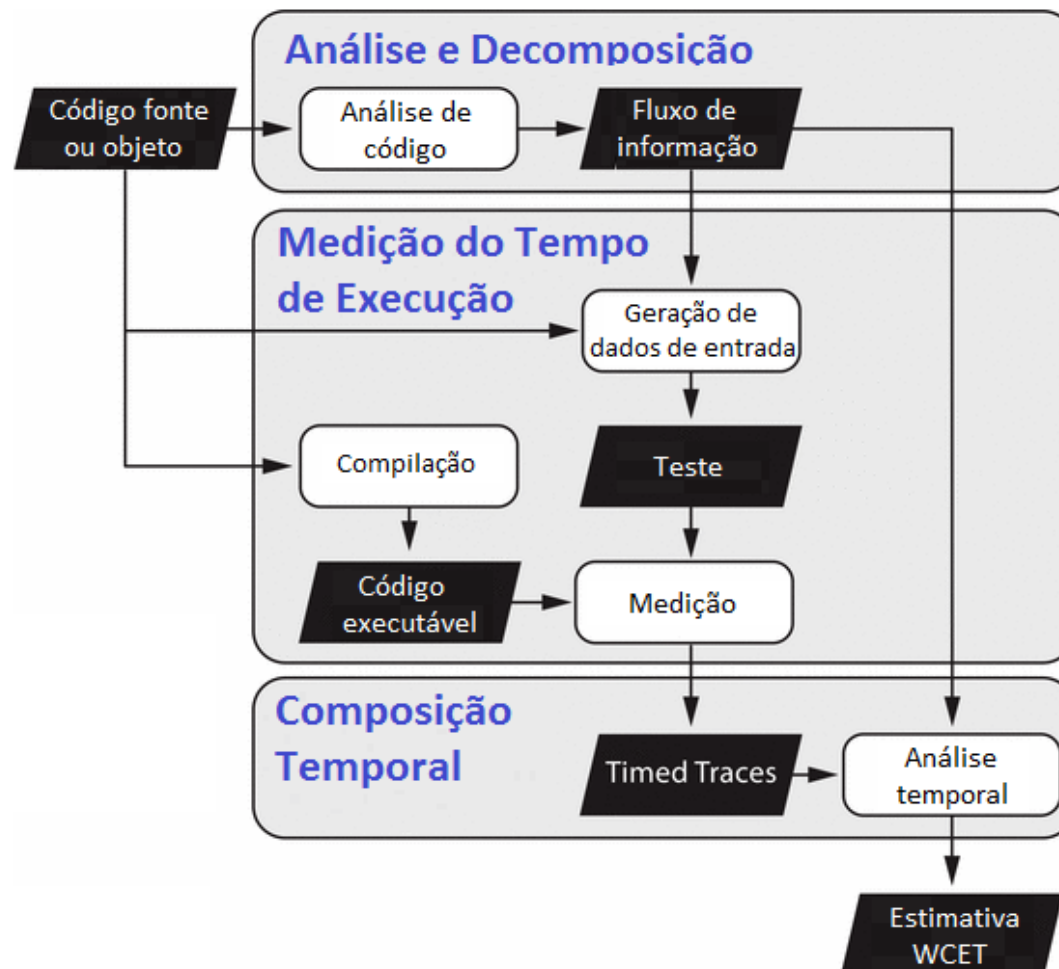
- Com o objetivo de obter um WCET
  - não tão pessimista como os gerados por análise estática
  - não tão otimista como os gerados por medições
- Existe a possibilidade de melhorar as estimativas do WCET de tarefas através da utilização de uma abordagem híbrida

- A análise WCET híbrida é uma combinação de ambas abordagens
- Objetivo de gerar um valor mais preciso para o pior caso de tempo de execução
- Abordagens de análise WCET híbrido genéricas incluem os cinco passos seguinte:
  - Análise estática do código do programa
  - Fluxo de controle particionado
  - Execução de caminho
  - Medições em tempo de execução
  - Cálculo do limite do WCET

