

DYCA: Multi-core Dynamically Adaptable Cache Bypassing Mechanism

Mariana Carmin

Orientador: Marco Antonio Zanata Alves



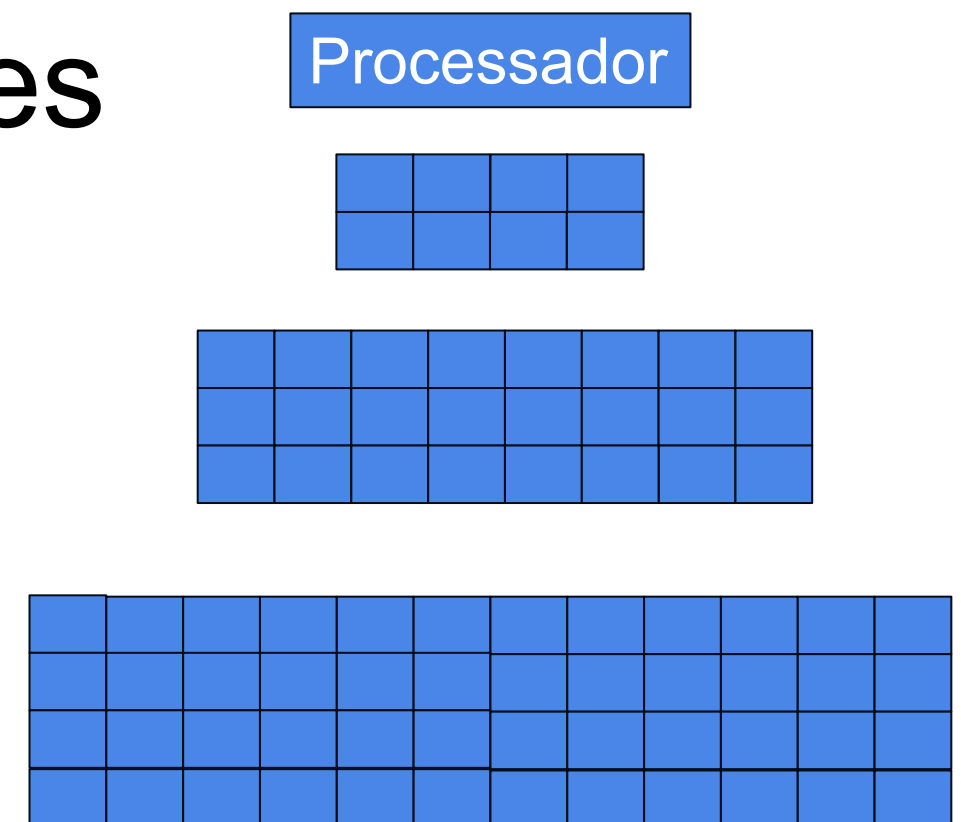
Por que pesquisar sobre
memórias cache?

Memórias *cache*

- Memórias menores e próximas do processador
 - Latência de acesso menor
- Usa o princípio de localidade temporal e espacial
 - Ilusão de uma memória maior e mais rápida

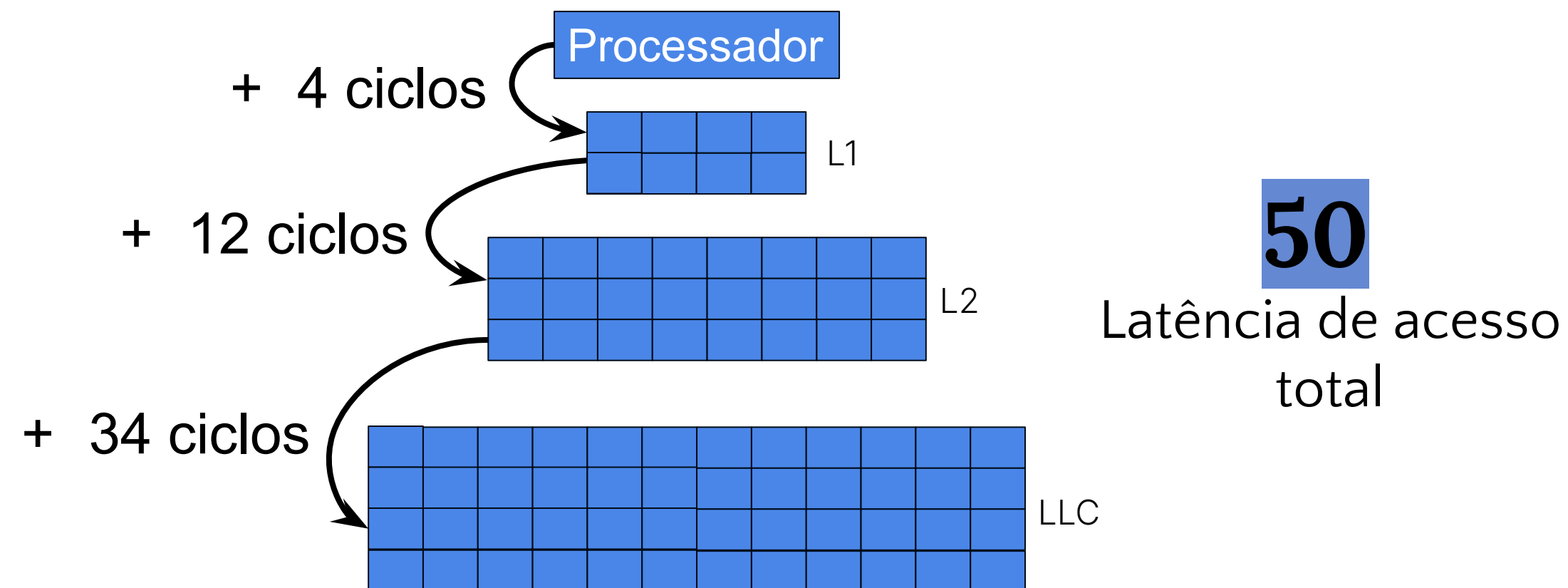
Funcionamento da *cache*

- Processadores modernos possuem *caches* de múltiplos níveis
- Diminui a latência média das requisições
- Acessos feitos de forma sequencial



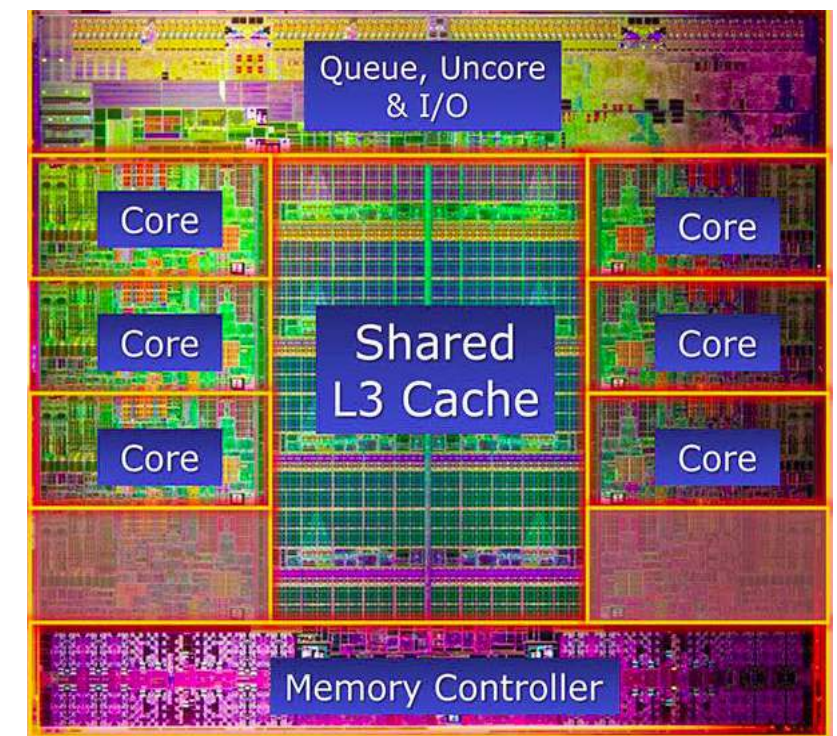
Funcionamento da *cache*

Em casos de falta de dados a requisição é acrescida da latência de acesso aos níveis de *cache*



Desafios da *cache*

- Ocupam até 60% da área dos *chips* modernos
- Responsáveis por até 23% do consumo energético
- Processadores modernos possuem cada vez mais núcleos
 - Maior pressão na *Last-Level cache* (LLC)
 - Poluição e conflitos na *cache*

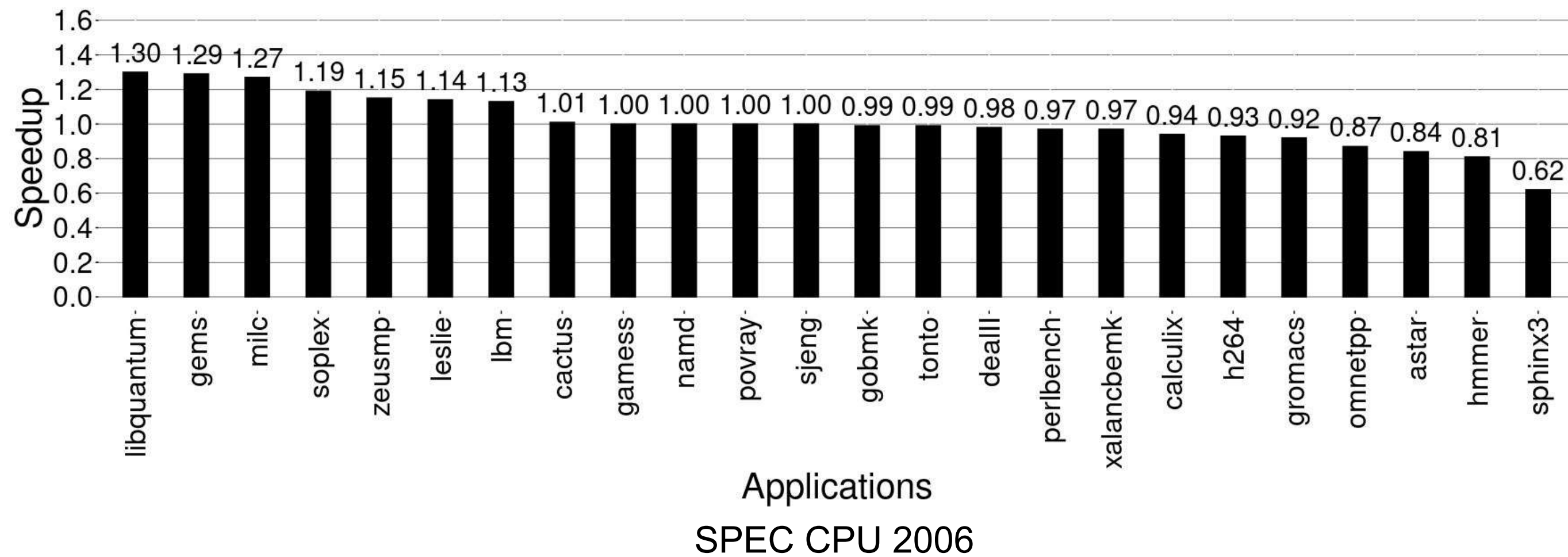


Intel i7 - 3960x

Desafios da *cache*

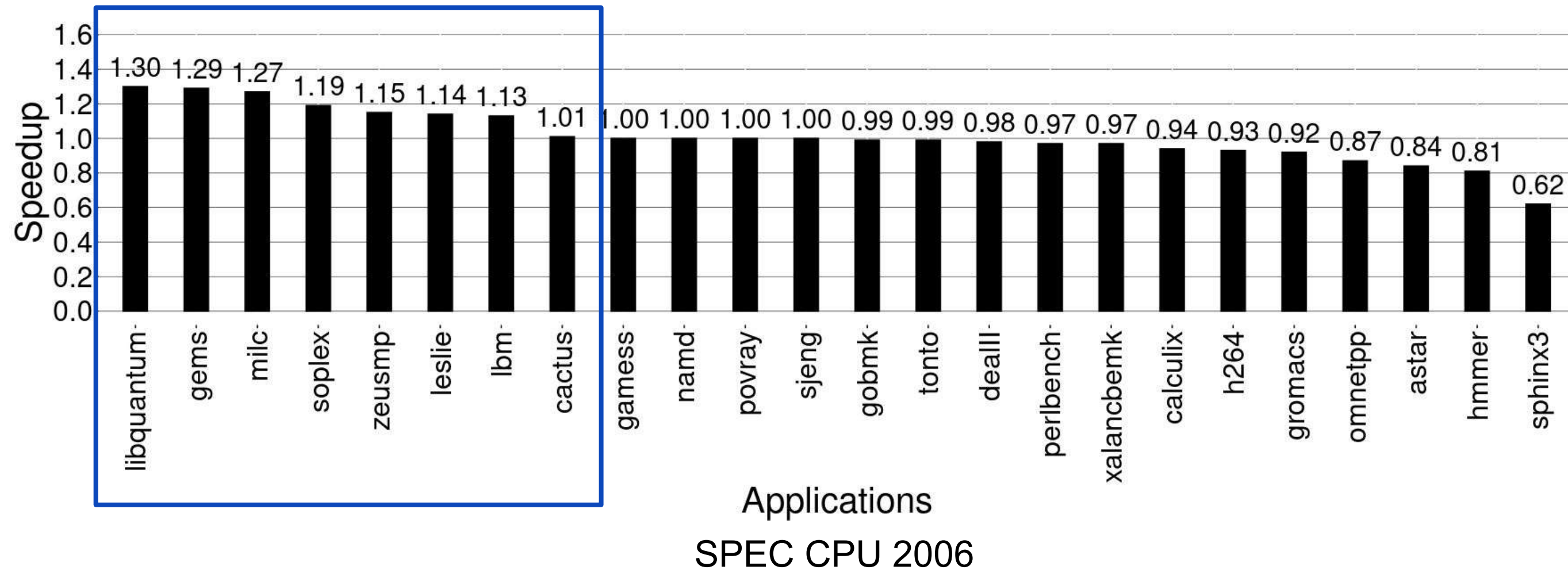
- Há aplicações que não apresentam localidade temporal ou espacial
 - Não se beneficiam do último nível de cache
 - Experimento:
 - Execução sem a LLC
 - Execução com LLC de 16MB

Resultados SPEC CPU 2006

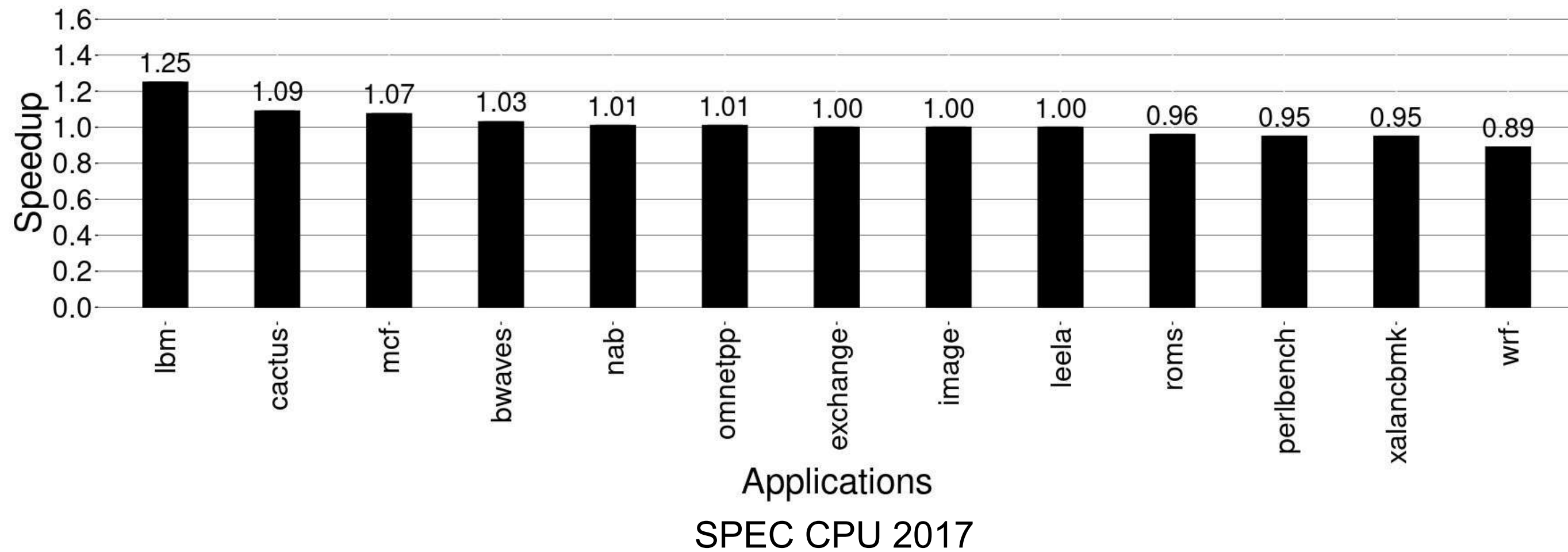


Resultados SPEC CPU 2006

Há aplicações que não se beneficiam da LLC

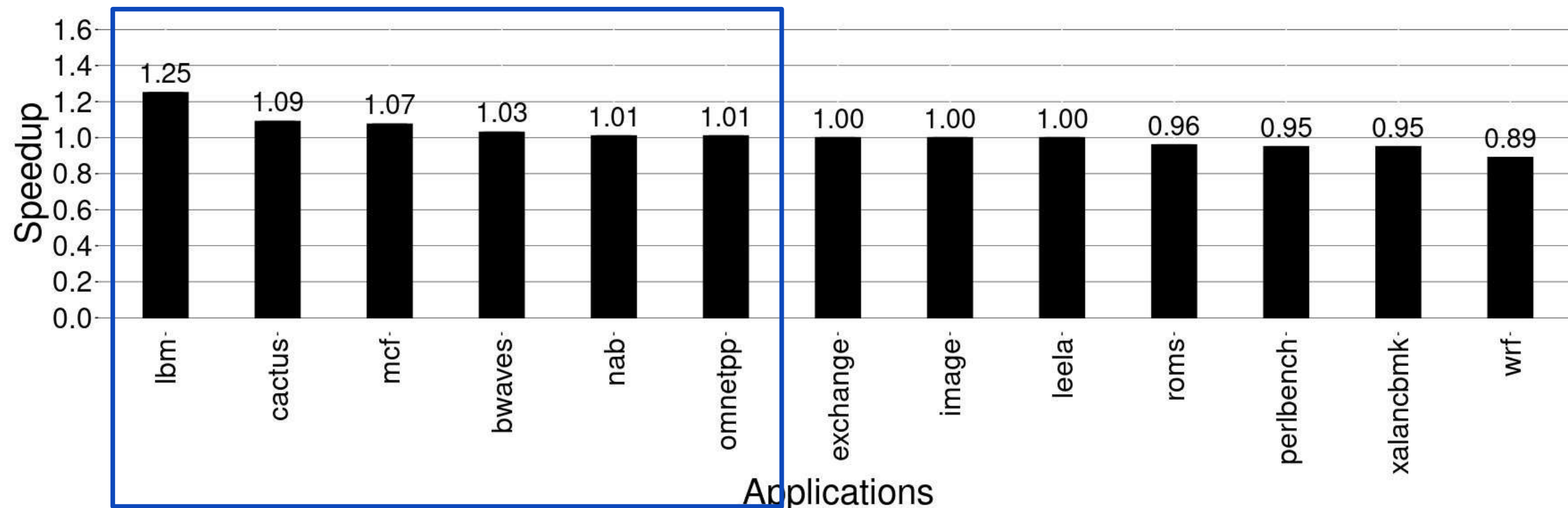


Resultados SPEC CPU 2017



Resultados SPEC CPU 2017

Há aplicações que não se beneficiam da LLC



SPEC CPU 2017

Memórias *cache* nem sempre
são sinônimo de desempenho.

Ideias

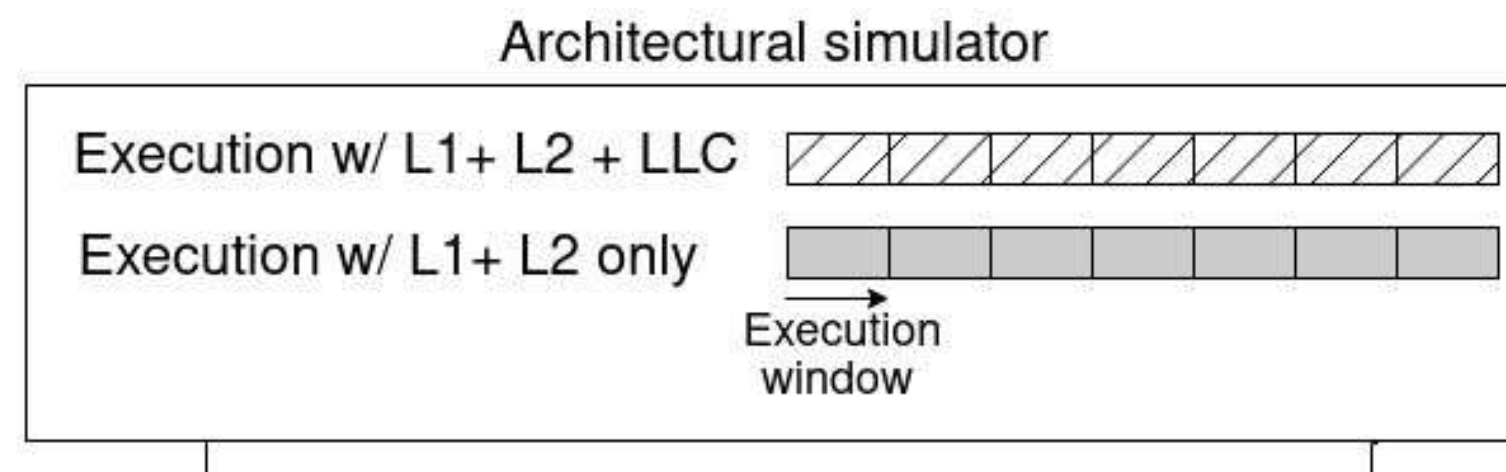
- Heterogeneidade já presente em processadores
 - ARM big.little
- Trabalhos correlatos que promovem desligamento de níveis da cache
 - Liu et. al. (2020)
 - Park, Kim e Hou. (2021)
 - Zhu e Zend (2021)

É possível melhorar o tempo de execução dos sistemas computacionais usando uma LLC dinâmica e adaptativa?