## Métodos Híbridos 3/4

- É medido o tempo de execução de segmentos de código
  - Tipicamente de blocos básicos
- Os tempos de execução medidos são então combinados e analisados
  - Para produzir uma estimativa do WCET
- Medições substituem a análise do comportamento do processador (métodos estático)
- Escolha dos caminhos de execução pode ser da mesma forma que na análise estática
  - Análise de fluxo de controle encontra todos os caminhos possíveis
  - Os tempos medidos para cada segmento de código são combinados em uma estimativa total
- Esta solução inclui todos os possíveis caminhos de execução, mas pode produzir resultados inseguros se o tempo de execução medido para cada segmento for menor que o seu pior caso
- Apenas um subconjunto de todos os possíveis contextos de execução será usado para a medição de cada segmento de código

- Estimativa através de um método híbrido
  - Fonte: Adaptado de Bunte, Zolda e Kirner (2011)



## A Ferramenta RapiTime 1/5

---

- RapiTime é uma ferramenta de software comercializada pela empresa Rapita Systems de York, Reino Unido
  - [www.rapitasystems.com/products/rapitime](http://www.rapitasystems.com/products/rapitime)
- RapiTime combina análise estática e dinâmica do código em sua plataforma alvo para prover informação detalhada sobre seu comportamento temporal
- Medições obtém informações sobre quanto tempo um particular segmento de código demora para executar na plataforma alvo
- Os resultados da medição são combinados de acordo com a estrutura do programa para determinar uma estimativa para o tempo de execução no pior caso
- RapiTime é capaz de computar a distribuição de probabilidade do tempo de execução do caminho mais longo assim como segmentos menores do código

## A Ferramenta RapiTime 2/5

---

- O código a ser analisado pode ser o código fonte, nas linguagens C, C++, ADA ou linguagem de montagem (assembly language)
- O resultado da análise são diversos relatórios em vários formatos
  - Descrição da estimativa do WCET
  - Tempos de execução realmente medidos
  - Dados separados para cada função ou segmento de código especificado pelo usuário



## A Ferramenta RapiTime 3/5

---

- A informação temporal pode ser capturada no sistema em execução através de diferentes formas
  - Bibliotecas com instrumentação puramente via software
  - Instrumentação via software pouco intrusiva auxiliada por hardware externo, como CAN, porta serial, Ethernet ou depuradores
  - Mecanismos completamente não intrusivos de monitoração via hardware próprio
  - Rastros de execução a partir de simuladores do processador
- O usuário pode adicionar anotações no código para guiar a instrumentação e a análise dos dados
- Existe suporte para ponteiros para funções, recursão, e estruturas de código complexas
  - algo normalmente não disponível em análise estática

## A Ferramenta RapiTime 4/5

---

- A ferramenta RapiTime não necessita de um modelo do processador
- Então, em princípio, pode lidar com qualquer tipo de processador, mesmo com
  - Execução fora de ordem
  - Múltiplas unidades de execução
  - Várias hierarquias de memória cache
  - Etc
- A limitação está na necessidade de extrair rastros de execução, os quais necessitam alguma forma de instrumentação e um mecanismo para extrair estes rastros da plataforma alvo para serem analisados

## A Ferramenta RapiTime 5/5

---

- A lista de processadores suportados por RapiTime é extensa e pode ser obtida no site da empresa
- Várias versões de ARM
  - ARM7, ARM9, ARM10, ARM11, Cortex-M, Cortex-R, Cortex-A
- Vários processadores dos fabricantes
  - Analog Devices
  - Freescale Semiconductor
  - IBM
  - Infineon
  - Intel
  - NXP Semiconductors
  - Renesas (previously Hitachi, Mitsubishi, NEC)
  - Rockwell Collins
  - Rolls Royce
  - STMicroelectronics
  - Texas Instruments

- Estimativa do WCET através de métodos estatísticos
  - Para a análise estática o WCET é um valor fixo
- Quando o WCET é estimado com um método estatístico, são obtidas evidências estatísticas sobre qual seria este valor, mas não há certeza
- O resultado da análise estatística a partir de medições fornece duas informações
- Uma estimativa probabilística para o WCET, chamado pWCET
- E a probabilidade de que essa estimativa seja ultrapassada na prática
- A probabilidade é baseada no princípio de que o comportamento futuro tende a seguir um padrão similar ao que já foi observado
  - O qual pode ser extrapolado
- É possível fazer previsões inclusive para probabilidades de excedência muito menores do que seria possível ao examinar diretamente apenas os dados obtidos das medições

- **Análise Temporal Probabilística Baseada em Medições (ATPBM)**
- Abordagem proposta recentemente para derivar estimativas probabilísticas de Worst-Case Execution Time (WCET)
- Tais estimativas são também conhecidas na literatura como Probabilistic Worst-Case Execution Times (pWCETs)
- São estimativas de WCET que são excedidas com probabilidade máxima que é conhecida e extremamente baixa (por exemplo  $10^{-15}$ )
- Baseia-se na aplicação da Teoria dos Valores Extremos (TVE) sobre medições dos tempos de execução da tarefa analisada
- Mesmo em sistemas críticos (por exemplo aviões), a probabilidade de falhas de natureza não-temporal (por exemplo estrutural) não é nula
- Podendo o sistema falhar com certa probabilidade não-nula por causas diversas, o custo/esforço adicional para prover garantias absolutas para os requisitos temporais não se justifica

- Métodos estatísticos quando aplicado corretamente podem fornecer confiabilidade suficiente, para a estimativa do WCET obtido através de medição
- É necessário um bom ajuste dos dados ao modelo
- Os tempos medidos devem seguir distribuições de probabilidade conhecidas
- As variáveis aleatórias associadas às medições devem ser independentes e identicamente distribuídas
- Utiliza a **Teoria de Valores Extremos**

# Métodos Estatísticos – Teoria dos Valores Extremos 1/15

---

- **Teoria de Valores Extremos – TVE**
- TVE é uma estrutura de inferência estatística com o objetivo de descrever comportamento extremo raro
- Sua aplicação varia de hidrologia, financiamento, gestão de risco, telecomunicações e, recentemente, para derivar limites pWCET
- Uma estimativa do pior caso de uma tarefa pode ser obtida a um determinado nível de precisão estatística por meio do uso de
  - Medição para recolher dados
  - TVE para generalizar os processos que produzem estas medições
- Pode oferecer uma abordagem alternativa para testes convencionais, bem como para análise estática do tempo de execução

## Métodos Estatísticos – Teoria dos Valores Extremos 2/15

---

- **Premissas básicas para a aplicabilidade da TVE**
- Os tempos de execução observados devem poder ser descritos como variáveis aleatórias independentes e identicamente distribuídas (i.i.d.)
- Os tempos de execução devem ser produzidos por um processo aleatório e estacionário (a distribuição dos valores medidos não muda com o passar do tempo)
- É necessário que a distribuição dos tempos de execução possa ser ajustada a uma distribuição assintótica de valores extremos



## Métodos Estatísticos – Teoria dos Valores Extremos 3/15

---

- **Para provar que as premissas da TVE são atendidas pelas medições obtidas, vários testes estatísticos podem ser utilizados**
- Teste Kolmogorov-Smirnov (KS): avaliar se duas amostras proveem de uma mesma distribuição
- Teste Ljung-Box (LB): verificar a independência nas observações
- Teste Turning Point (TP): verificar se existe correlação entre observações vizinhas
- Teste Wald-Wolfowitz runs (runs): verificar a aleatoriedade da amostra. E se os eventos são amostrados de uma população estacionária
- Função acf: verificar a auto-correlação entre as observações

## Métodos Estatísticos – Teoria dos Valores Extremos 4/15

---

- TVE é normalmente referida como uma técnica baseada na medição e consiste de 5 etapas
- Medir quanto tempo uma tarefa leva para executar um conjunto representativo de seus possíveis dados de entrada
- Selecionar uma amostra de máximos a partir dessas medições
  - Apenas um subconjunto das medições é usado nos cálculos
- Selecionar um modelo de valor extremo que melhor se ajuste aos valores máximos amostrados
- Verificar se o modelo obtido é consistente com as medições realizadas
- Uma vez que o modelo derivado é validado, obtenção da estimativa de pior caso pWCET para uma dada probabilidade de excedência