DYCA

- *Multi-core Dynamically Adaptable Cache Bypassing Mechanism:*
 - Mecanismo dinâmico
 - Consciente de aplicação
 - Utiliza bypass para adaptar o uso da LLC
- Modelo de decisão baseado na predição do IPC

DYCA - Visão geral

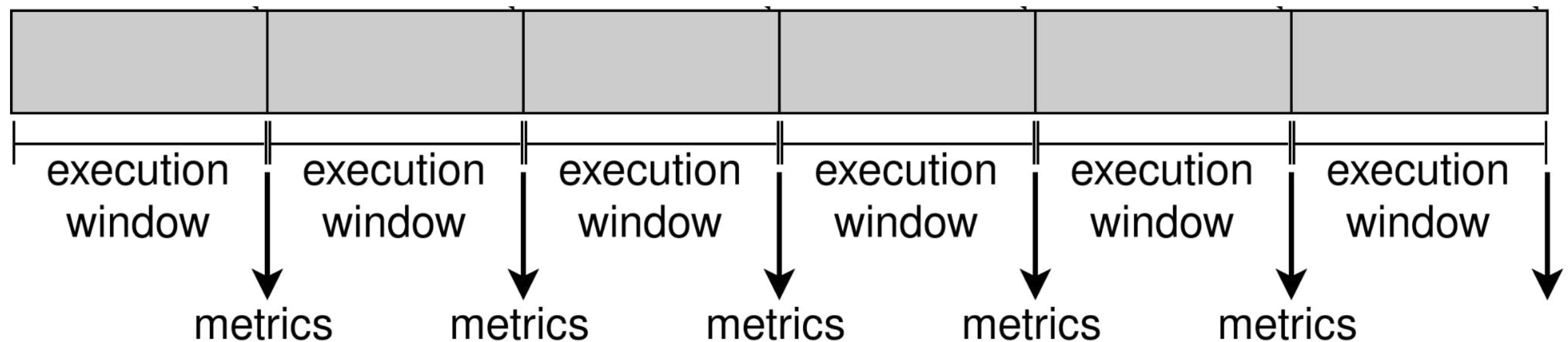


DYCA - Janelas de execução

- Janela de execução - período de execução onde a aplicação é selecionada pelo sistema operacional para ser executada
 - Objetivo: identificar as diferentes fases de um programa

DYCA - Janelas de execução

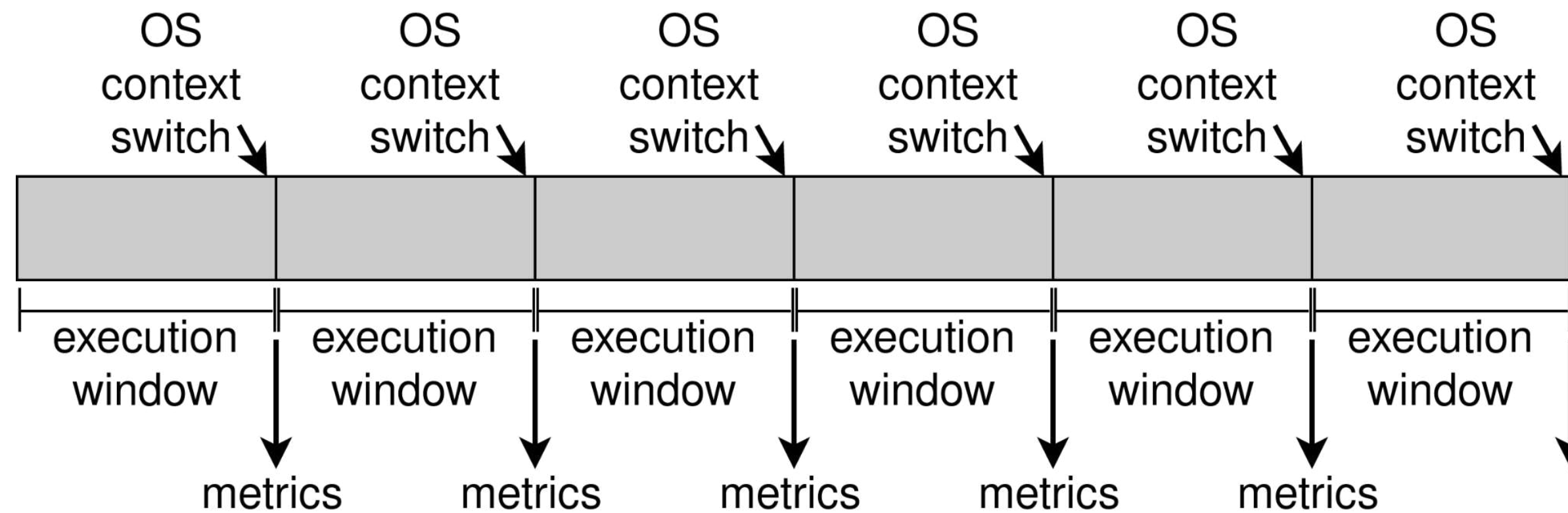
- Medimos os contadores de *hardware* ao fim de cada janela



DYCA - Janelas de execução

- 200 milhões de ciclos:
 - Média de tempo em que a troca de contexto ocorre.
 - Mitigamos a latência de adaptar o uso da LLC
 - Mitigamos o efeito de cache fria

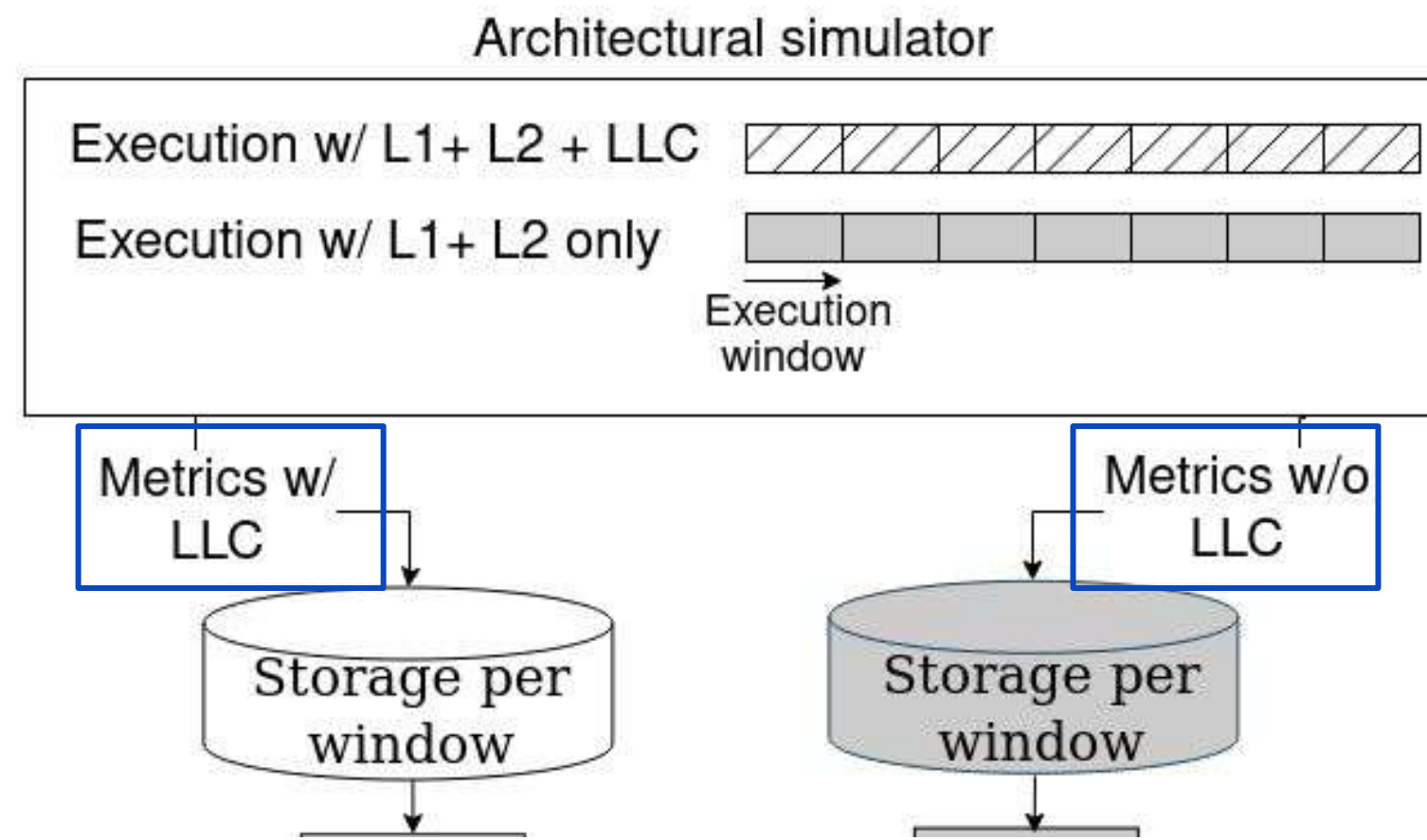
DYCA - Janelas de execução



DYCA - Janelas de execução

- Granularidade mais fina aumentaria o *overhead* de desempenho
- Uma fase pode se estender por milhões de ciclos

DYCA



| Component | Hardware counters | Execution window | Full execution |
|------------------|--|-------------------------|-----------------------|
| L1, L2, and LLC | Number of misses | ● | ● |
| | Number of hits | ● | ● |
| | Number of accesses | ● | ● |
| | Number of writes | ● | ● |
| | Miss rate | ● | ● |
| | Hit rate | ● | ● |
| | Misses per Kilo Instructions (MPKI) | ● | ● |
| | Hits per Kilo Instructions (HPKI) | ● | ● |
| Processing Core | Instructions per Cycle (IPC) using LLC | ● | ● |
| | IPC without LLC | ● | ● |
| | Number of instructions | | ● |
| Memory system | Load | ● | ● |
| | Store | ● | ● |

DYCA - *Features*

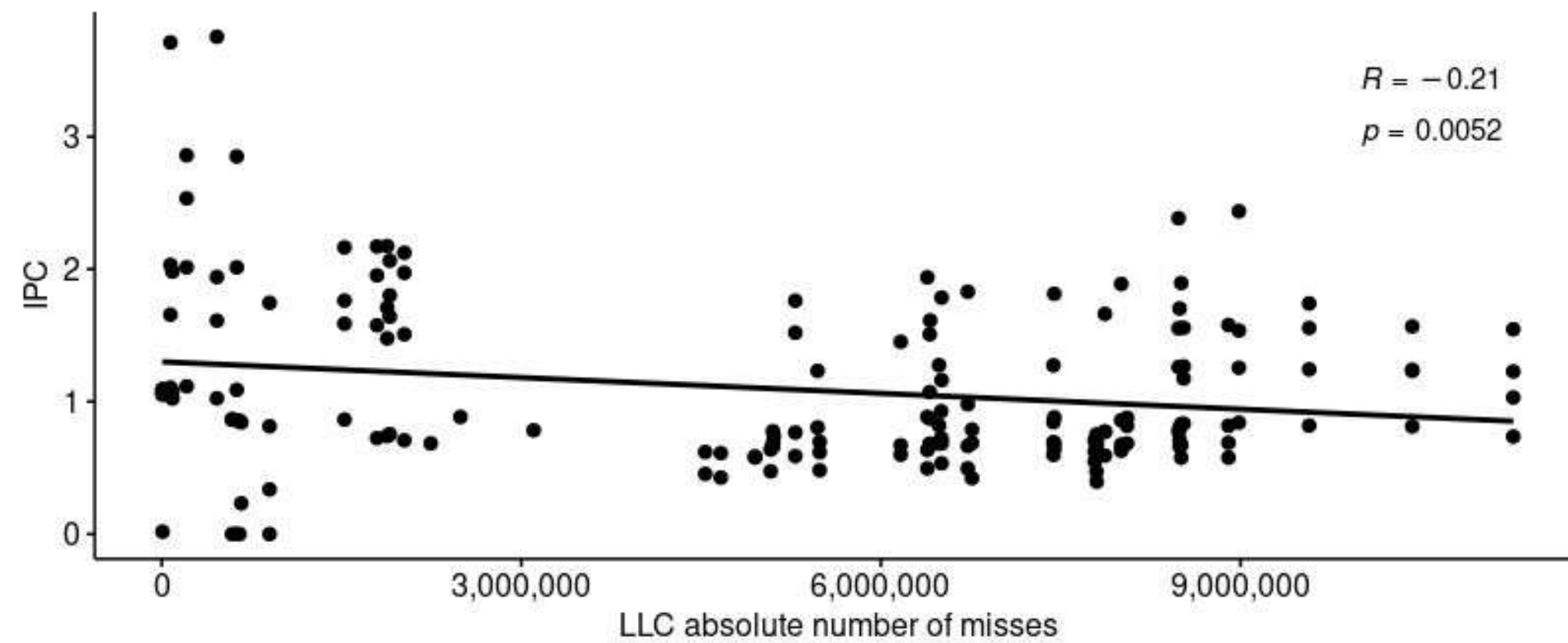
- Coletadas em cada janela de execução
- Objetivo:
 - Avaliação justa da influência dos contadores de *hardware*
 - Correlação com IPC
 - Redução do espaço de busca para construção do modelo

DYCA - *Features*

- Pearson
 - R: mede a força e direção da correlação $[-1, 1]$
 - p-value: potencial de observar um R específico considerando a hipótese nula
 - < 0.05

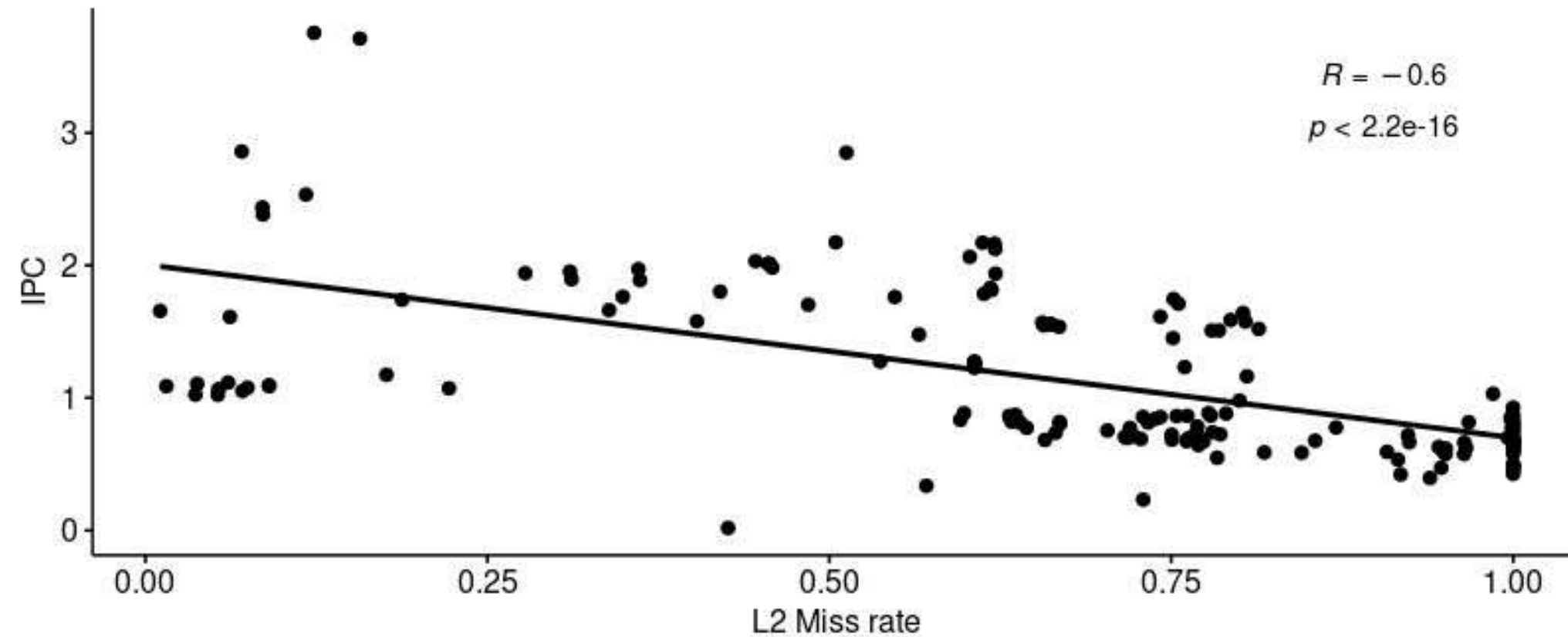
Valores absolutos

- Valor absoluto de *misses* na LLC

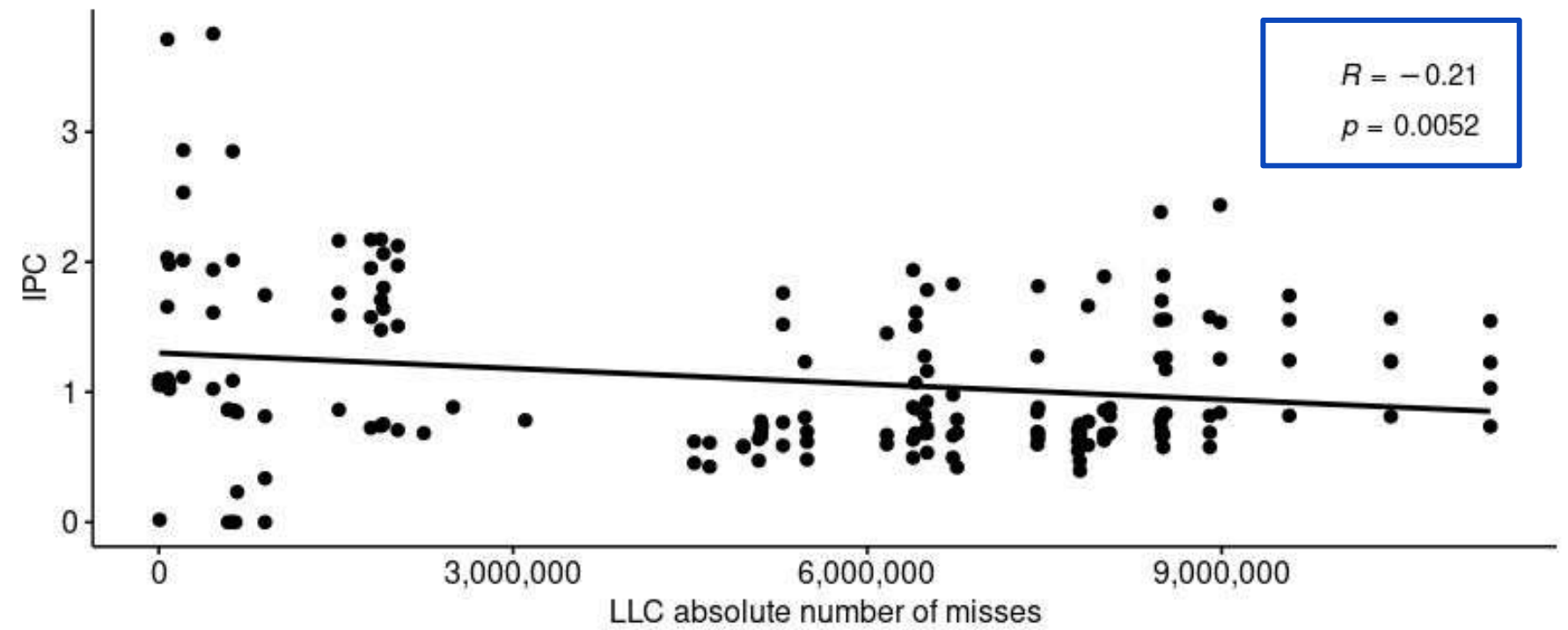
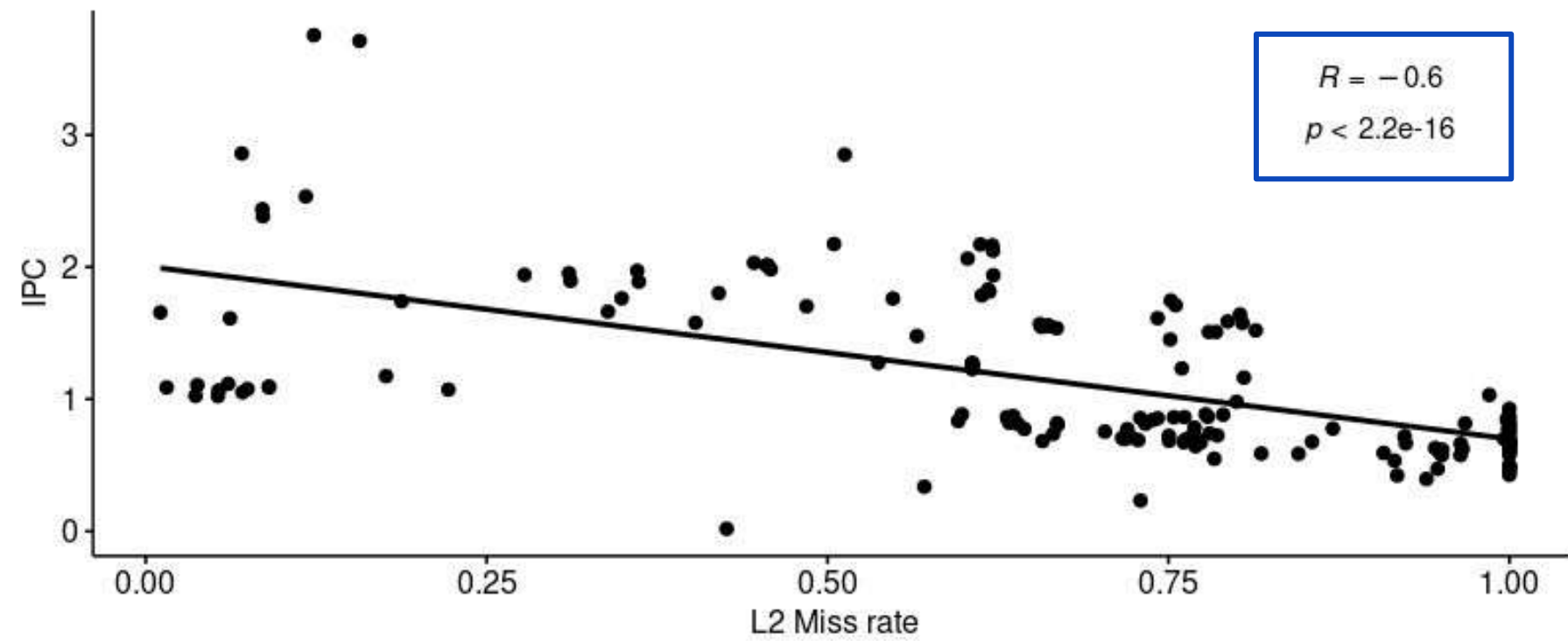


Valores comumente usados

- Taxa de *miss* na L2



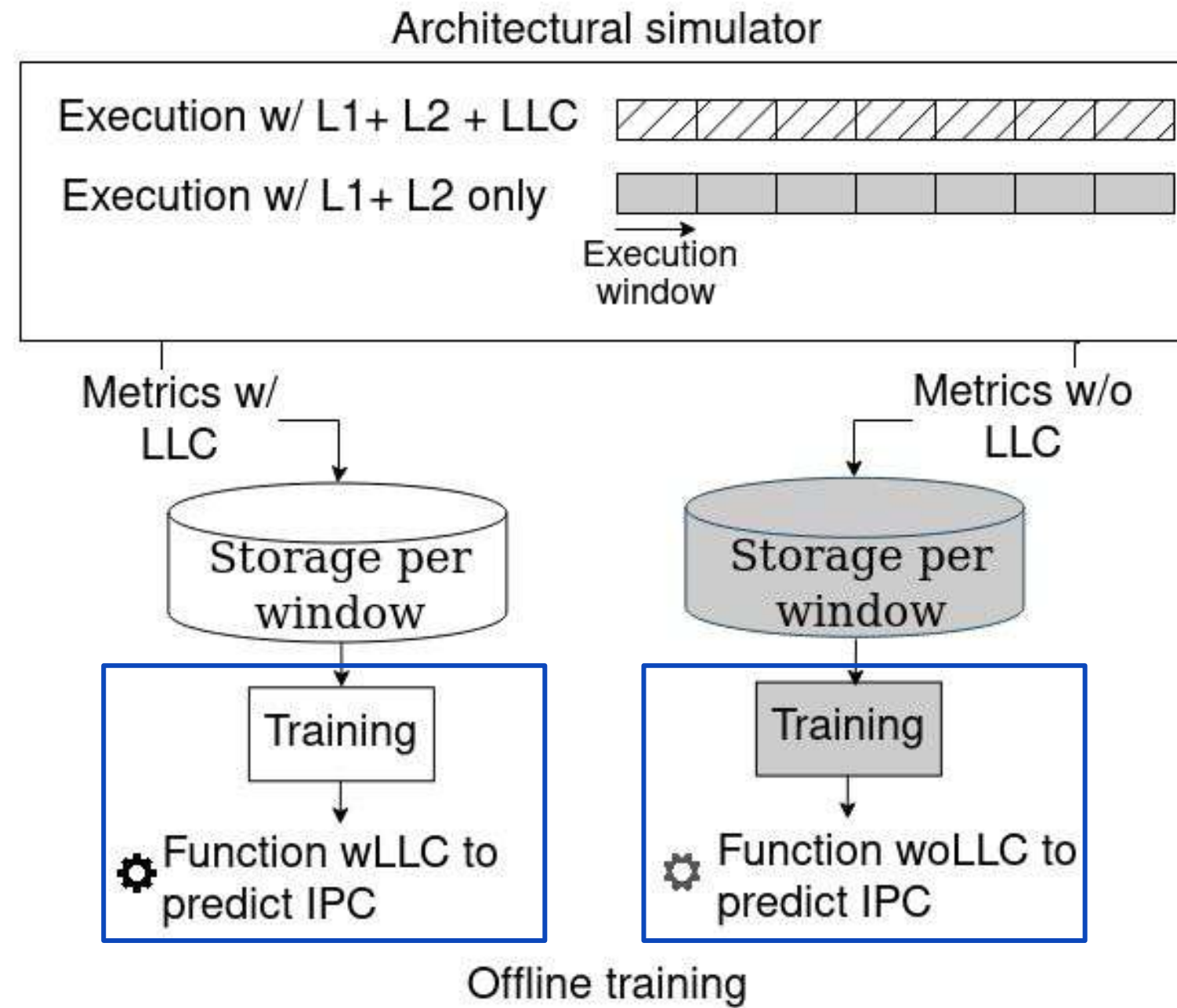
Comparação de *features*



DYCA - *Features*

- Problemas encontrados:
 - *Features* com uma correlação baixa
 - Correlação não linear
- Soluções:
 - Uso do modelo GAM
 - Combinação de métricas