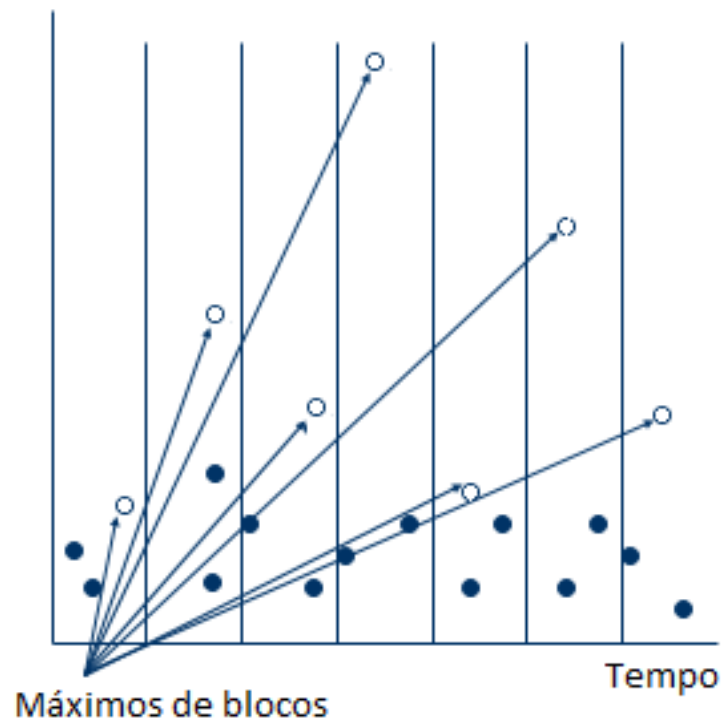
# Métodos Estatísticos – Teoria dos Valores Extremos 5/15

---

- **Amostragem de máximos**
- Em uma amostra qualquer só existe um máximo e um mínimo
- É necessário uma seleção de máximos da amostra para a análise
  - Achar uma curva que mostre a probabilidade de valores extremos serem observados
- Há duas abordagens principais para isto:
  - Máximos de Blocos
  - Picos Acima do Limiar

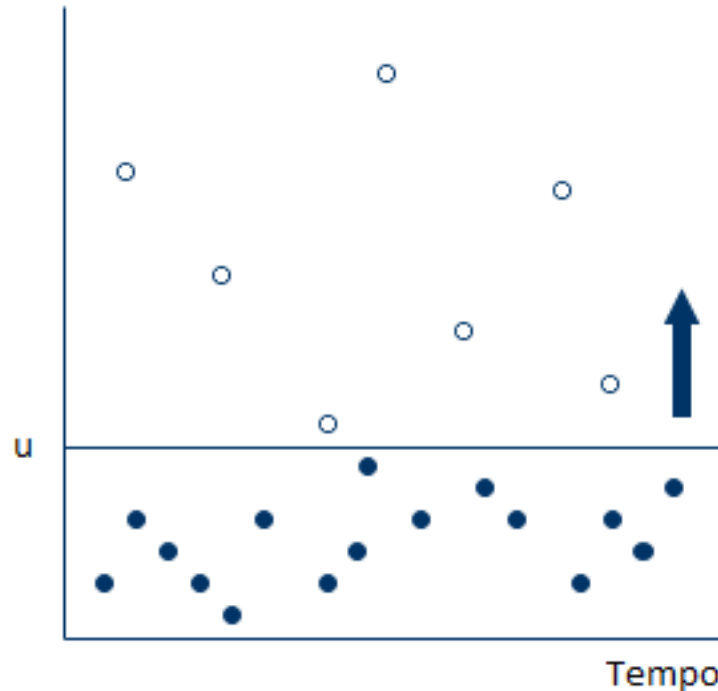
# Métodos Estatísticos – Teoria dos Valores Extremos 6/15

- **Máximos de Blocos (Block Maxima – BM)**
- O conjunto de dados amostrados (medições do tempo de execução realizadas) é particionado em blocos do mesmo tamanho
  - Por exemplo: 500.000 medições particionadas em 1000 blocos de 500 medições
- Seleciona-se apenas o valor máximo de cada bloco



# Métodos Estatísticos – Teoria dos Valores Extremos 7/15

- **Picos Acima do Limiar (Peaks Over Threshold – POT)**
- É escolhido um limiar (threshold) para definir a partir de qual valor serão consideradas as medições (valores da cauda)
- Apenas as medições acima deste limiar ( $u$ ) são consideradas
- O método ajusta uma distribuição de probabilidade para os valores acima deste limiar