DYCA

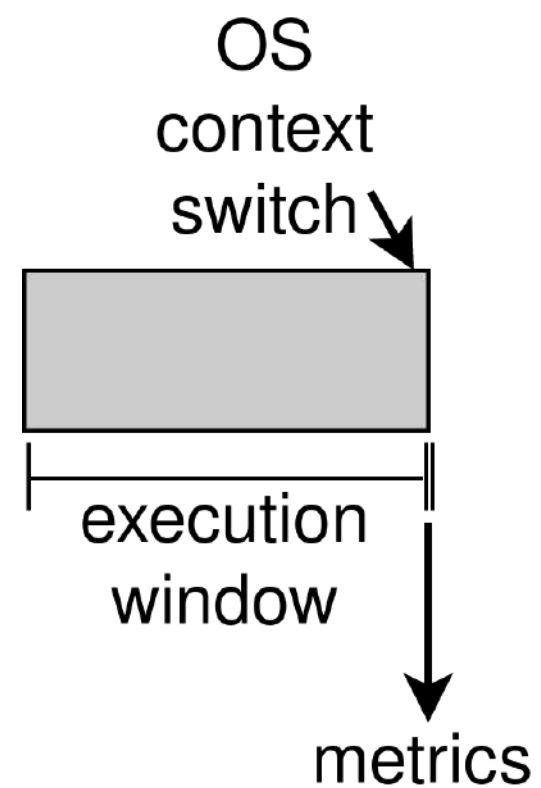


DYCA - Modelo de regressão

- Uso dos contadores de hardware para prever o IPC
 - Predição para diferentes configurações
 - Usa informação da janela de execução anterior para prever o comportamento da próxima

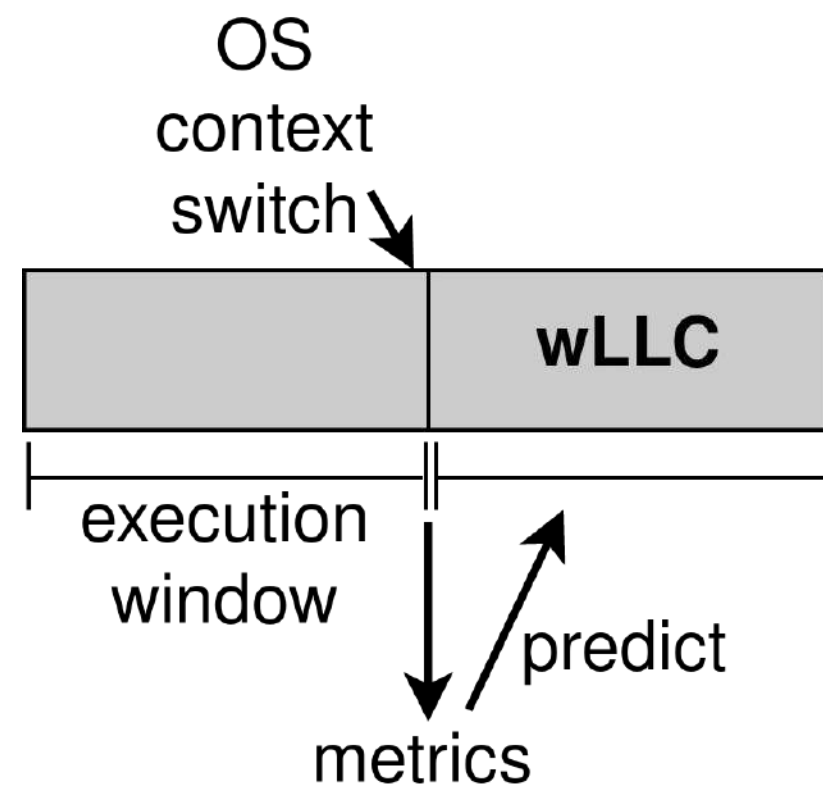
DYCA - Modelo de regressão

program execution



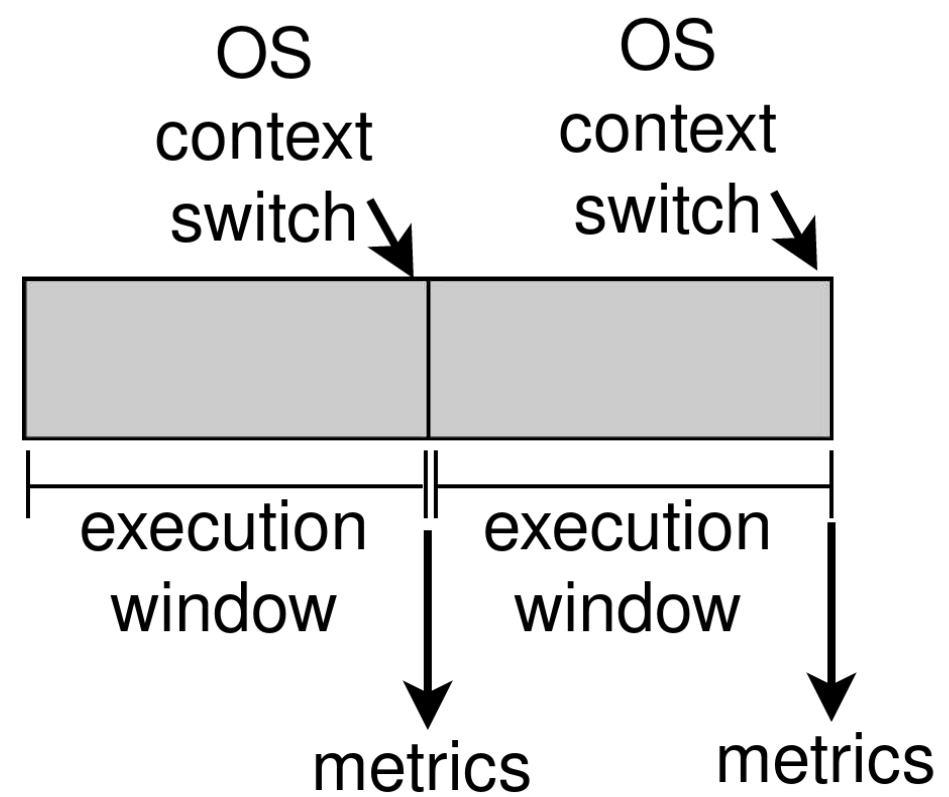
DYCA - Modelo de regressão

program execution

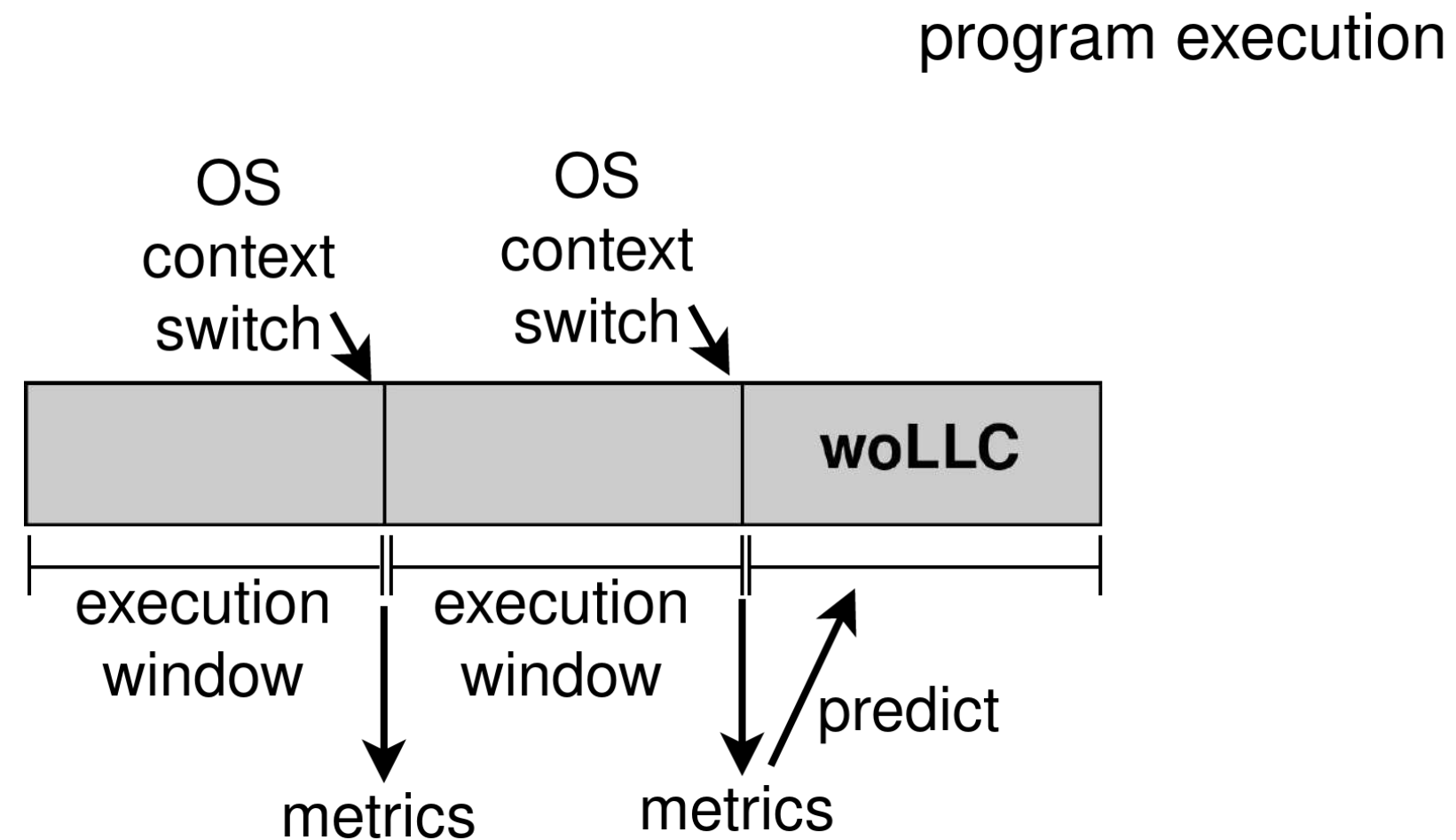


DYCA - Modelo de regressão

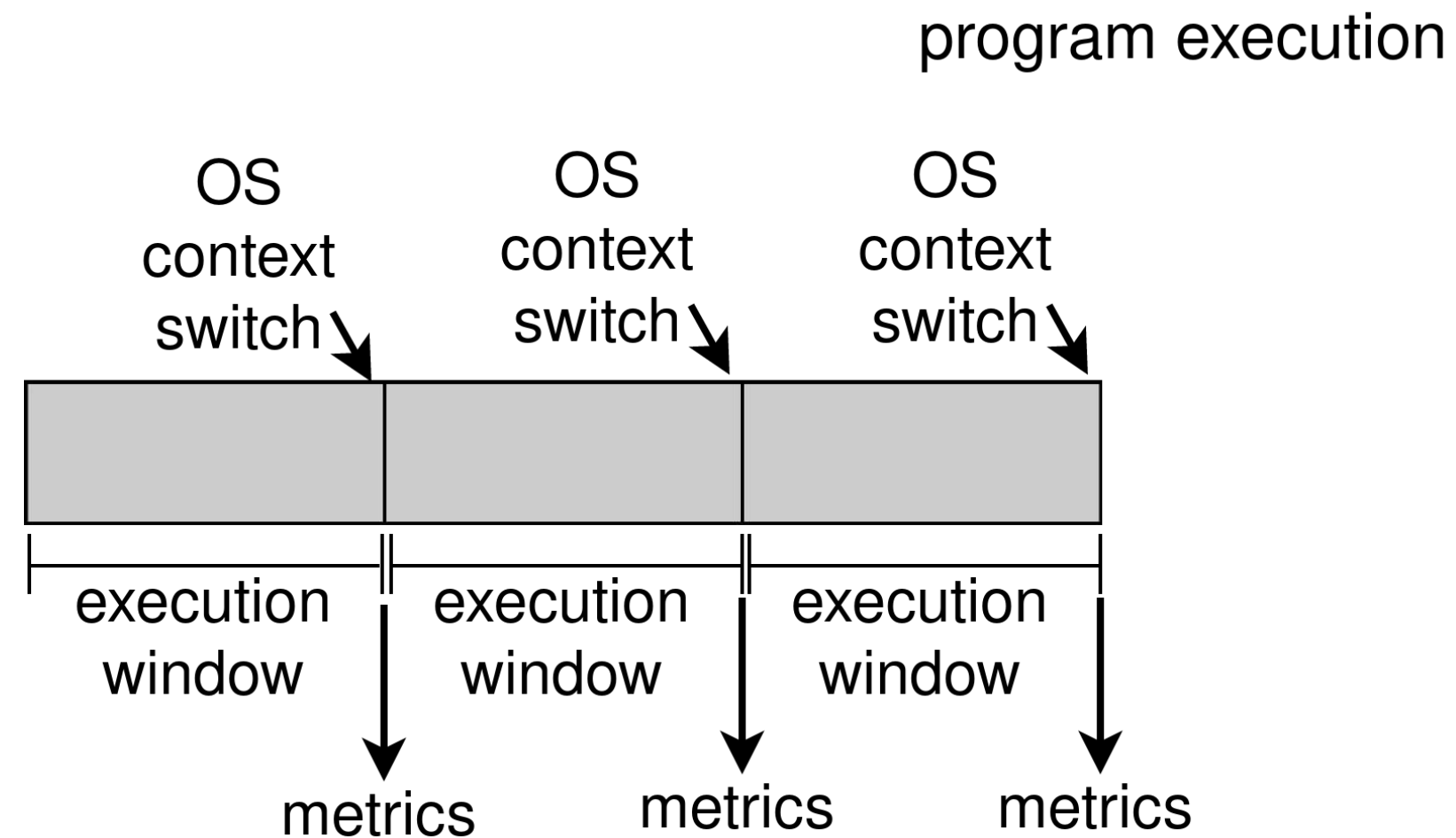
program execution



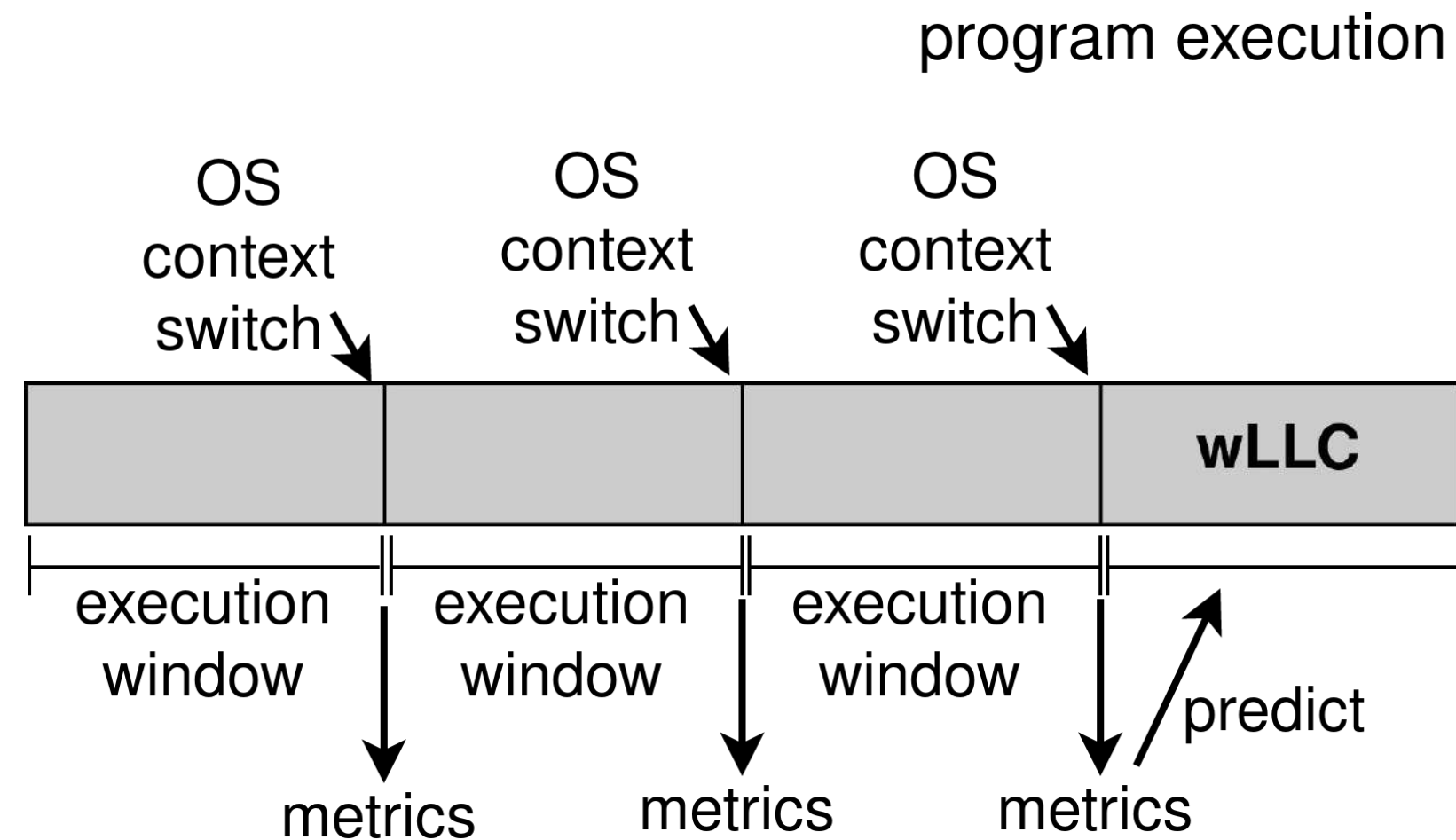
DYCA - Modelo de regressão



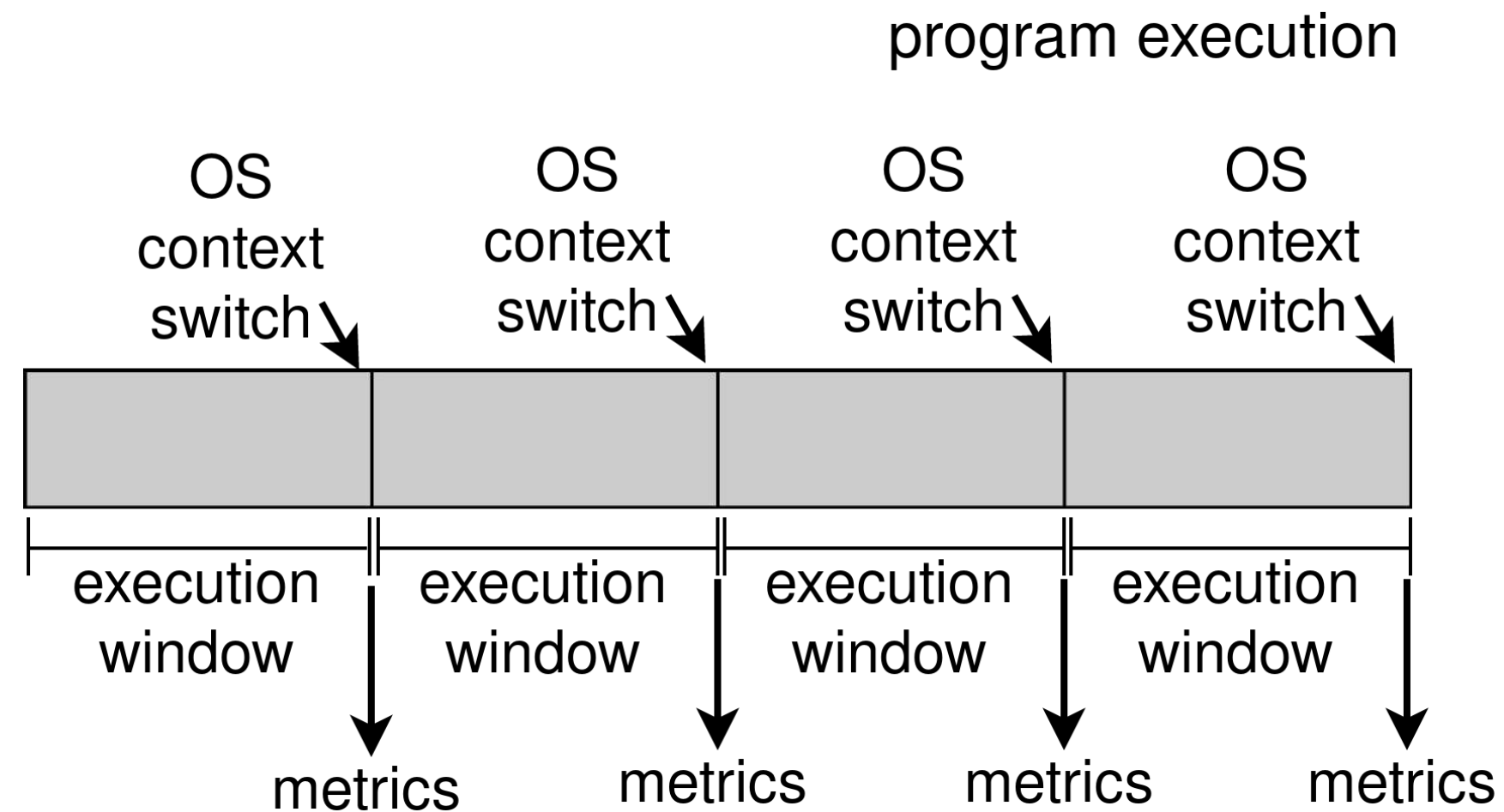
DYCA - Modelo de regressão



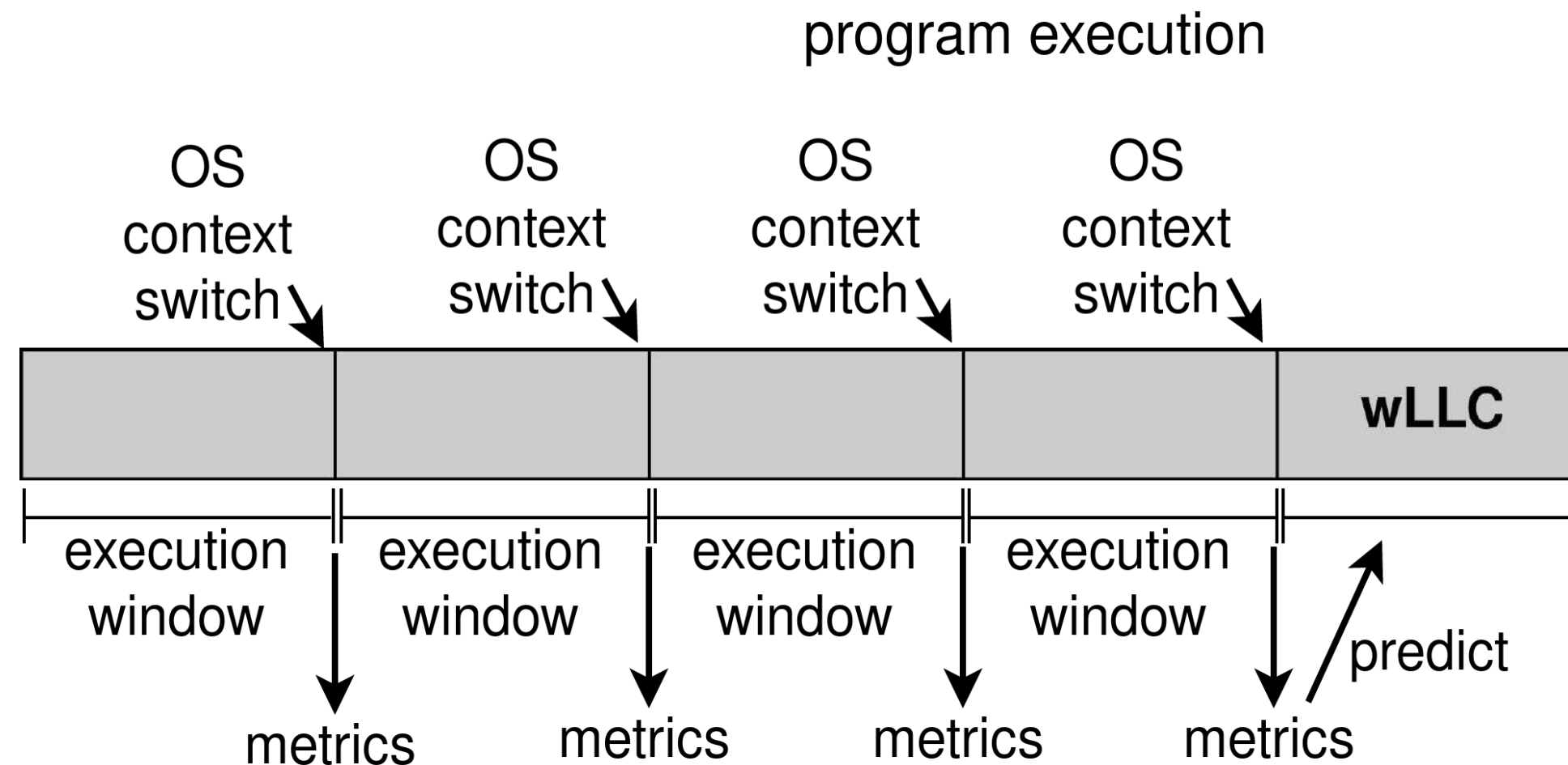
DYCA - Modelo de regressão



DYCA - Modelo de regressão



DYCA - Modelo de regressão



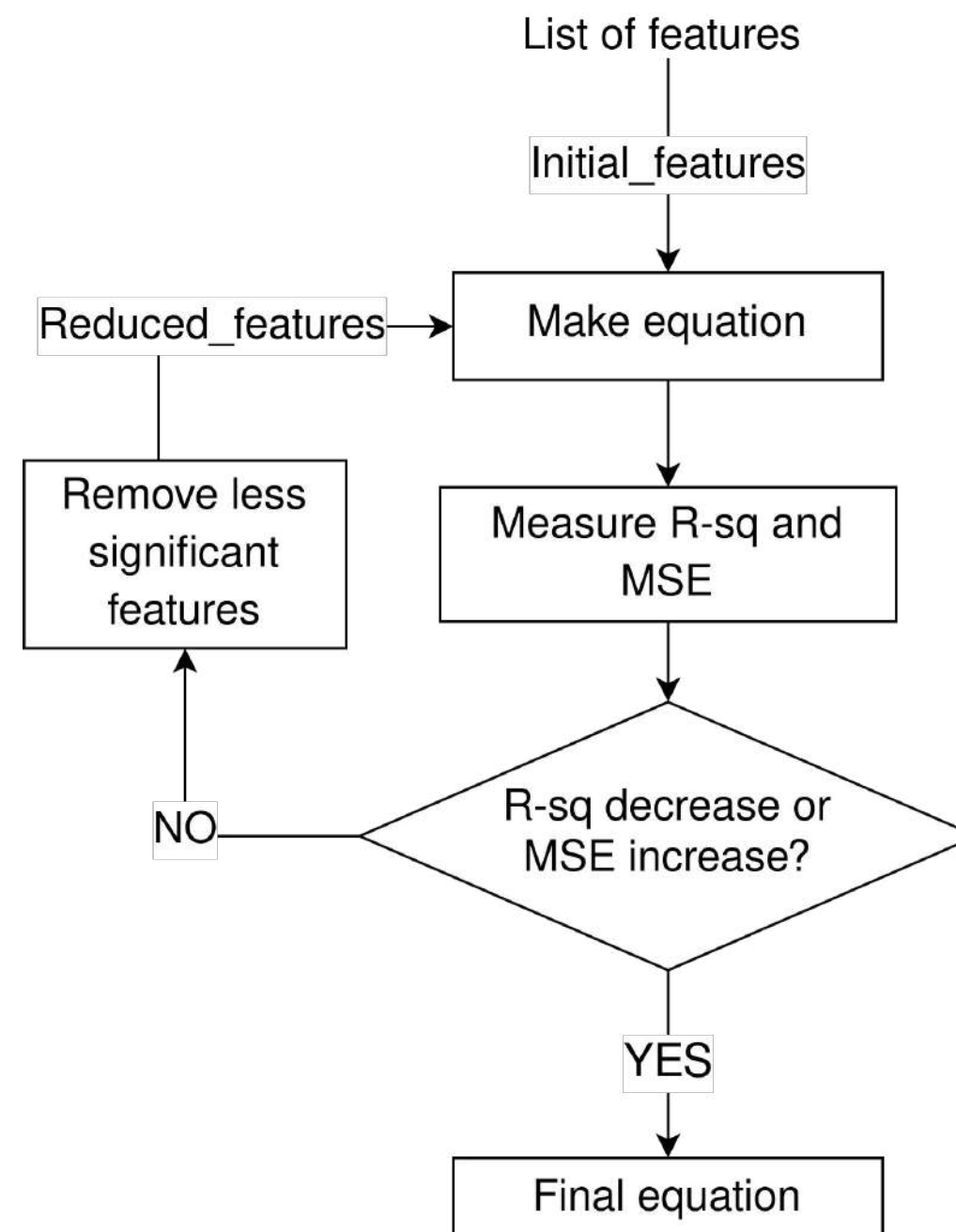
DYCA - Base de dados

- Treino
 - SPEC-CPU 2006
 - 24 aplicações
- Teste
 - SPEC-CPU 2017
 - 17 aplicações

DYCA - *Features*

- L1, L2 e LLC - Taxa de *miss*, taxa de *hit*, HPKI, MPKI, etc.
- Núcleo - IPC wLLC e IPC woLLC.
- Sistema de memória - Número de *loads* e *stores*

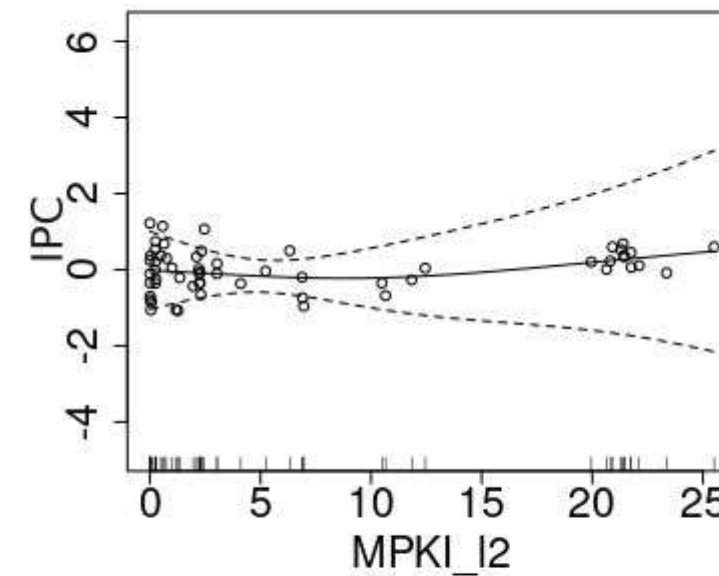
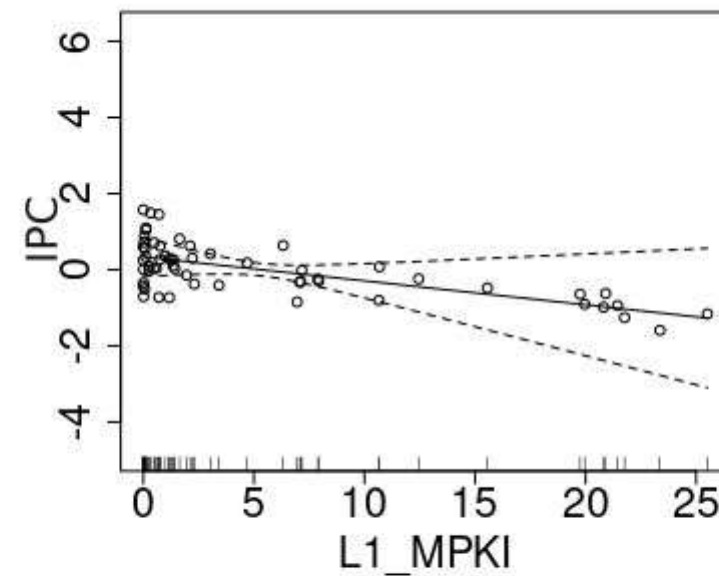
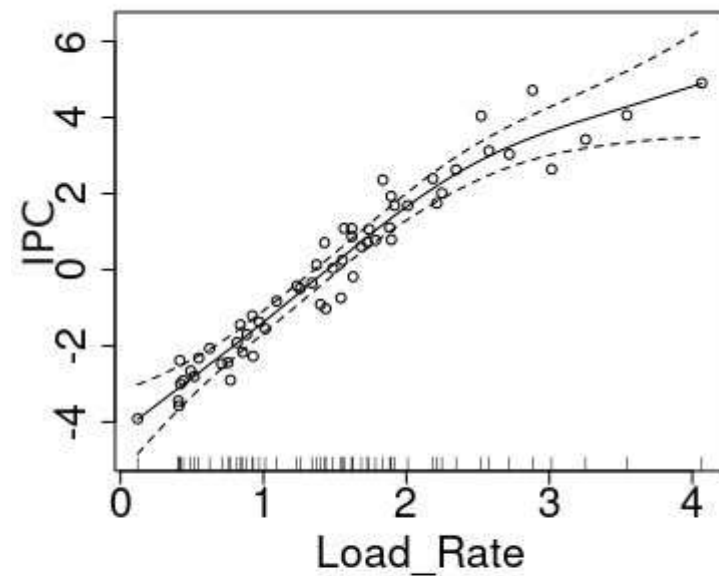
DYCA - Modelo de regressão



DYCA - Modelo de regressão

- Modelo de regressão - wLLC

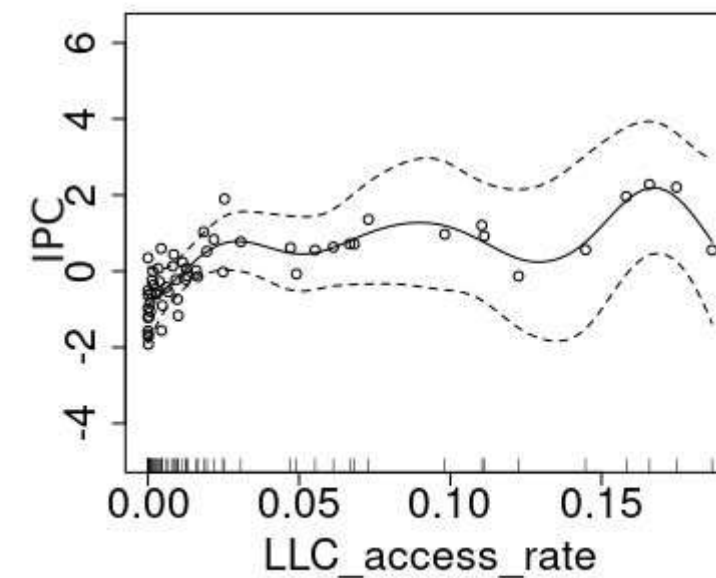
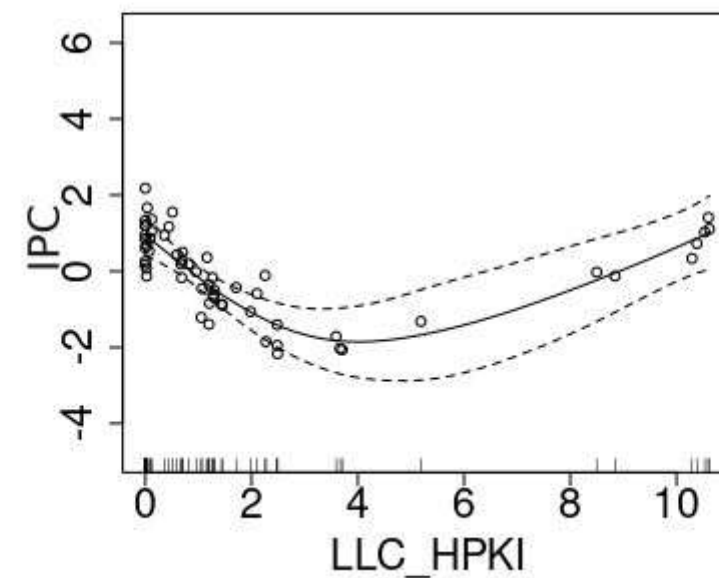
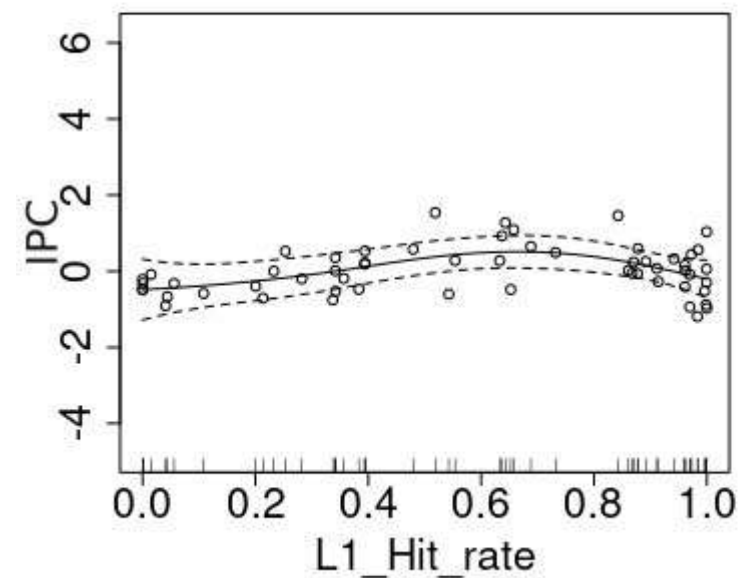
$$IPC = s_0L1\ MPKI + s_1L1\ hit\ ratio + s_2L2\ MPKI + s_4LLC\ HPKI + s_5LLC\ accesses + s_6Loads$$



DYCA - Modelo de regressão

- Modelo de regressão - wLLC

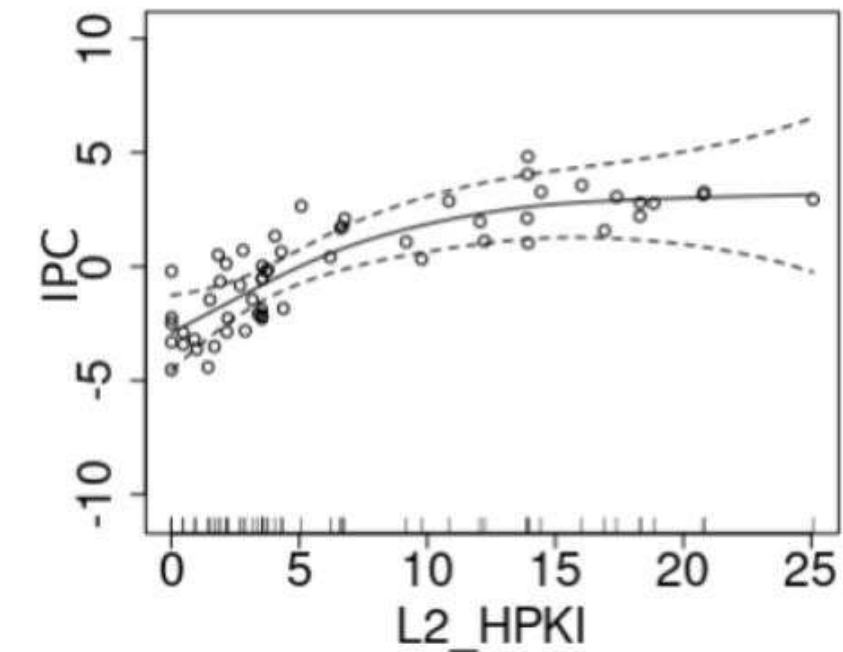
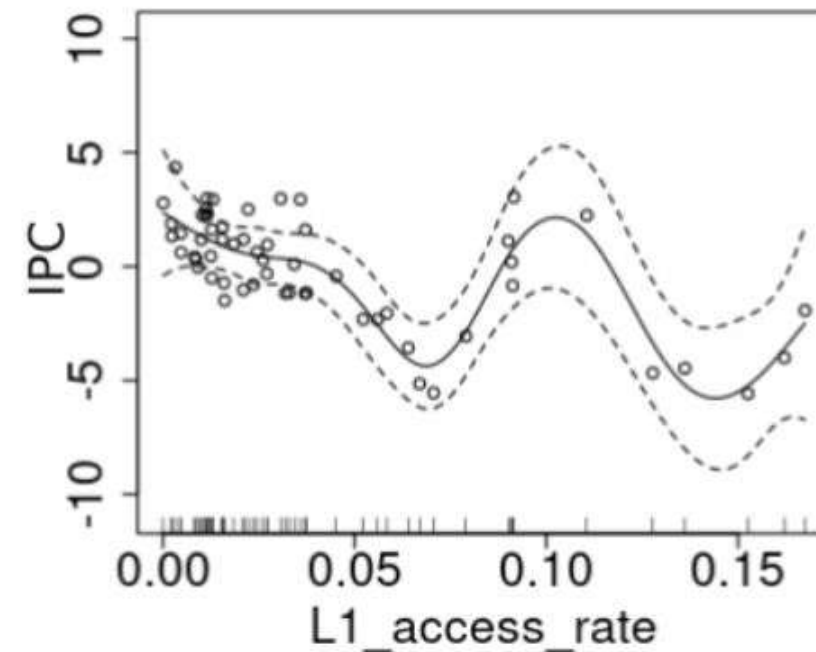
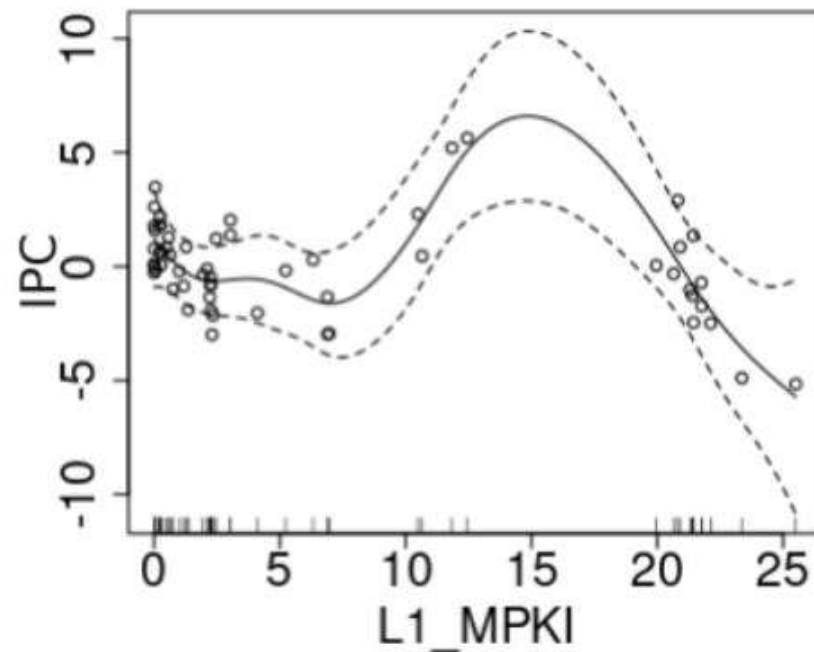
$$IPC = s_0 L1 MPKI + s_1 L1 hit ratio + s_2 L2 MPKI + s_4 LLC HPKI + s_5 LLC accesses + s_6 Loads$$



DYCA - Modelo de regressão

- Modelo de regressão - woLLC

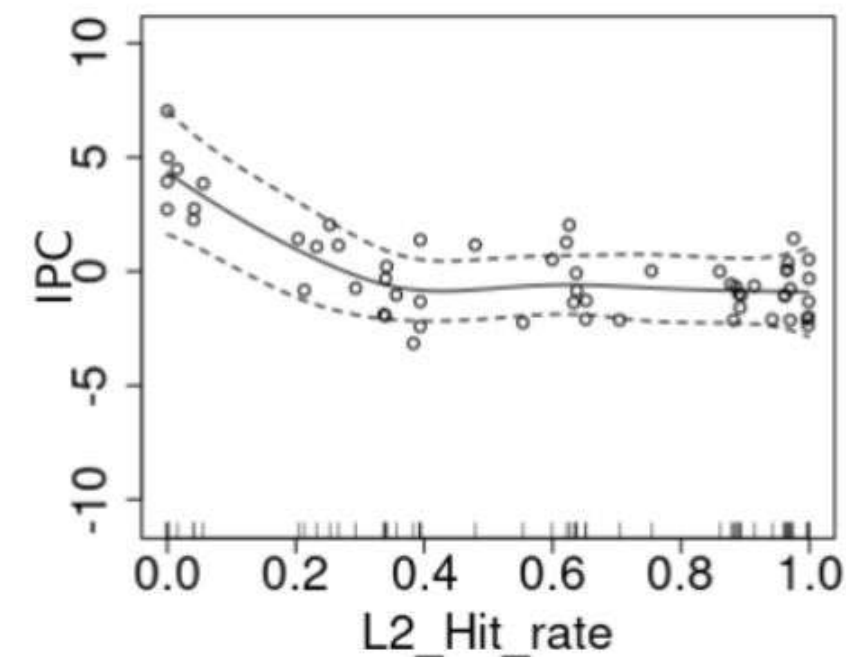
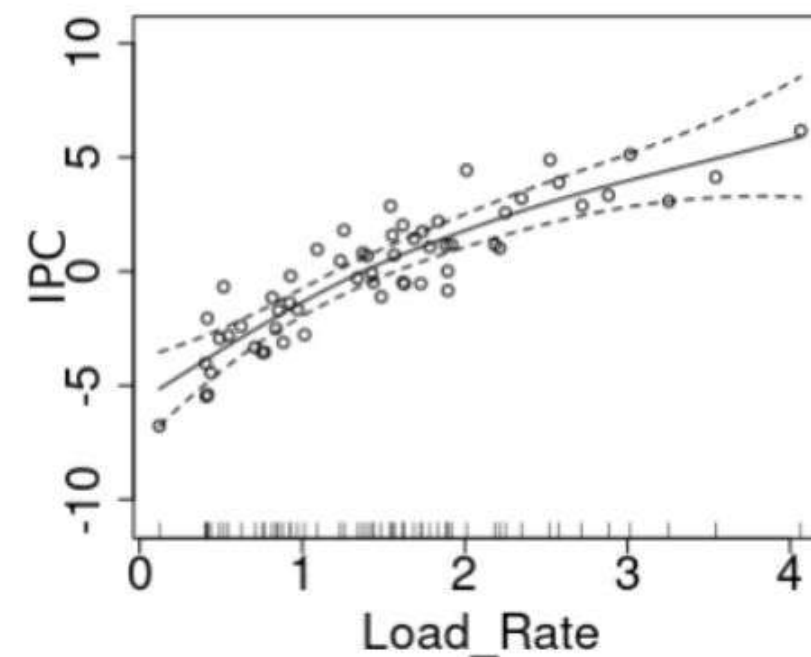
$$IPC = s_0 L1\ MPKI + s_1 L1\ acesses + s_2 L2\ HPKI + s_3 L2\ Hitrate + s_4 Loads$$