# Métodos Estatísticos – Teoria dos Valores Extremos 8/15

---

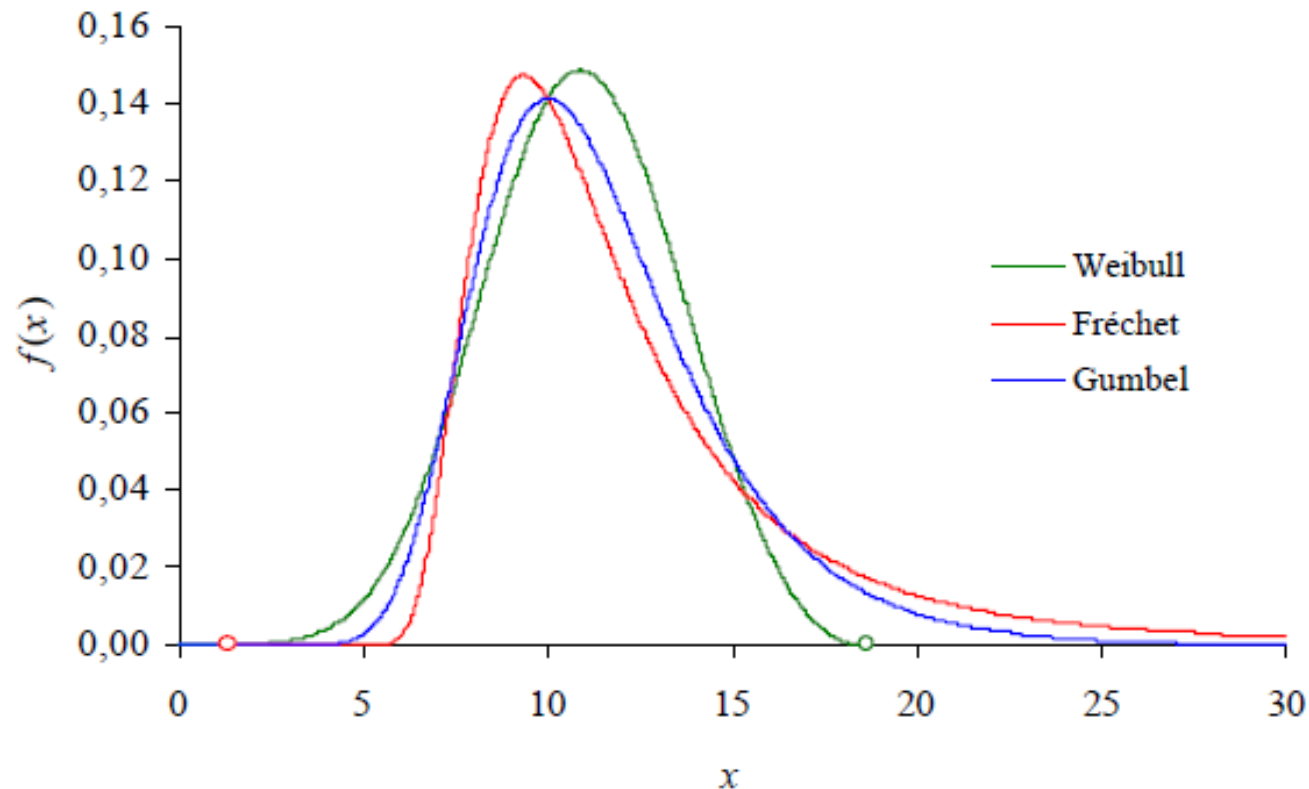
- **Modelo de Valor Extremo usando Máximos de Blocos (BM)**
- Uma amostra de máximos, selecionada via BM, deve ser ajustada para a distribuição Generalizada de Valores Extremos (GVE)
- GVE unifica as três distribuições Weibull, Gumbel e Fréchet, e tem função de distribuição acumulada de probabilidade dada por

$$F(x) = \exp \left\{ - \left[ 1 + \xi \left( \frac{x-\mu}{\sigma} \right) \right]^{\frac{1}{\xi}} \right\}$$

- Parâmetros:  $\mu$  localização,  $\sigma$  escala, e  $\xi$  forma
  - *Weibull*:  $\xi < 0$ , cauda curta delimitada
  - *Gumbel*:  $\xi = 0$ , cauda ilimitada ( $\xi \rightarrow 0$ )
  - *Fréchet*:  $\xi > 0$ , cauda pesada ilimitada