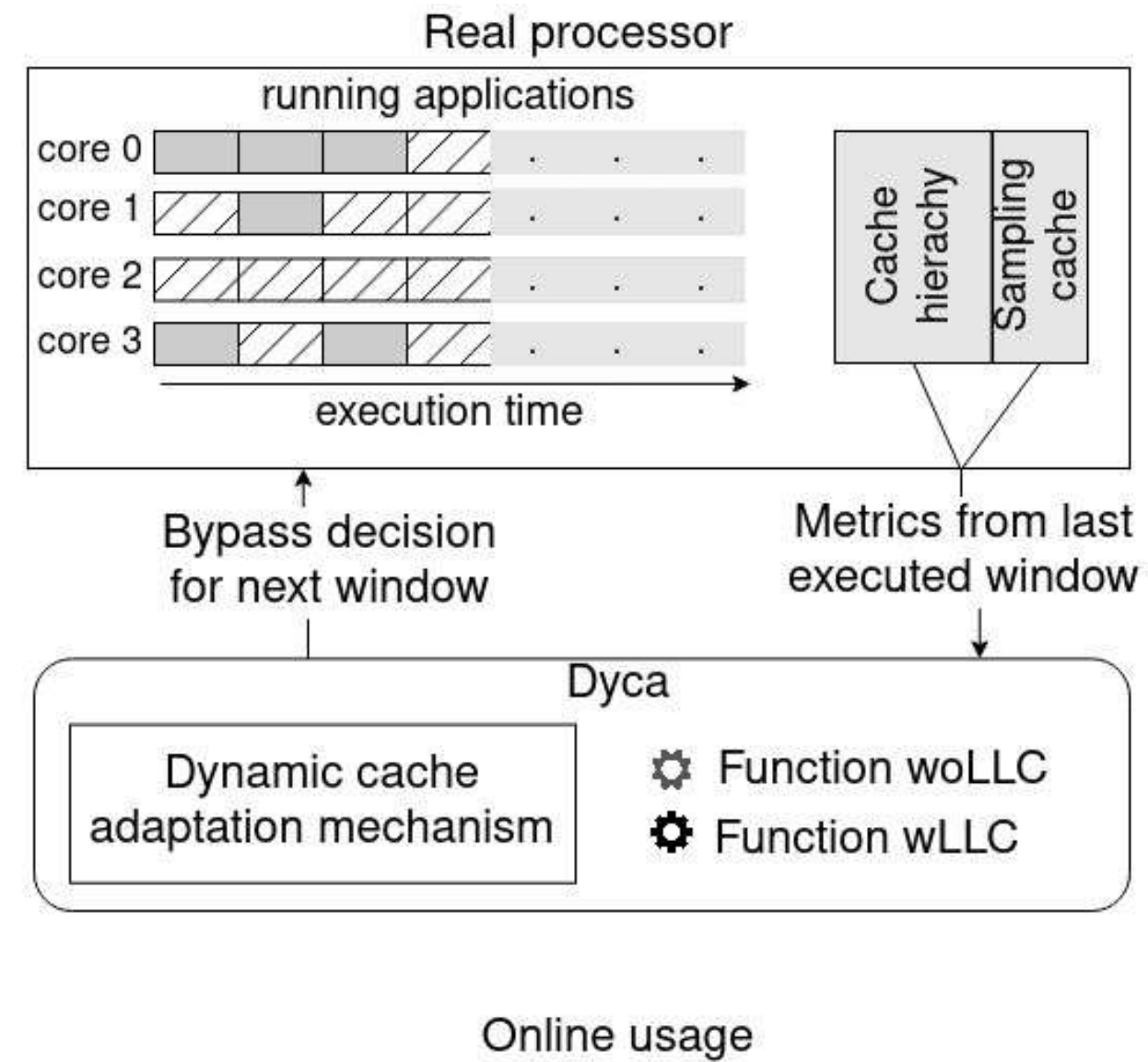
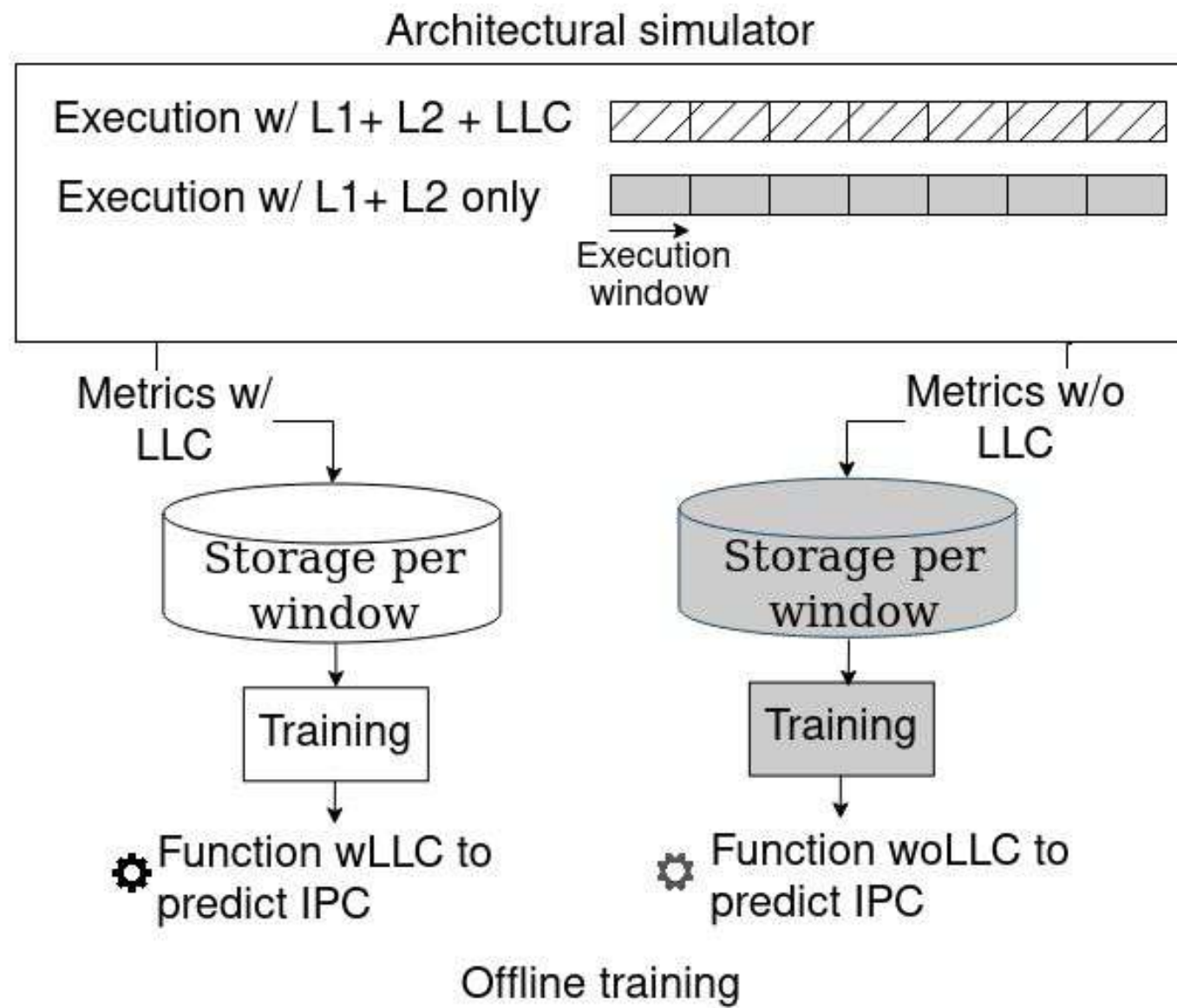
DYCA - Modelo de regressão

- Modelo de regressão - woLLC

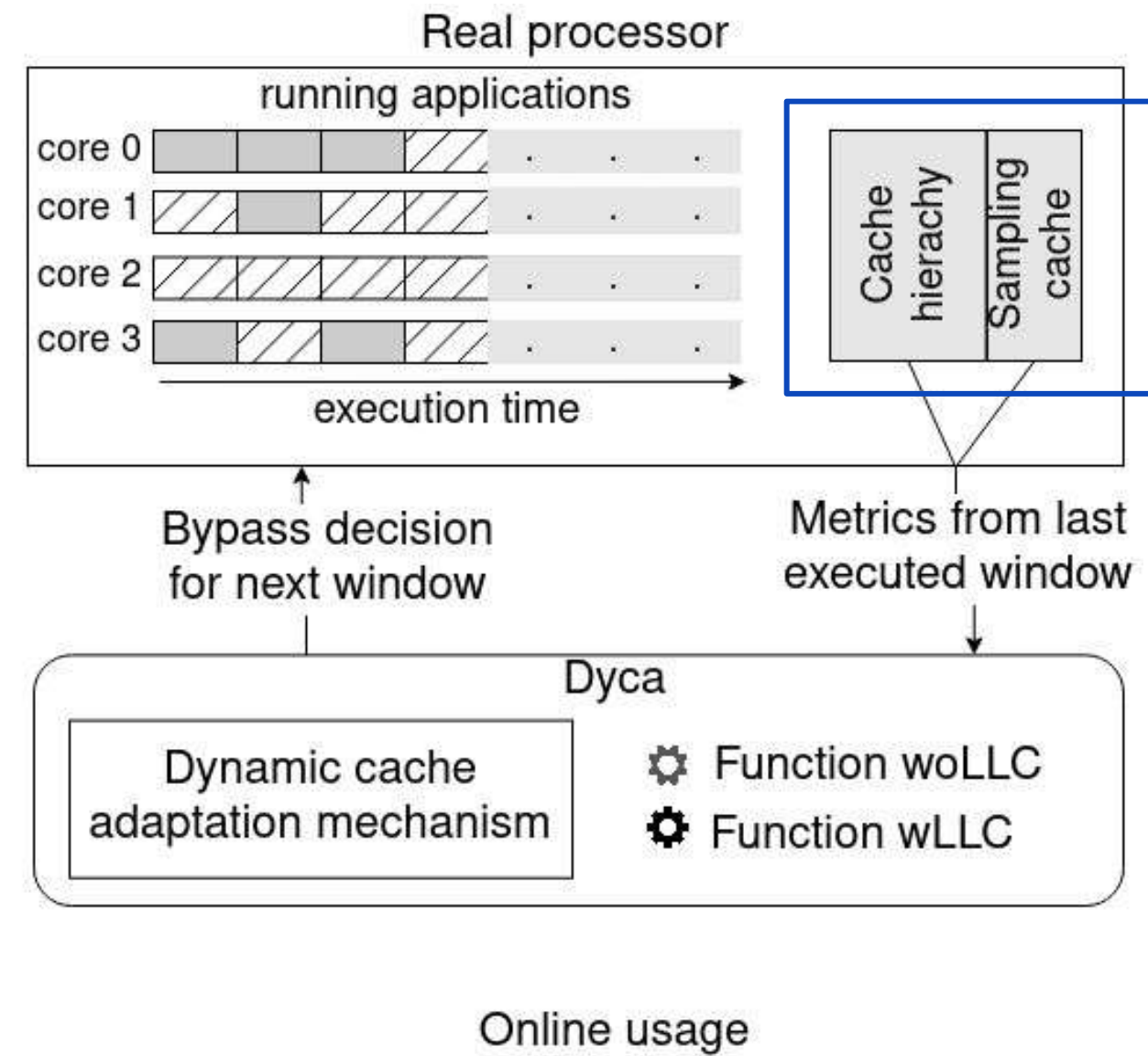
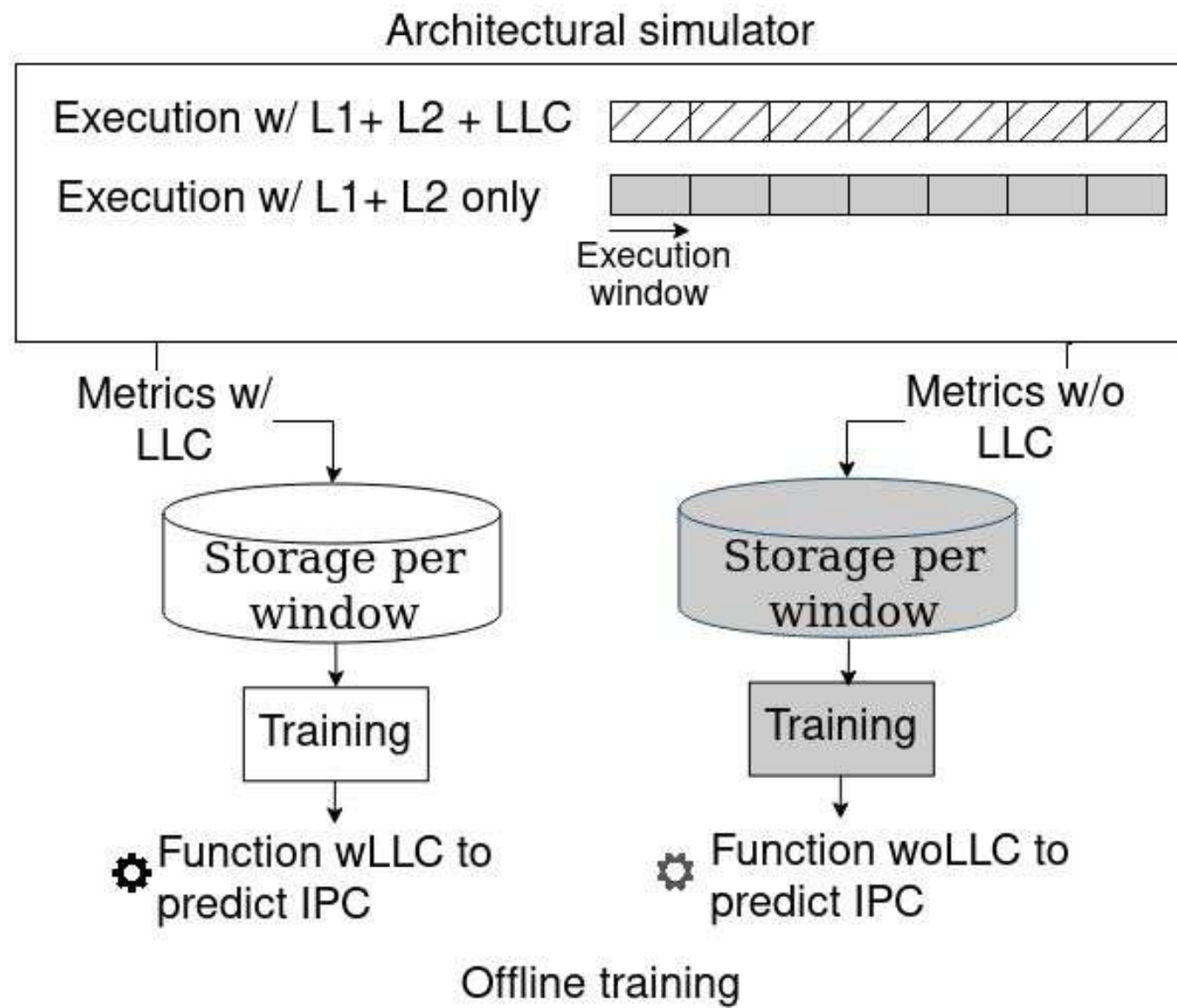
$$IPC = s_0 L1 MPKI + s_1 L1 acesses + s_2 L2 HPKI + s_3 L2 Hitrate + s_4 Loads$$



DYCA



DYCA

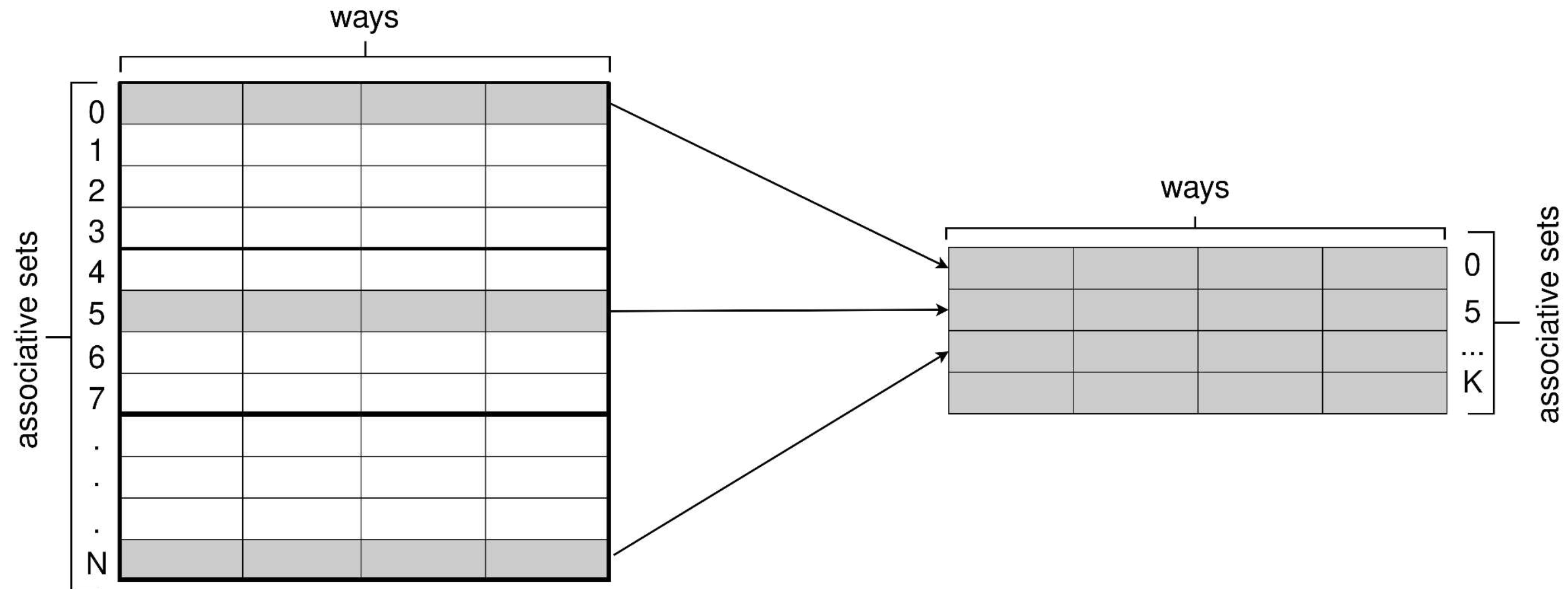


DYCA - *Sampling cache*

- Simular os contadores de *hardware* mesmo quando a LLC não está sendo usada
- Simula o comportamento da LLC em um pequeno número de conjuntos associativos¹
- Armazena apenas a TAG
- Baixo *overhead* de *hardware*

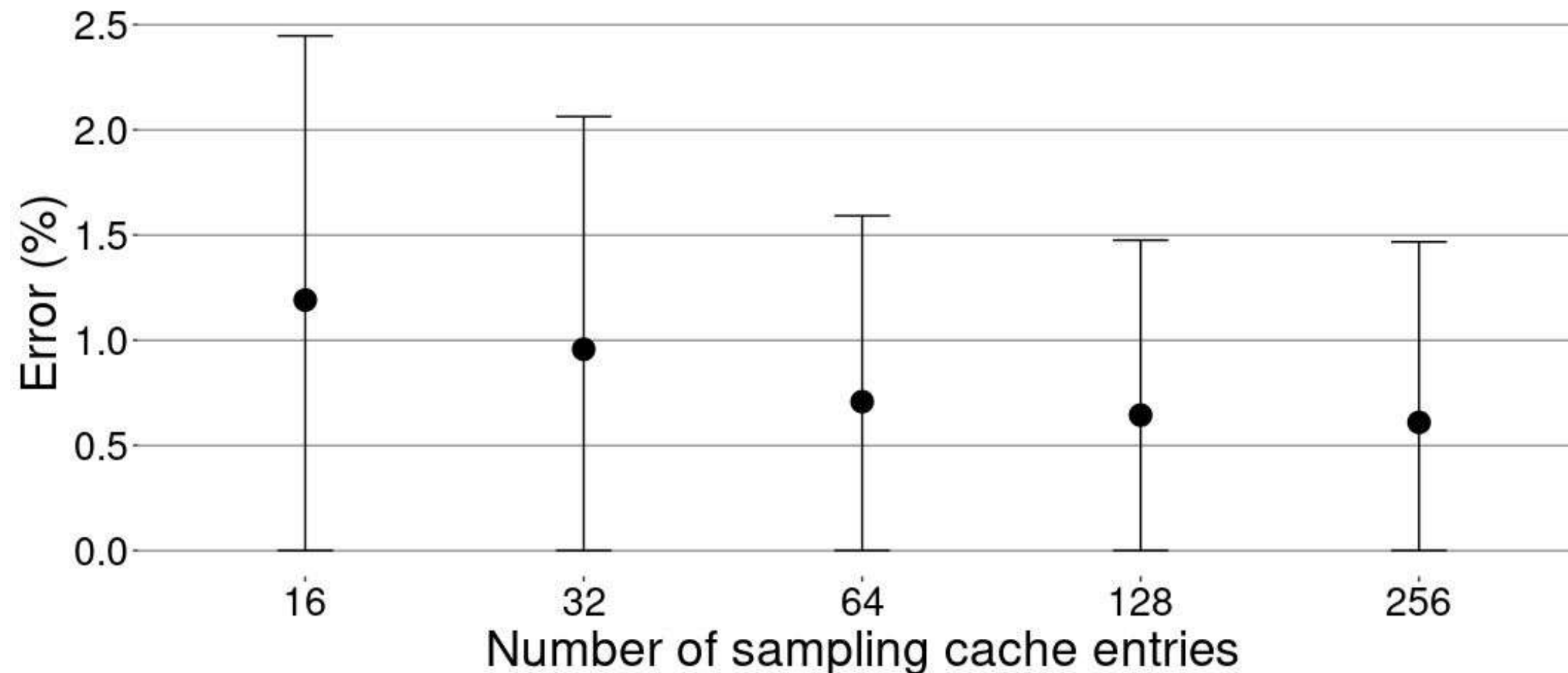
¹ Qureshi et al. (2006).