# Métodos Estatísticos – Teoria dos Valores Extremos 9/15

- Função densidade de probabilidade da distribuição GVE
  - Weibull  $\xi=-0,3$  Gumbel  $\xi->0$  e Fréchet  $\xi = 0,3$
  - com  $\mu=10$  e  $\sigma=2,6$



## Métodos Estatísticos – Teoria dos Valores Extremos 10/15

- **Modelo de Valor Extremo usando Picos Acima do Limiar (POT)**
- Uma amostra de máximos, selecionada via POT, deve ser ajustada para a distribuição Generalizada de Pareto (GP)
- A GP é uma aproximação para a distribuição dos excessos acima de um threshold elevado  $u$ . A função de distribuição GP é dada por

$$H_y(x|u, \sigma_u) := \begin{cases} 1 - (1 + \xi \frac{x-u}{\sigma_u})^{\frac{-1}{\xi}}, x \geq u & \text{se } \xi > 0 \\ 1 - \exp(-\frac{x-u}{\sigma_u}), x \geq u & \text{se } \xi = 0 \\ 1 - (1 + \xi \frac{x-u}{\sigma_u})^{\frac{-1}{\xi}}, u \leq x \leq u - \frac{\sigma_u}{\xi} & \text{se } \xi < 0 \end{cases}$$

- Parâmetros: *threshold*  $u$  localização,  $\sigma$  escala, e  $\xi$  forma
  - *Beta*:  $\xi < 0$
  - *Exponencial*:  $\xi = 0$
  - *Pareto*:  $\xi > 0$



# Métodos Estatísticos – Teoria dos Valores Extremos 11/15

---

- **Estimação dos Parâmetros**
- A partir dos valores selecionados via BM (ou POT), são estimados os parâmetros que definem a curva GEV (ou GP) que mais se ajusta aos valores observados nas medições
  - No caso GEV (ou GP) representa a distribuição de valores extremos que foi ajustada aos dados obtidos, as medições dos tempos de execução
  - A curva obtida de certa forma explica o comportamento destas medições, em termos probabilistas
- Existem diferentes métodos para estimar os parâmetros da distribuição de valores extremos. Os métodos mais usuais são
  - Maximum-likelihood estimation (MLE)
  - Generalized maximum-likelihood estimation (GMLE)
  - L-moments
  - Bayesian

# Métodos Estatísticos – Teoria dos Valores Extremos 12/15

---

- **Verificação do Modelo Obtido**
- Uma vez que os parâmetros do modelo são estimados, os testes de qualidade de ajuste (goodness-of-fitness) são usados para verificar a adequação (aderência) do modelo obtido com relação aos dados
  - O quão bem a curva GEV (ou GP) obtida se ajusta às medições realizadas
- Estão disponíveis testes estatísticos para avaliar a qualidade de ajuste dos modelos estimados
- A verificação pode ser:
- Visual (gráficos Quantile-Quantile)
- Técnicas matemáticas (avaliação numérica para a precisão)

## Métodos Estatísticos – Teoria dos Valores Extremos 13/15

---

- **Obtenção do pWCET para uma dada probabilidade de excedência**
- A obtenção da estimativa de pior caso pWCET para uma dada probabilidade de excedência é feita a partir da simples leitura da curva (modelo) obtido
- Para cada valor pWCET do eixo-X da curva GEV (ou GP), o eixo-Y fornece a respectiva probabilidade de excedência
- Pode-se escolher uma probabilidade de excedência, por exemplo de  $10^{-9}$  (eixo-Y) e localizar o valor pWCET correspondente (eixo-X)
- Pode-se escolher um valor para o pWCET (eixo-X) e obter a probabilidade de excedência (eixo-Y), ou seja, a probabilidade de ser observado um tempo de execução superior ao valor pWCET