DYCA - *Sampling cache*



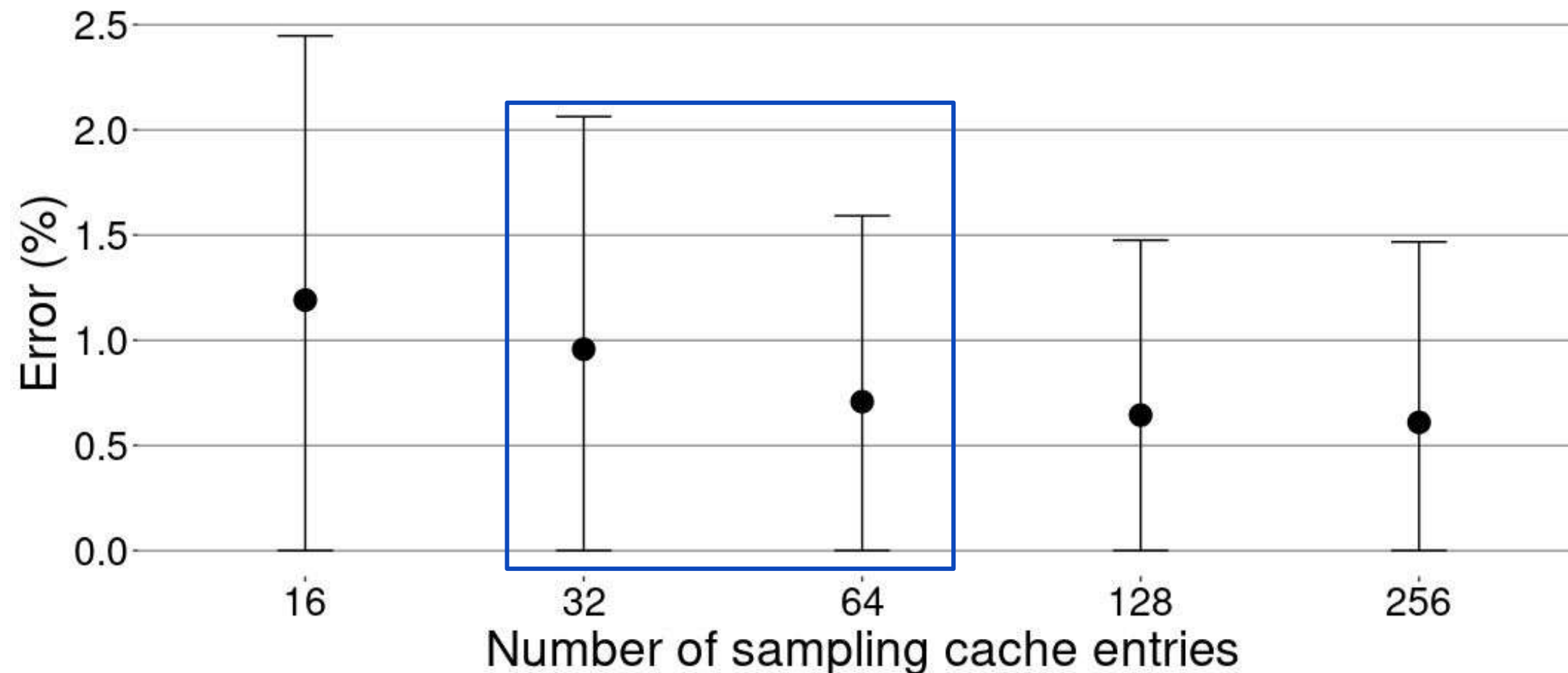
DYCA - *Sampling cache*

- Acurácia para um diferente número de entradas na *sampling cache*



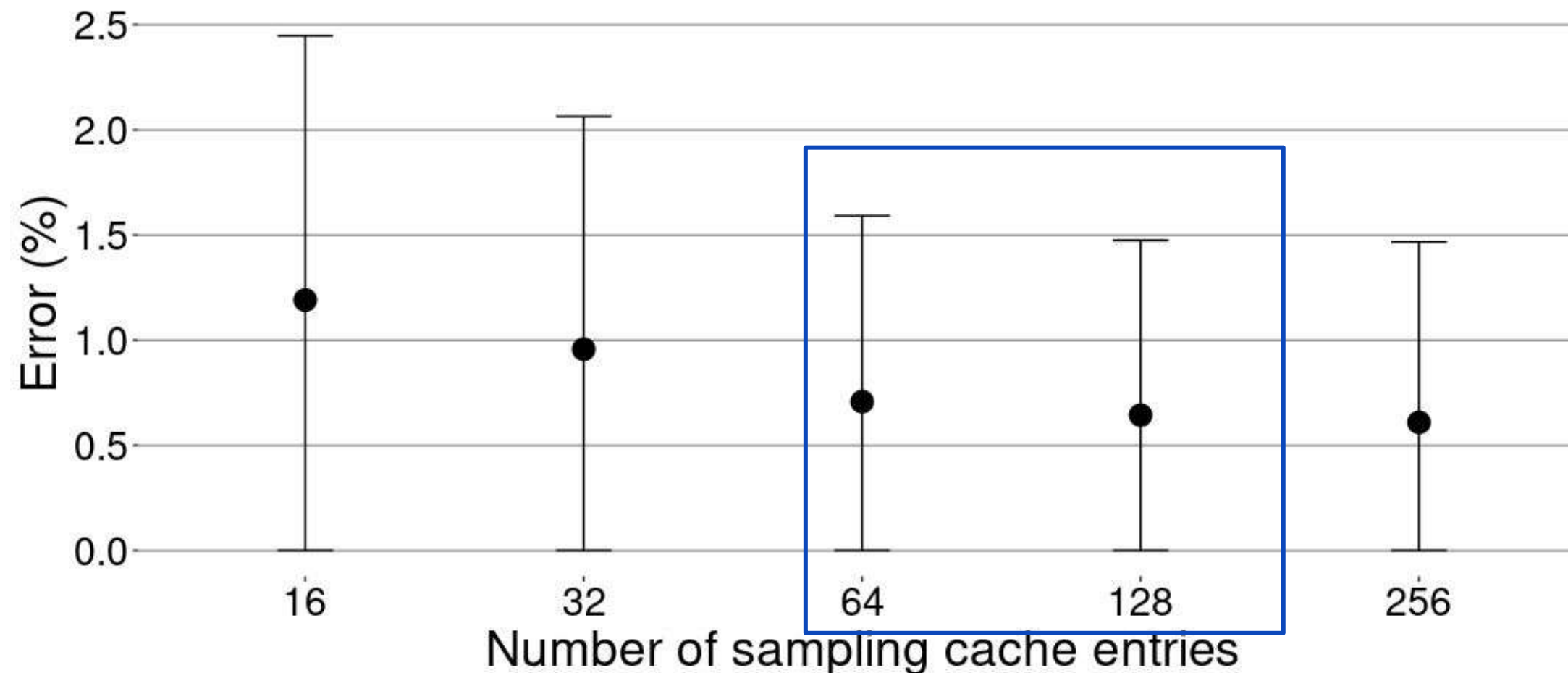
DYCA - *Sampling cache*

- Acurácia para um diferente número de entradas na *sampling cache*



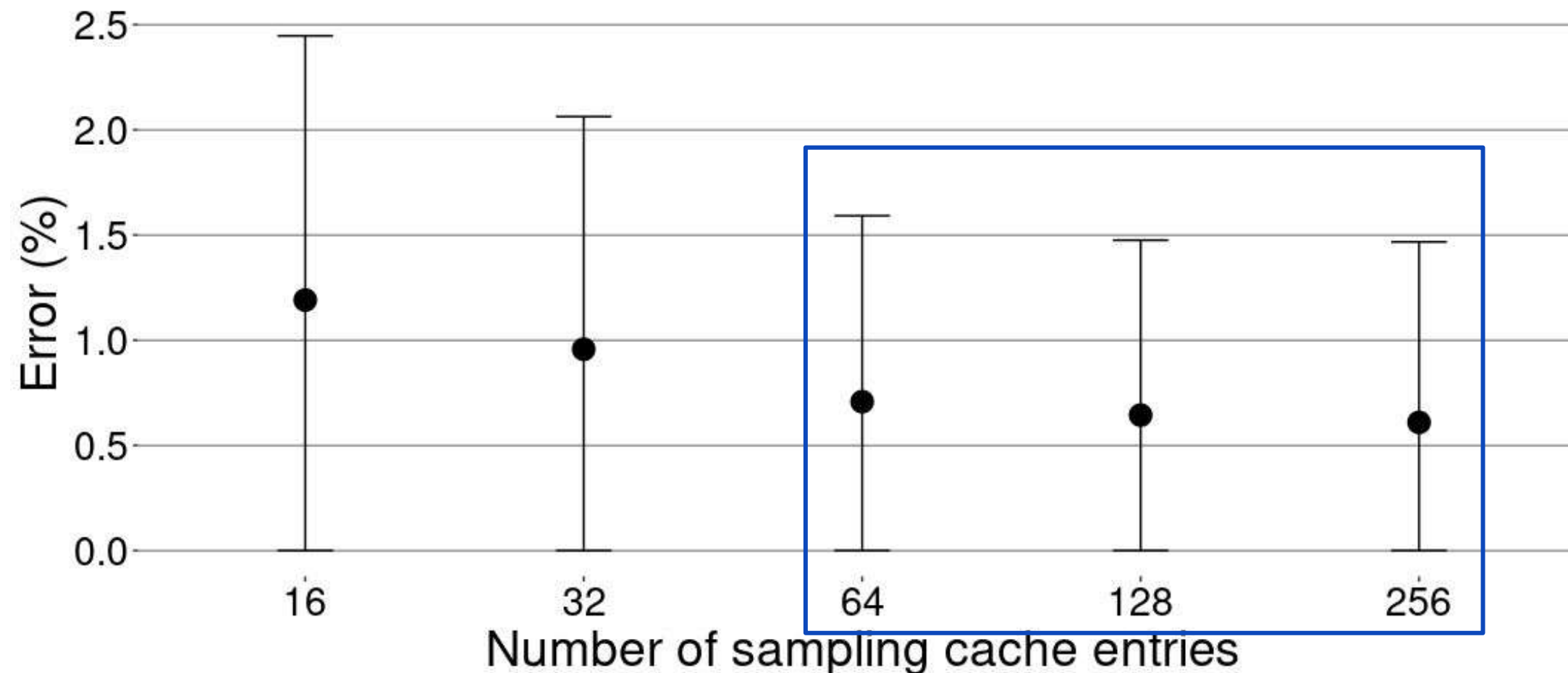
DYCA - *Sampling cache*

- Acurácia para um diferente número de entradas na *sampling cache*



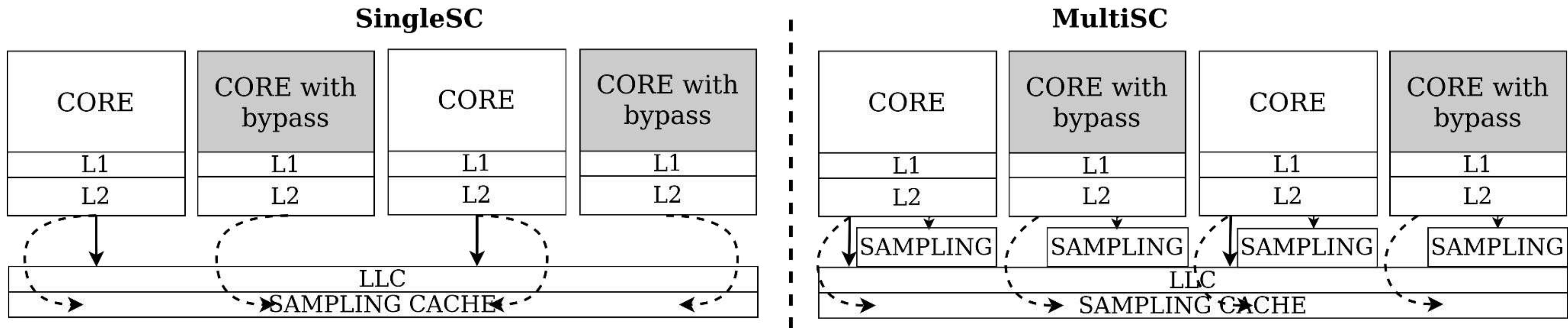
DYCA - *Sampling cache*

- Acurácia para um diferente número de entradas na *sampling cache*



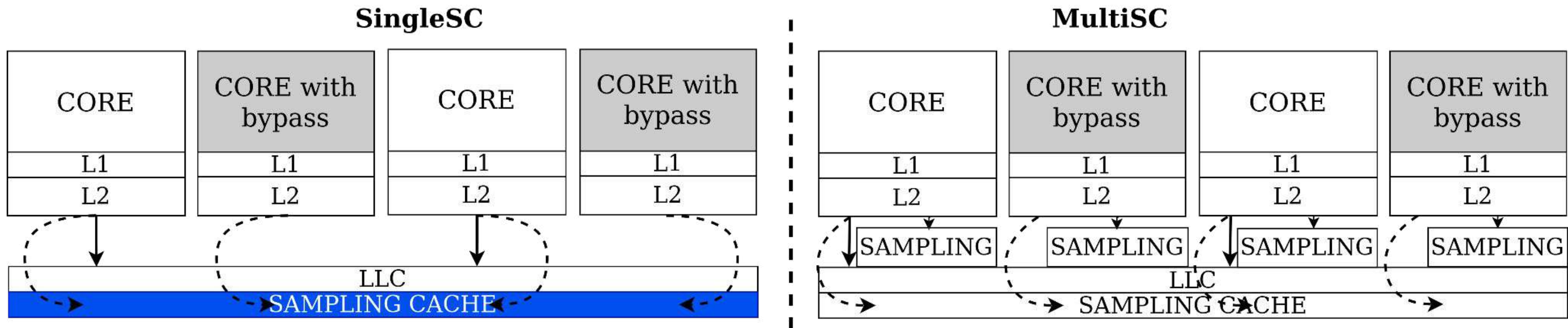
DYCA - *Sampling cache*

- Arquitetura com a *sampling cache*



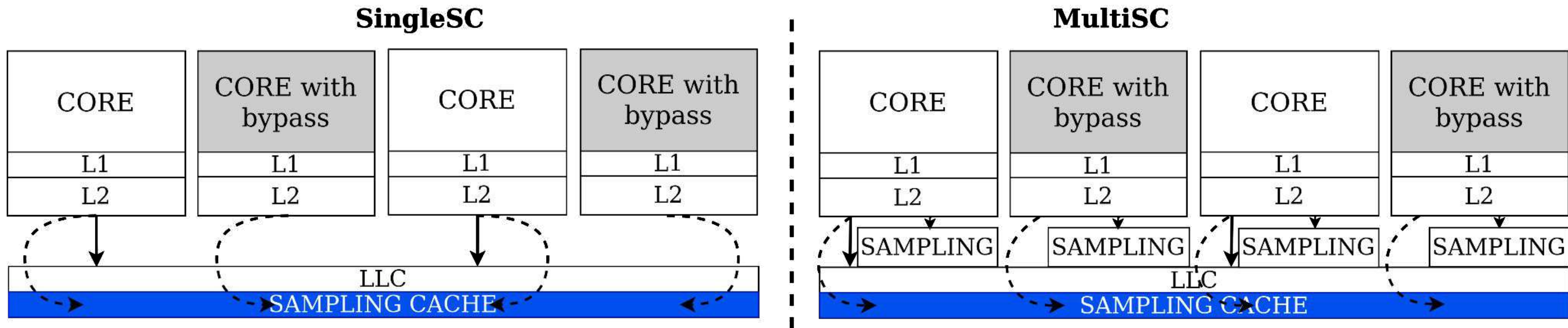
DYCA - *Sampling cache*

- Arquitetura com a *sampling cache*



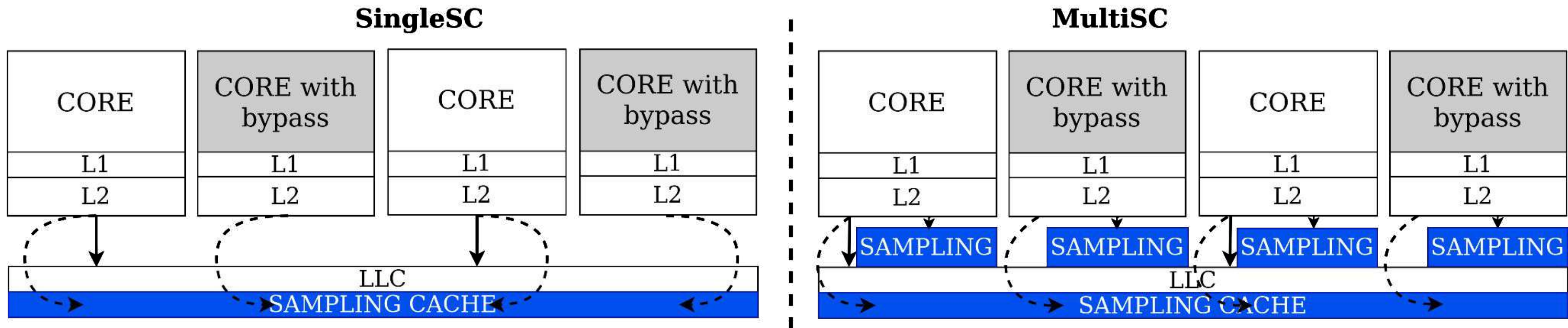
DYCA - *Sampling cache*

- Arquitetura com a *sampling cache*



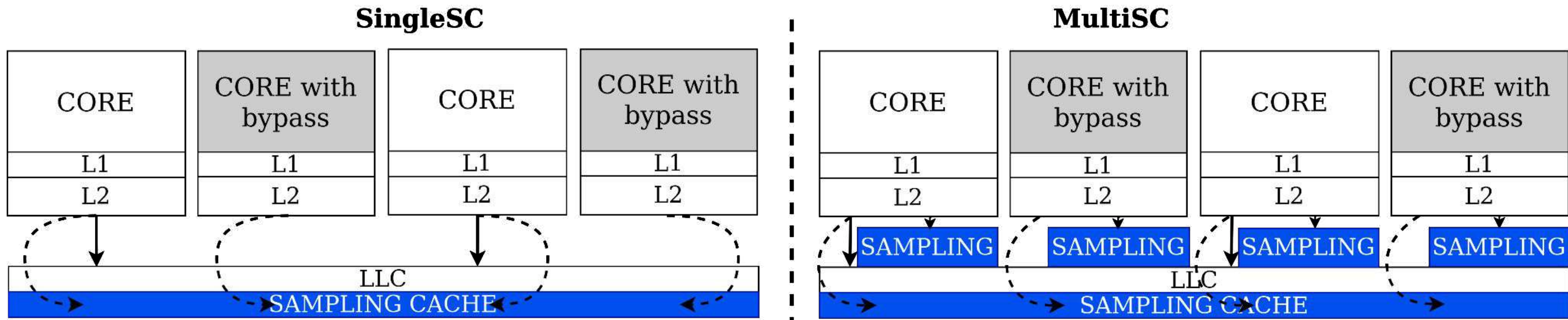
DYCA - *Sampling cache*

- Arquitetura com a *sampling cache*



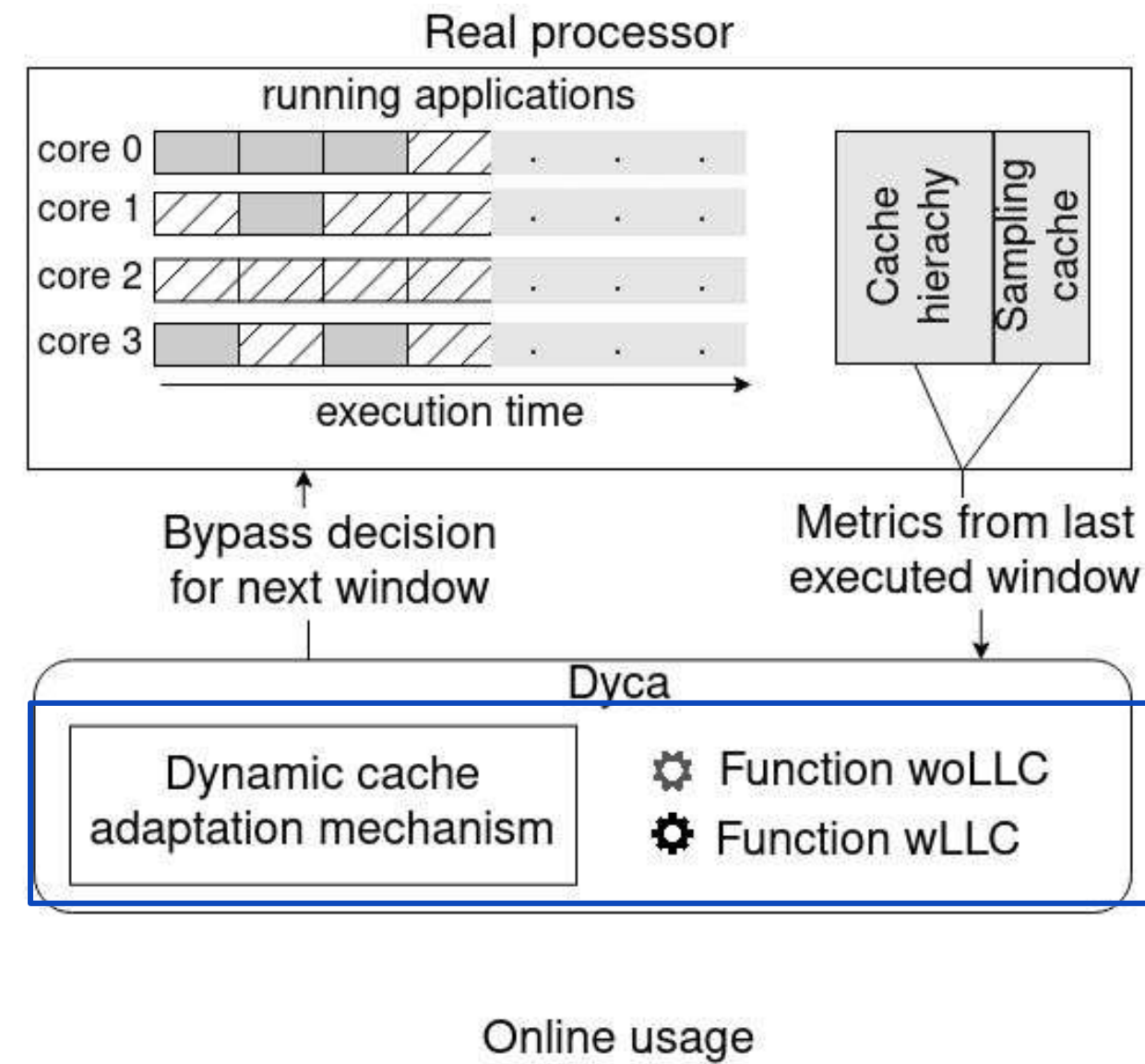
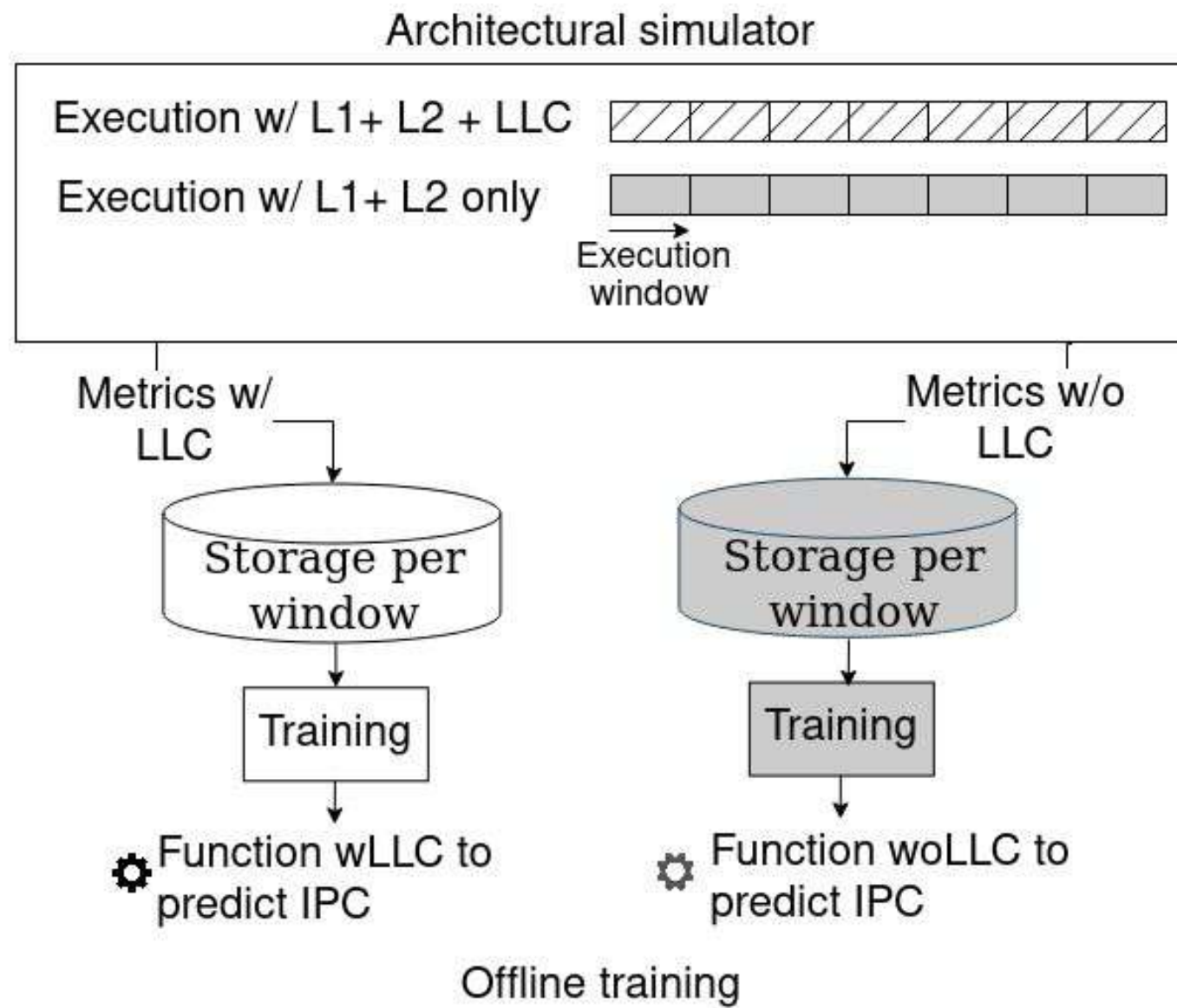
DYCA - *Sampling cache*

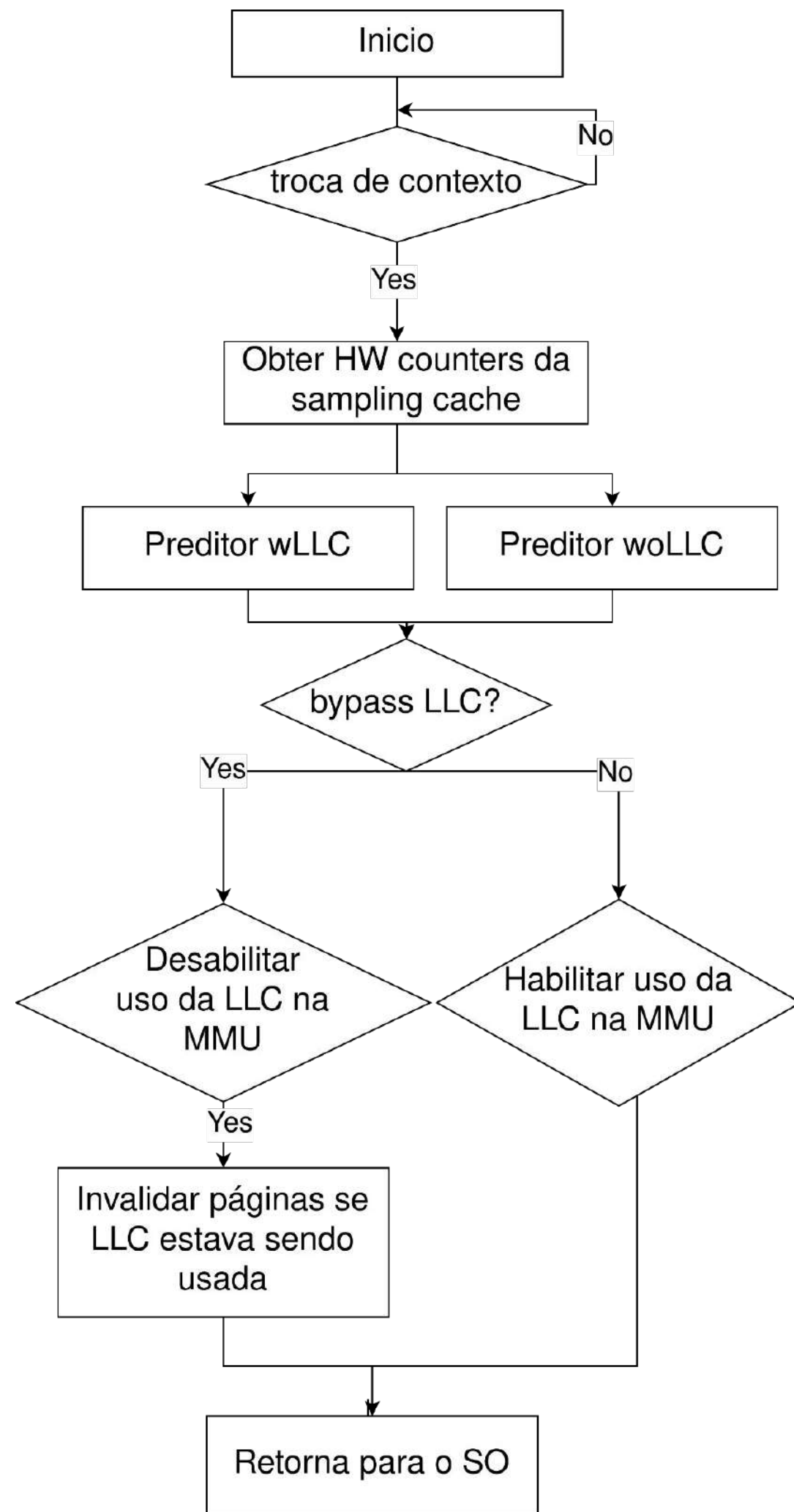
- Arquitetura com a *sampling cache*



Acurácia 2% maior

DYCA





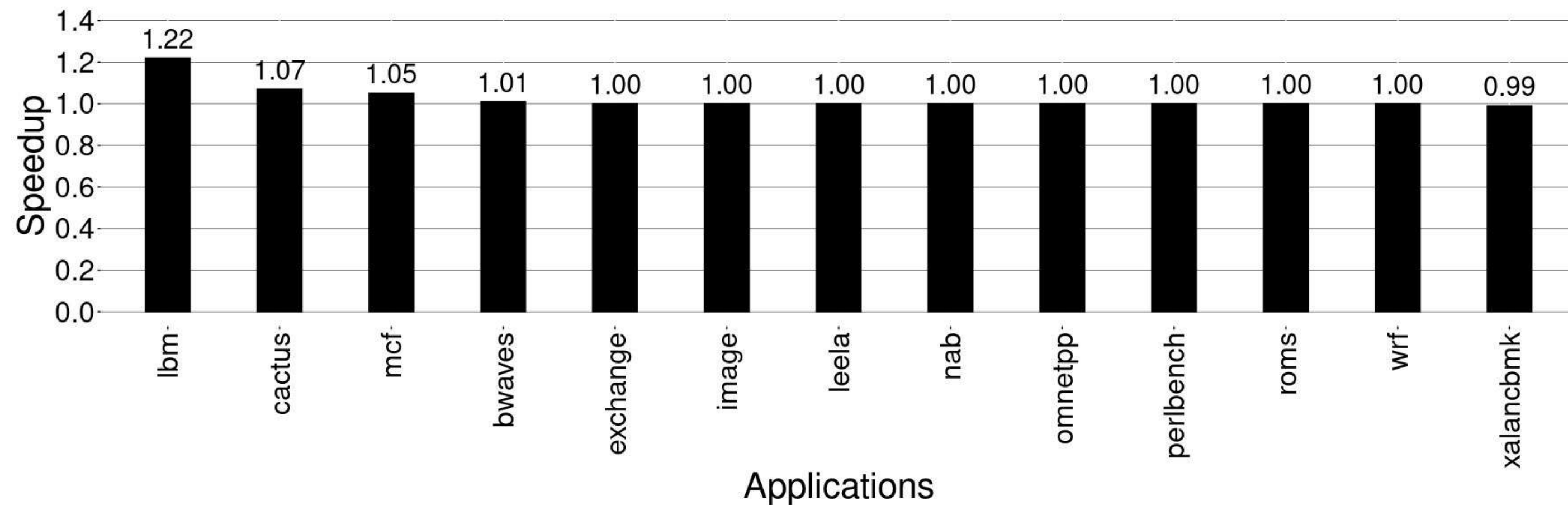
Resultados

- Abordagem estática
- *Single application*
- Mecanismo oráculo
- Multiple applications