# Métodos Estatísticos – Teoria dos Valores Extremos 14/15

---

- **Em resumo**
- Mede-se tempos de execução da tarefa-alvo na plataforma-alvo
- As condições de medição devem se igualar ou ser pessimistas em relação ao pior caso esperado no ambiente de operação
- Amostras devem ser representativas com relação às situações extremas esperadas no ambiente em que o sistema irá operar
- Os tempos medidos são sujeitos aos requisitos da TVE, como i.i.d. e aderência dos máximos a modelos de valores extremos
- Mesmo que todos os requisitos da TVE sejam satisfeitos, os resultados produzidos podem provar-se não confiáveis.
- Condições de medição e características construtivas do sistema analisado impactam diretamente na confiabilidade dos resultados

# Métodos Estatísticos – Teoria dos Valores Extremos 15/15

---

- **Em resumo**
- Tempos de execução diferem fenomenologicamente daqueles para os quais a TVE foi inicialmente criada (e.g. nível do mar)
- Variabilidade temporal é induzida por fatores controláveis e/ou especulativos, e pode portanto não ter comportamento aleatório
- As principais fontes de variabilidade temporal são:
  - O hardware do processador utilizado
  - Os caminhos de execução que são efetivamente medidos
- Tanto os efeitos do hardware quanto dos caminhos de execução dependem direta ou indiretamente dos dados de entrada usados

# Métodos Estatísticos – TVE – Análise Experimental 1/9

---

- **On Using GEV or Gumbel Models when Applying EVT for Probabilistic WCET Estimation**
  - Karila Palma Silva, Luís Fernando Arcaro, Rômulo Silva de Oliveira
  - RTSS'2017
- Estudo experimental sobre a aplicação de TVE para obter pWCET
- A tarefa escolhida para realização da análise foi a **Bsort**
  - ordena um vetor de 10 elementos pelo método bubble sort
- O caminho de execução foi fixado para aquele que executa o maior número possível de operações elementares

## Métodos Estatísticos – TVE – Análise Experimental 2/9

---

- Os tempos de execução foram medidos em uma plataforma de hardware aleatorizada em tempo, executada a 50MHz na FPGA.
  - processador dual-core
  - Memória RAM separada (instruções e dados)
  - barramento compartilhado arbitrado com uma política aleatória
  - Pipeline de cinco estágios que implementa instruções MIPS
  - time-randomized set-associative cache memory com políticas de posicionamento e substituição aleatórias

# Métodos Estatísticos – TVE – Análise Experimental 3/9

---

- A tarefa foi executada de forma exclusiva e ininterrupta.
- Aplicação de máximos de blocos com
  - Tamanho da amostra: 50000
  - Tamanho de bloco: 50
- Testes de i.i.d:
  - Os resultados dos testes estatísticos foram aceitáveis



# Métodos Estatísticos – TVE – Análise Experimental 4/9

---

- **Histogramas crus:**
- Foi coletada uma amostra composta de 1000 blocos de tamanho 50 de tempos de execução da tarefa analisada
  
- **Histograma de máximos:**
- Considerou-se o valor máximo de cada bloco e descartou-se os demais valores
  - gerando um conjunto reduzido de máximos de bloco

## Métodos Estatísticos – TVE – Análise Experimental 5/9

---

- **Ajuste de modelo:**
- Ajustou-se os máximos de bloco as distribuições GEV e Gumbel.
- Os parâmetros das distribuições foram ajustados através dos métodos L-moments para GEV e Maximum Likelihood Estimation (MLE) para Gumbel

## Métodos Estatísticos – TVE – Análise Experimental 6/9

---

- **Verificação de ajuste:**
- Aplicou-se testes estatísticos a fim de evidenciar a aderência dos dados à distribuição e, assim, a validade dos resultados.
- Para isso, utilizou-se os gráficos de comparação de quantis e gráficos de comparação de probabilidades.
- O modelo GEV representa corretamente os tempos de execução máximos observados, ressaltando que é susceptível de aderir a uma das distribuições Weibull, Gumbel ou Fréchet.
- O modelo de Gumbel não representa bem os máximos analisados, mas leva a limites superiores confiáveis para superestimar quantiles altos.

# Métodos Estatísticos – TVE – Análise Experimental 7/9

---

- **Estimativa do pWCET em ciclos de clock do processador**

● Modelo	GEV	Gumbel
● pWCET( $10^{-6}$ )	46638.9927	46801.7270
● pWCET( $10^{-7}$ )	46654.6620	46864.6327
● pWCET( $10^{-8}$ )	46667.2991	46927.5385
● pWCET( $10^{-15}$ )	46708.2801	47367.9007

## Métodos Estatísticos – TVE – Análise Experimental 8/9

---

- Os modelos GEV e Gumbel foram avaliados com relação à confiabilidade das estimativas pWCET produzidas
- Foram feitas  $10^8$  execuções e observado o HWM
- Em  $10^8$  execuções o HWM deve ficar abaixo do pWCET( $10^{-15}$ )
  
- Confiabilidade
  - Espera-se que as estimativas não excedam o maior valor observado, uma vez que a magnitude da amostra considerada é muito menor do que a probabilidade de excedência utilizada

# Métodos Estatísticos – TVE – Análise Experimental 9/9

---

- **Conclusões sobre a confiabilidade**
- **GEV**
  - Deixa dúvidas quanto à confiabilidade
  - $pWCET(10^{-15})$  converge para valores muito próximos dos maiores valores observados em  $10^8$  execuções
- **Gumbel**
  - É potencialmente confiável
  - Todos os  $pWCET(10^{-15})$  e seus intervalos de confiança permanecem acima dos maiores valores observados em  $10^8$  execuções

## Considerações Gerais 1/4

---

- Métodos estáticos computam um limite superior para o tempo de execução
- Análise de fluxo de controle permite considerar todos os possíveis caminhos de execução
- São usadas abstrações para cobrir todos as possíveis dependências de contexto de execução no processador em questão
  
- O preço pago pela segurança é a necessidade de modelos para o comportamento do processador e a possibilidade de resultados imprecisos
  - Superestimação do WCET
  
- A análise estática não requer a execução do sistema, o que poderia requerer aparatos complexos para simular o hardware e periféricos

## Considerações Gerais 2/4

---

- Métodos baseados em medição substituem a análise do comportamento do processador por medições
- Dificilmente todos os possíveis caminhos de execução são medidos
- Dificilmente o estado inicial do processador (cache, etc) no pior caso é conhecido
- Algumas possibilidades para o tempo de execução que dependam de contexto específico não serão observadas
  - O método não é garantido observar o WCET
- Métodos híbridos são possíveis onde análise de fluxo de controle é usada para garantir que todos os possíveis caminhos de execução foram considerados (medidos ou mostrados serem mais rápidos do que os medidos)
- Métodos baseados em medição são mais simples para aplicar em processadores modernos, não necessitam de um modelo para o comportamento do processador
- Eles produzem estimativas para o WCET que são mais precisas do que os métodos analíticos
  - Especialmente para processadores complexos e aplicações complexas
  - Mas é muito difícil estimar o quanto otimista é uma estimativa obtida via medições



## Considerações Gerais 3/4

---

- As técnicas baseadas em análise estática para estimação do WCET são limitadas no que se refere a plataformas de hardware suportadas e também estilo de programação
- As abordagens de teste convencional fornecem apenas estimativas grosseiras sobre o tempo de execução no pior caso
- Para suportar a demanda computacional das aplicações executadas em STR, o uso de hardware complexo será fundamental
- Portanto, métodos para orientar os testes a serem realizados e métricas que quantifiquem o conceito de confiabilidade para estimativas do WCET são necessários
- Uma possibilidade são os métodos estatísticos que agora começam a ser definidos

## Considerações Gerais 4/4

---

- Existem 3 principais dimensões a serem consideradas
- Aplicabilidade, dado que muitas técnicas de análise possuem aplicabilidade limitada em termos de hardware e design do software
- Confiabilidade dos valores de WCET estimados, ou seja, o grau de confiança de que o verdadeiro WCET foi observado ou pelo menos um valor suficientemente próximo dele, ou maior que ele
- Custo da análise, dado que em qualquer projeto de software a etapa de testes é uma etapa cara assim como a análise estática é cara, e a obtenção de uma estimativa para o WCET não pode inviabilizar economicamente o desenvolvimento do sistema

- Introdução
- Medição do Tempo de Execução
- Teste de Software Baseado em Busca
- Métodos Híbridos
- A Ferramenta RapiTime
- Métodos Estatísticos
- Métodos Estatísticos – Teoria dos Valores Extremos
- Métodos Estatísticos – TVE – Análise Experimental
- Considerações Gerais