Resultados - Simulação

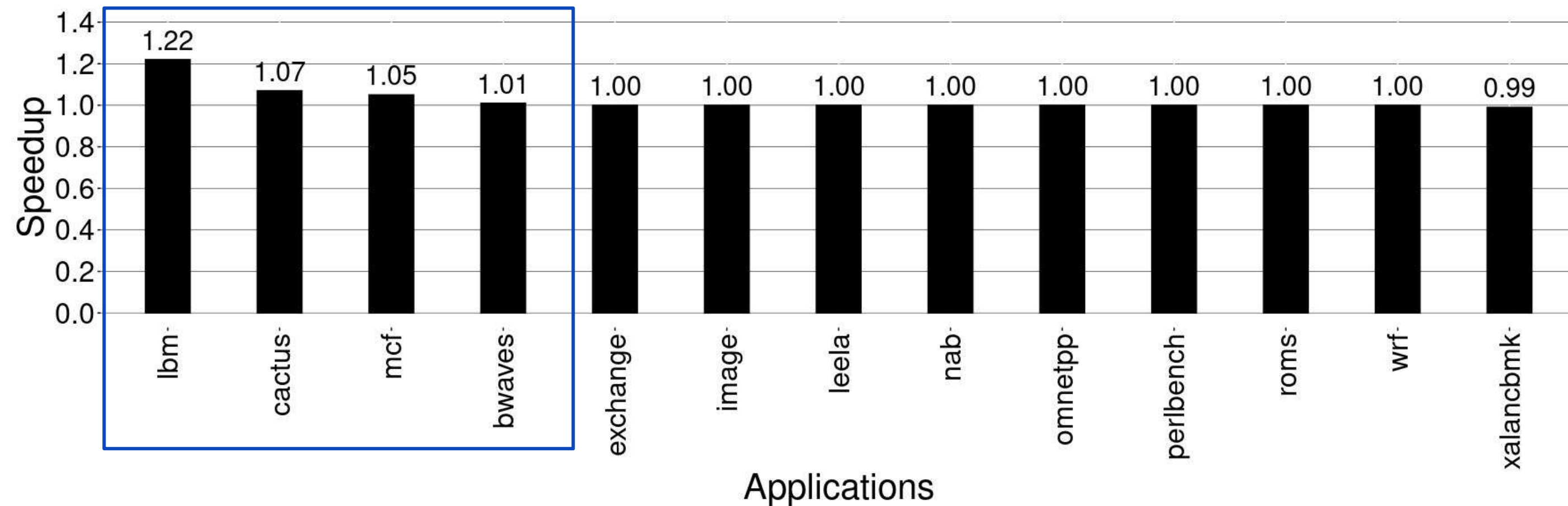
- Simulador Ordinary Computer Simulator (OrCS)
 - Acurácia de ciclos
 - Orientado a traços
 - Baseado no SiNUCA¹
- Traços com 2 bilhões de instruções mais representativas dos *benchmarks*
 - Obtido com a ferramenta PinPoints

Resultados - *Single application*



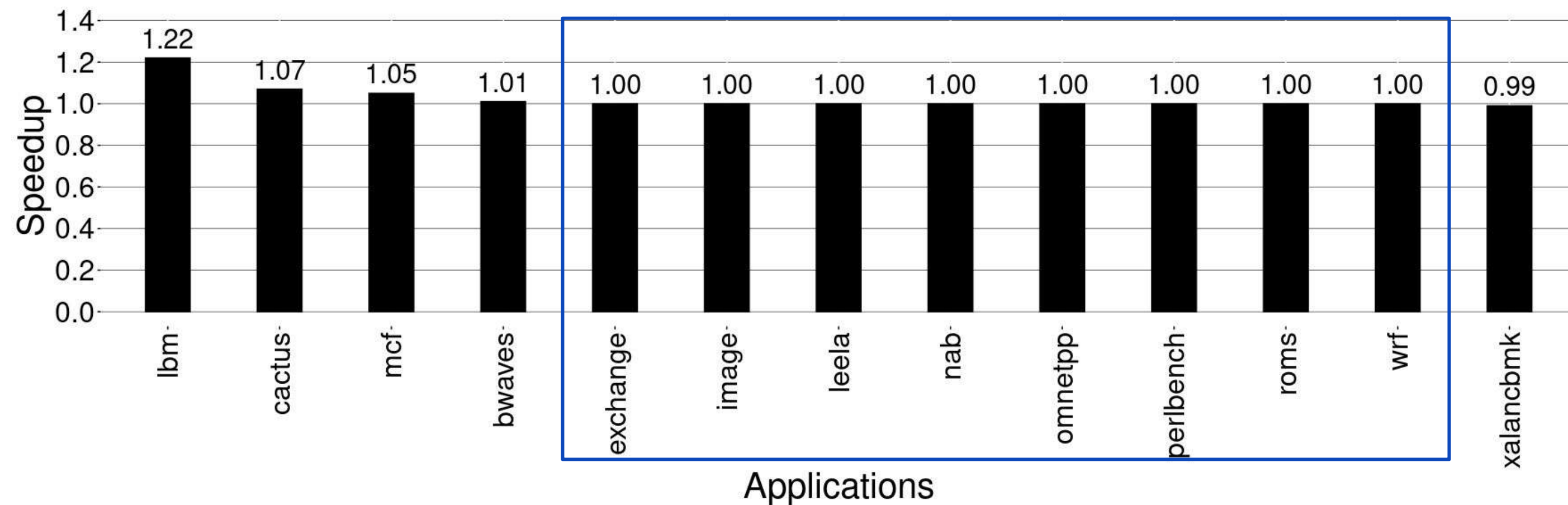
SPEC CPU 2017

Resultados - *Single application*



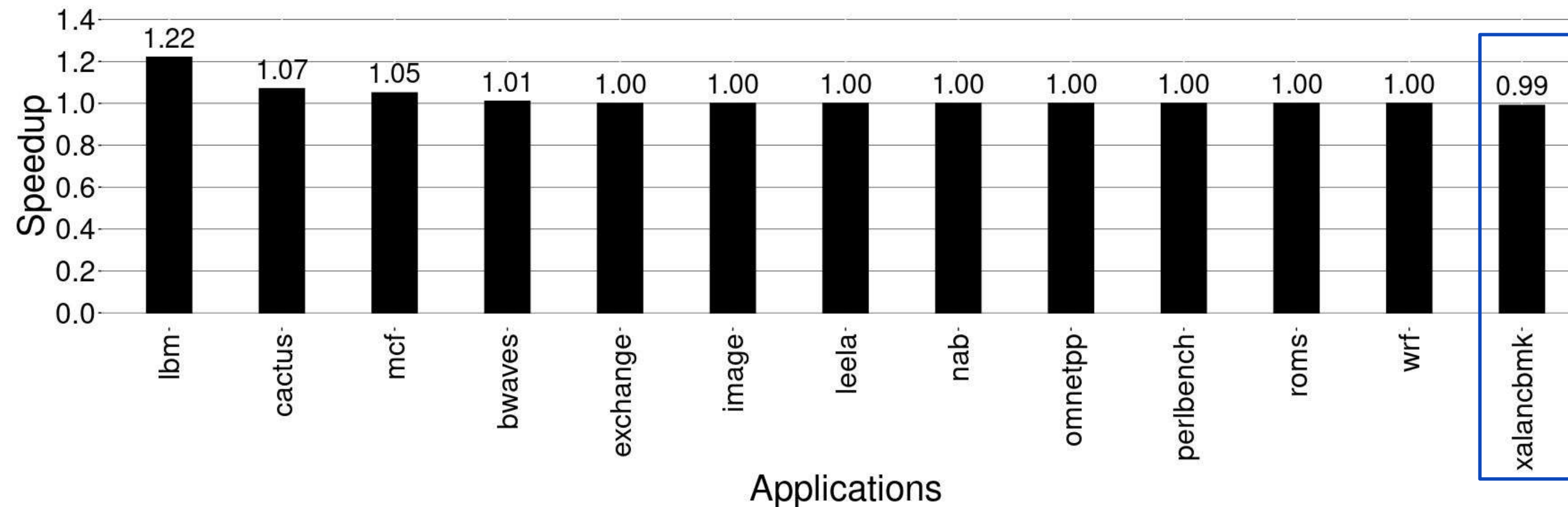
SPEC CPU 2017

Resultados - Single application



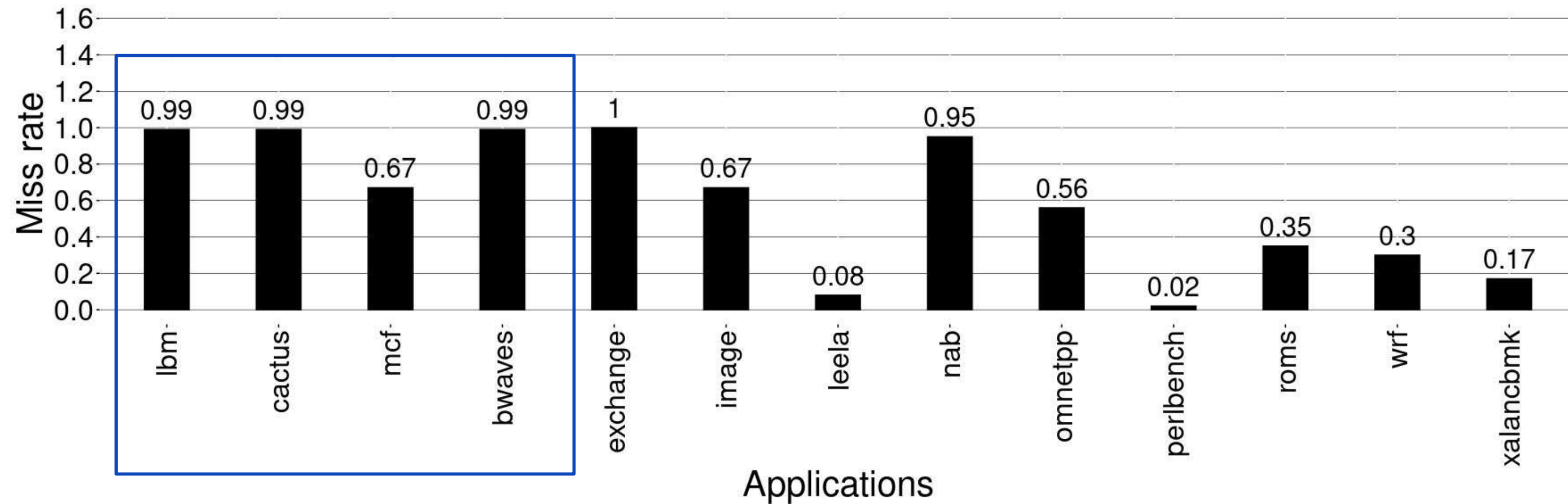
SPEC CPU 2017

Resultados - *Single application*



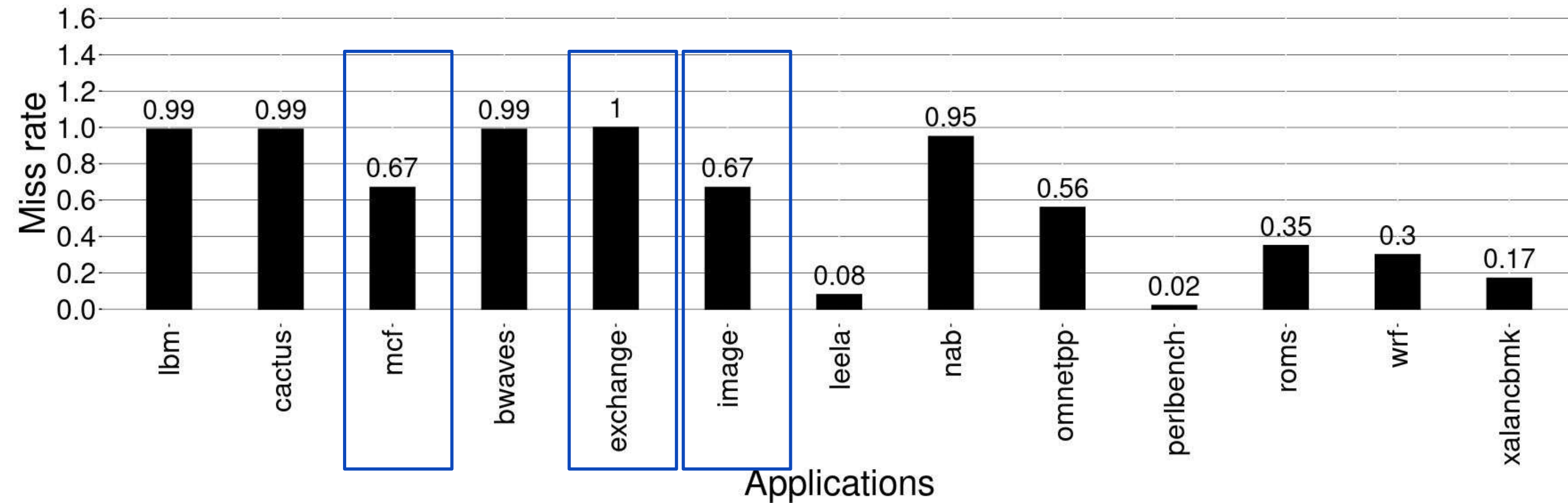
SPEC CPU 2017

Resultados - *Single application*



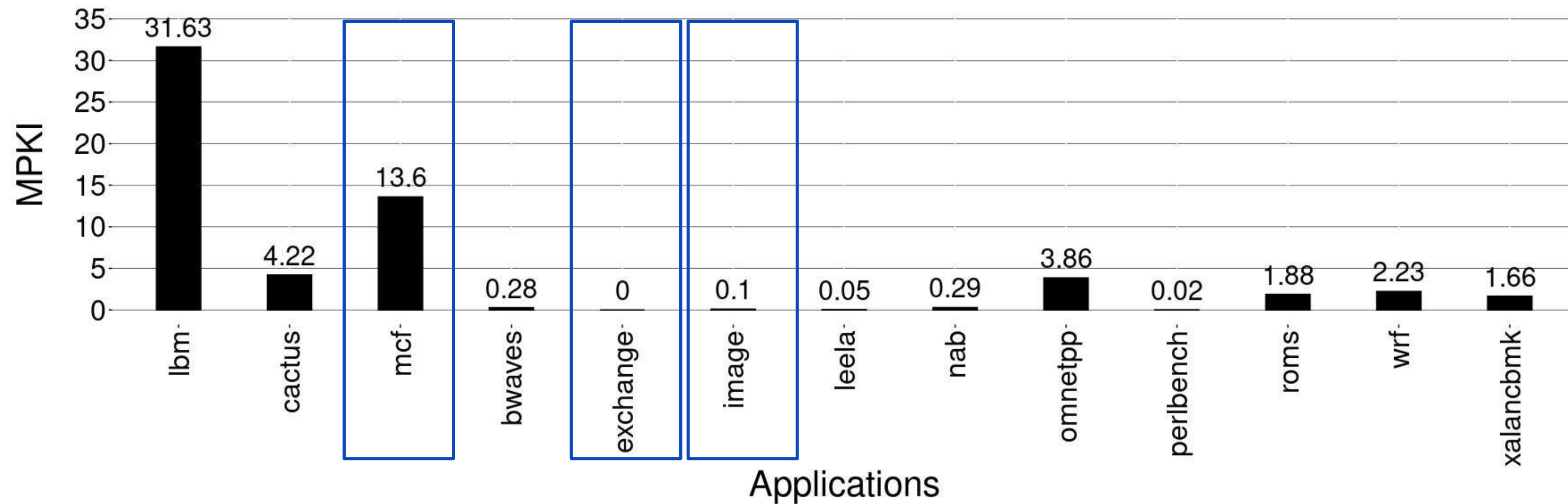
SPEC CPU 2017

Resultados - *Single application*



SPEC CPU 2017

Results - *Single application*

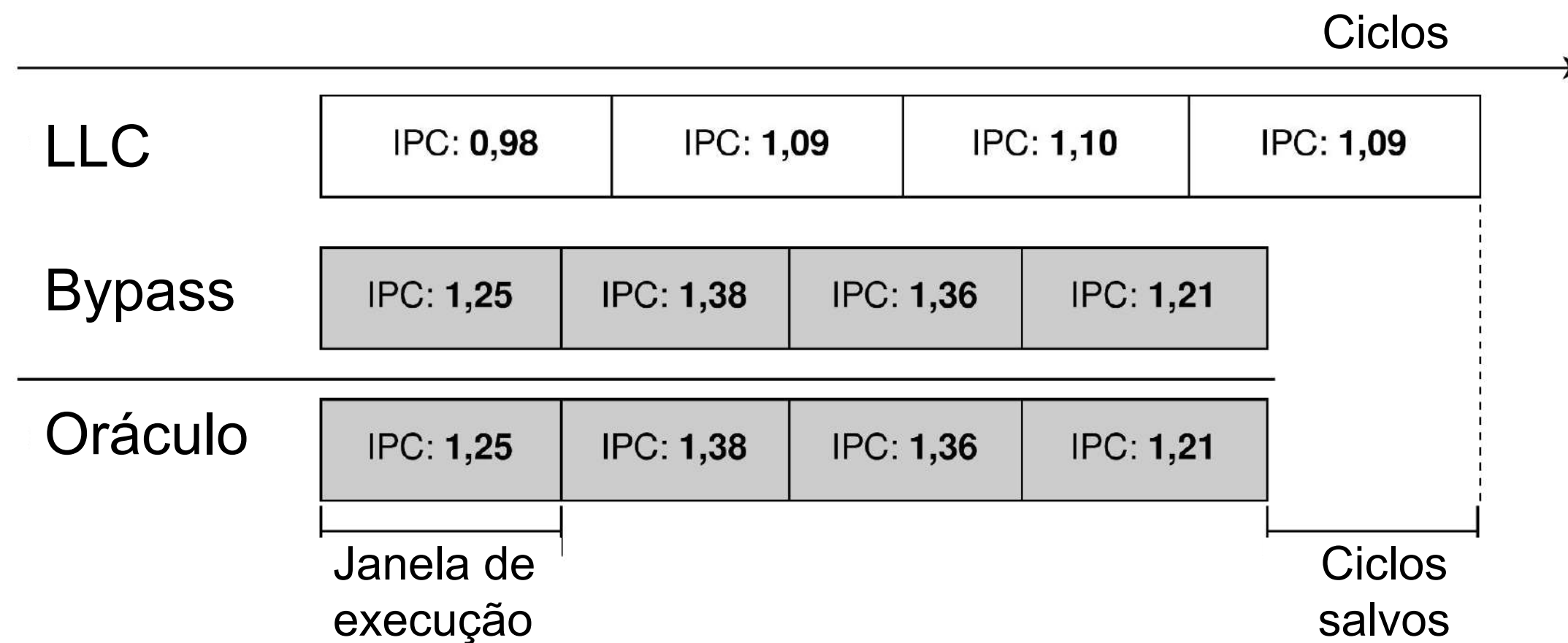


SPEC CPU 2017

Resultados - Mecanismo oráculo

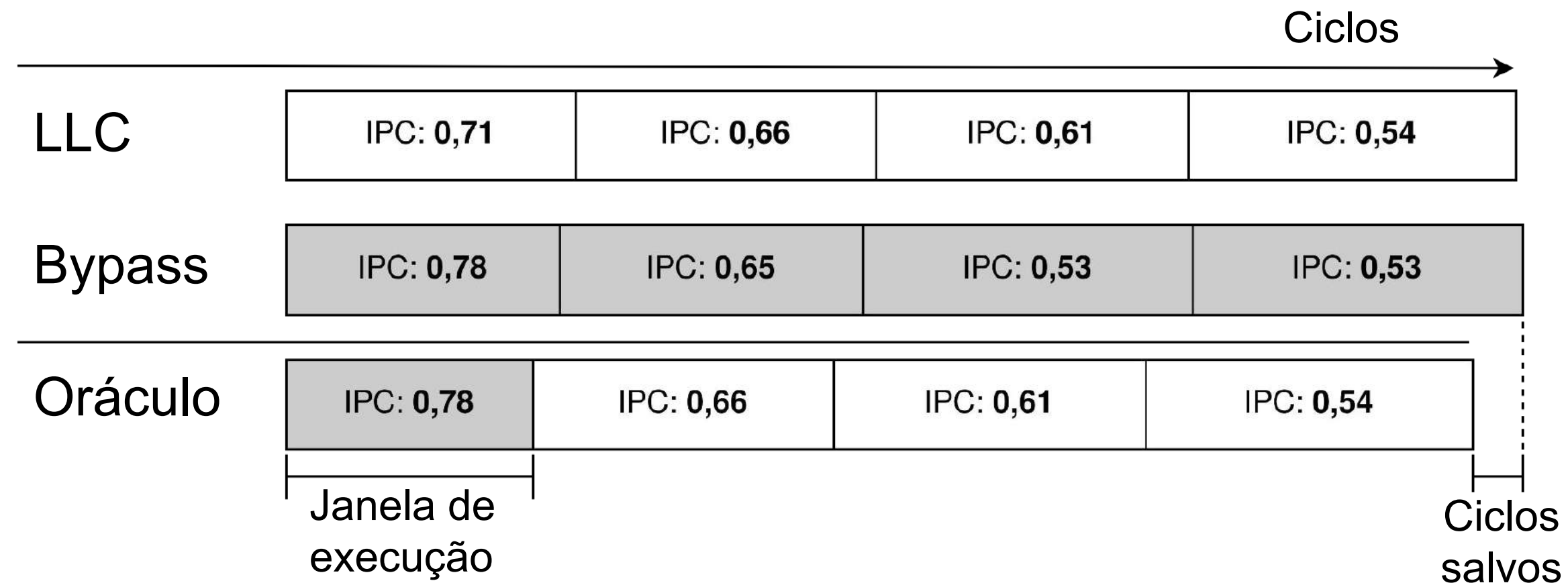
- Estimar o ganho máximo
- Usa duas execuções:
 - Uma usando a LLC (16MB)
 - Uma sem a LLC (0MB)
- Para cada janela de execução seleciona a configuração com o maior IPC
- Cria uma nova execução

Resultados - Mecanismo oráculo



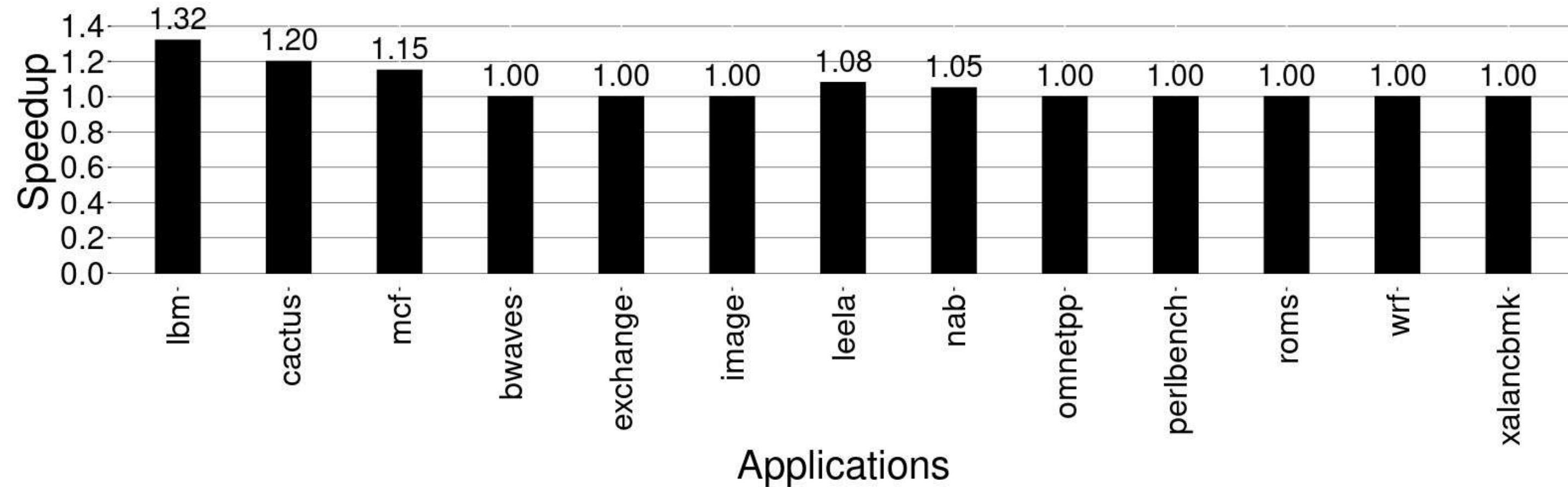
Aplicação *Ibm* (SPEC CPU 2017)

Resultados - Mecanismo oráculo



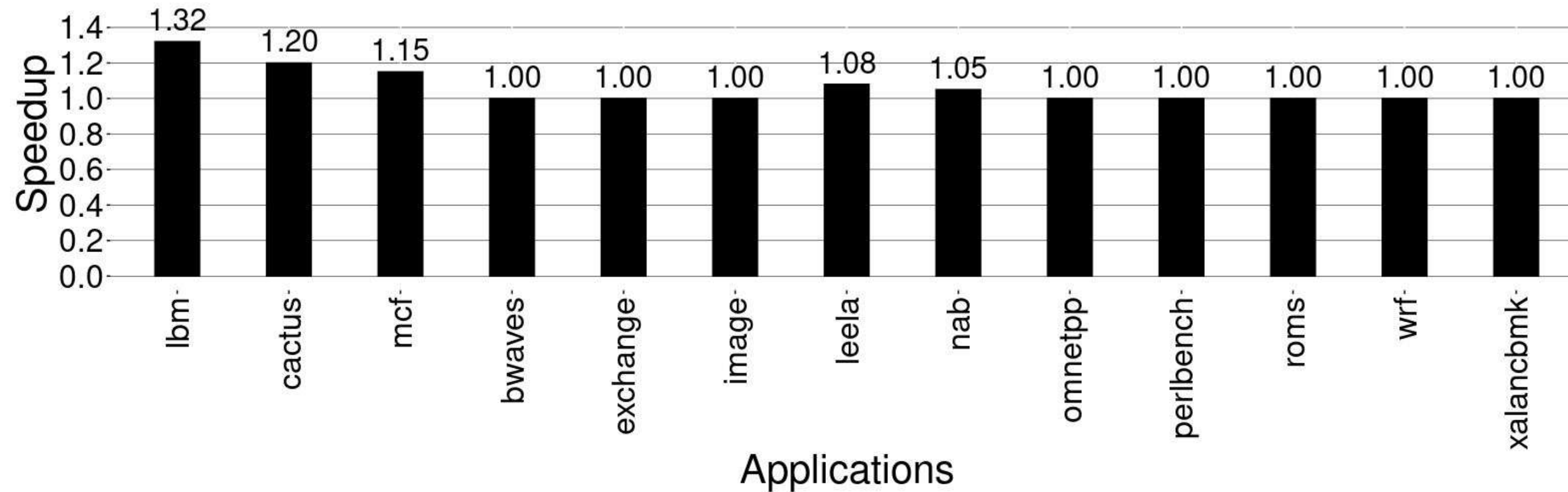
Aplicação *gcc* (SPEC CPU 2006)

Resultados - Mecanismo oráculo

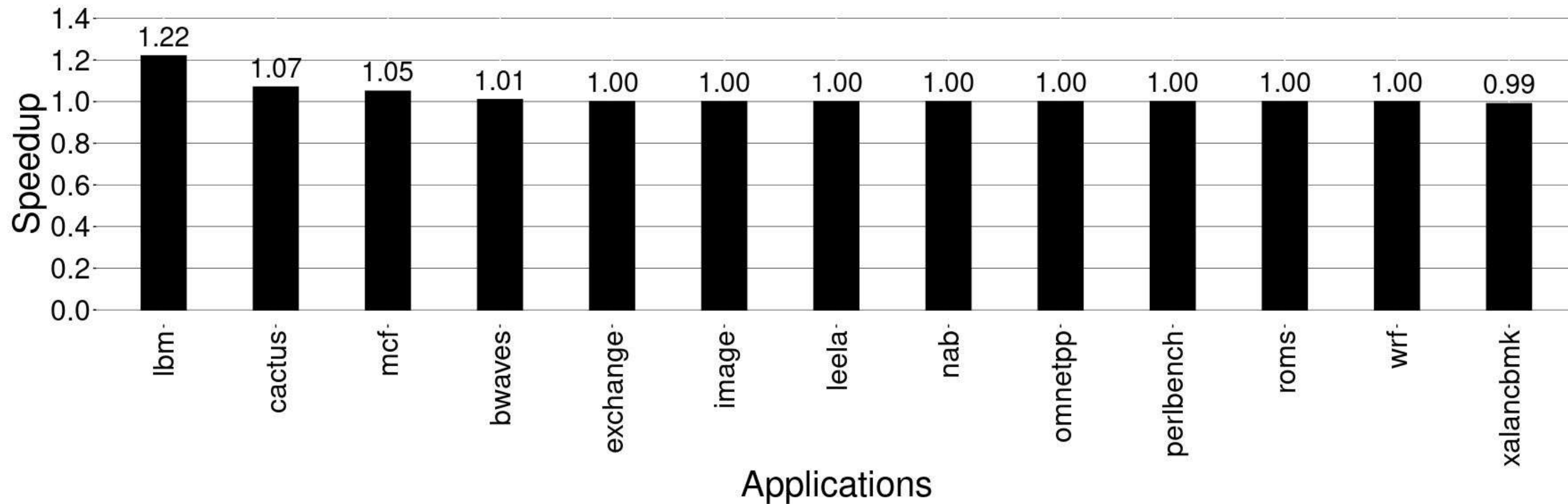


SPEC CPU 2017

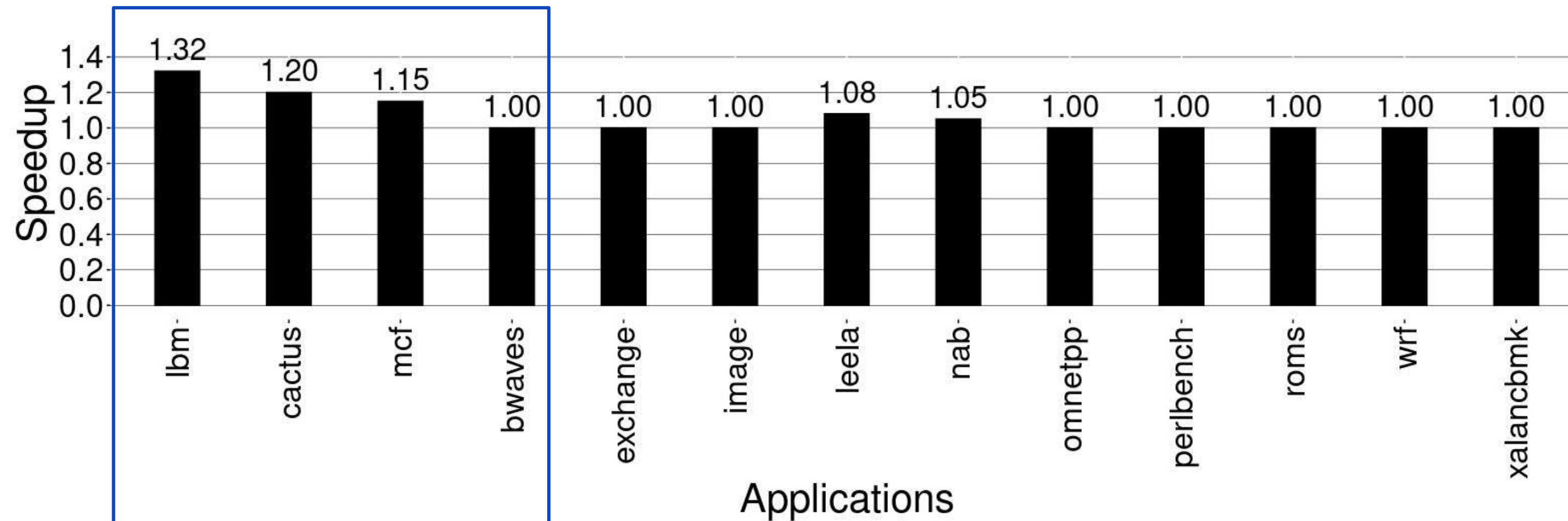
oráculo



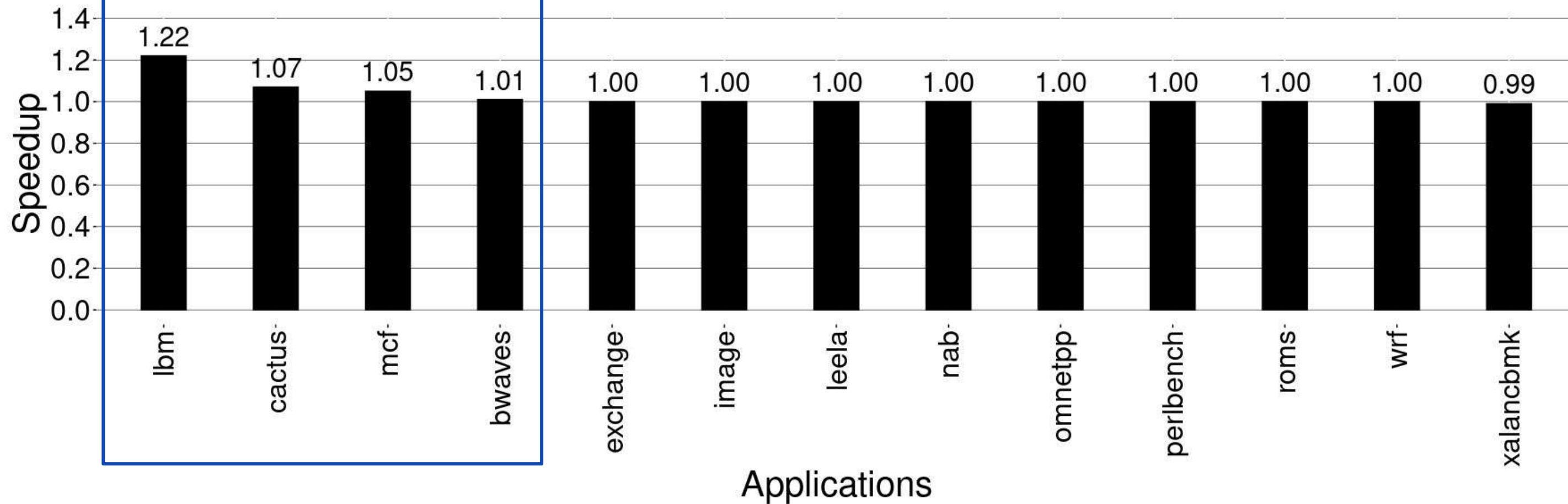
DYCA



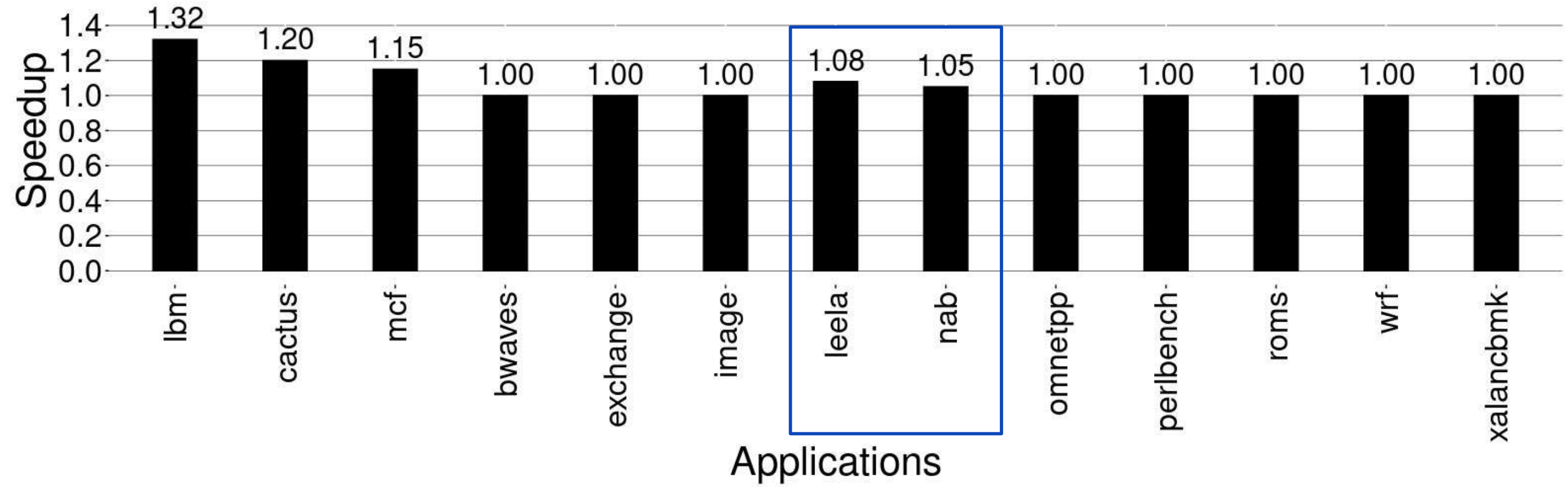
oráculo



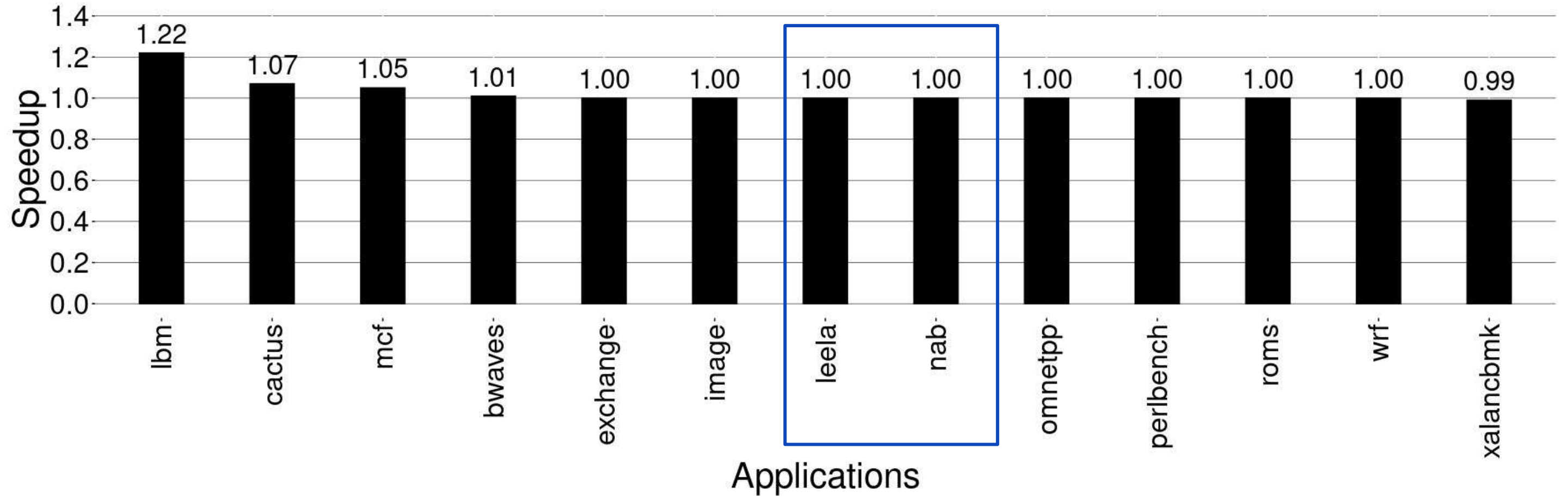
DYCA



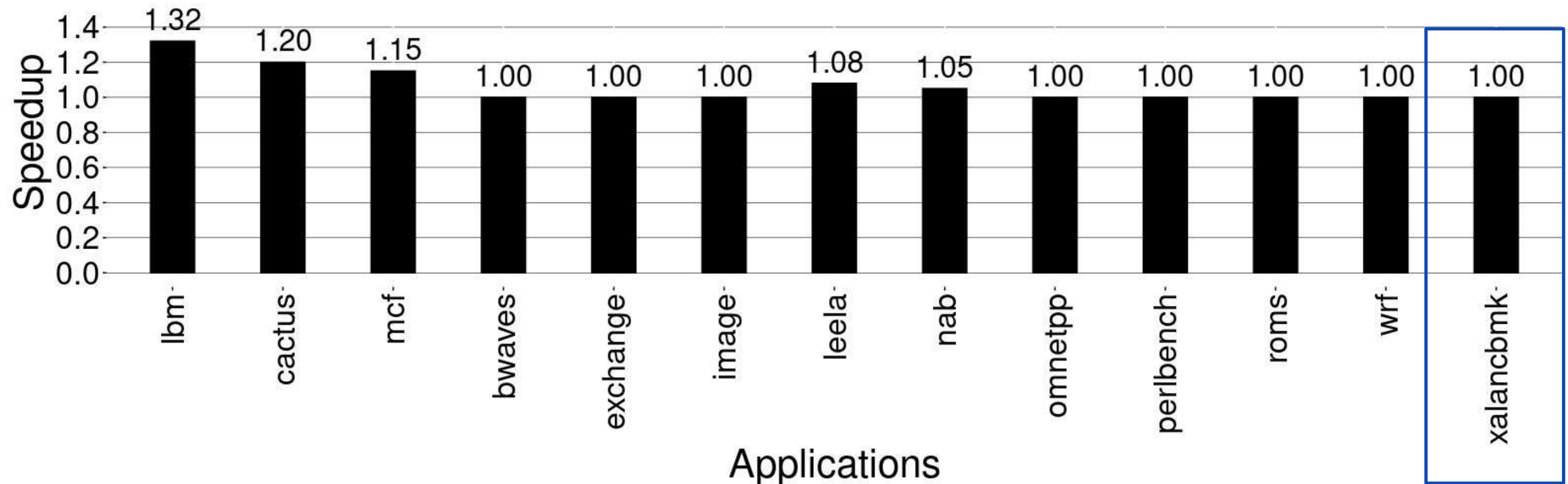
oráculo



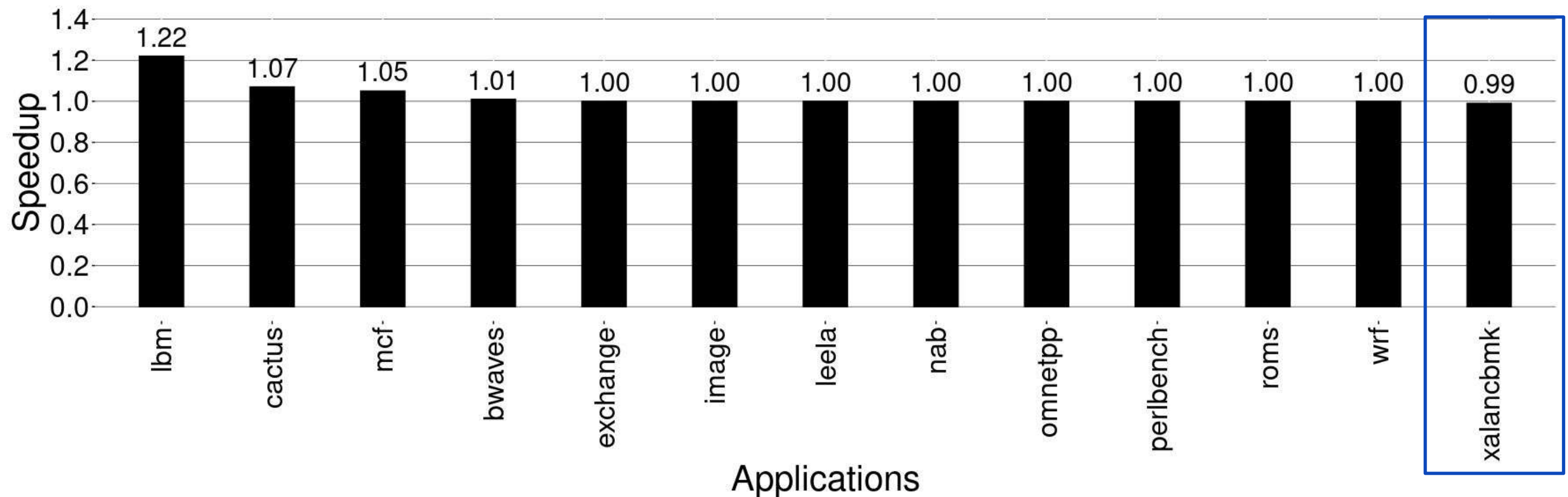
DYCA



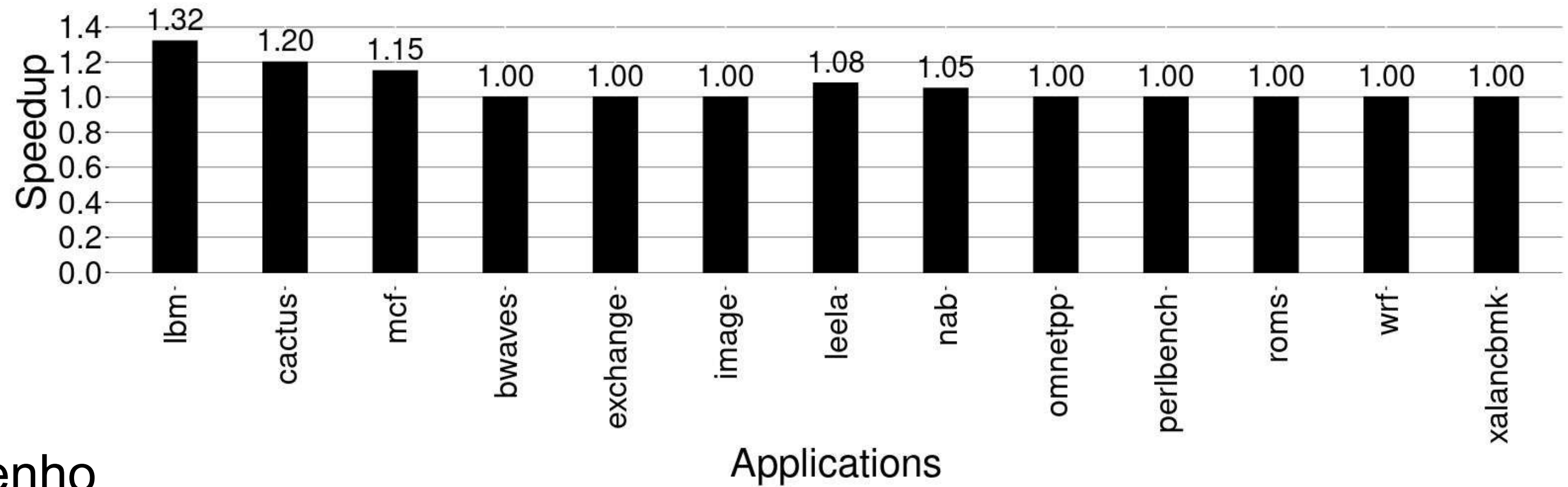
oráculo



DYCA

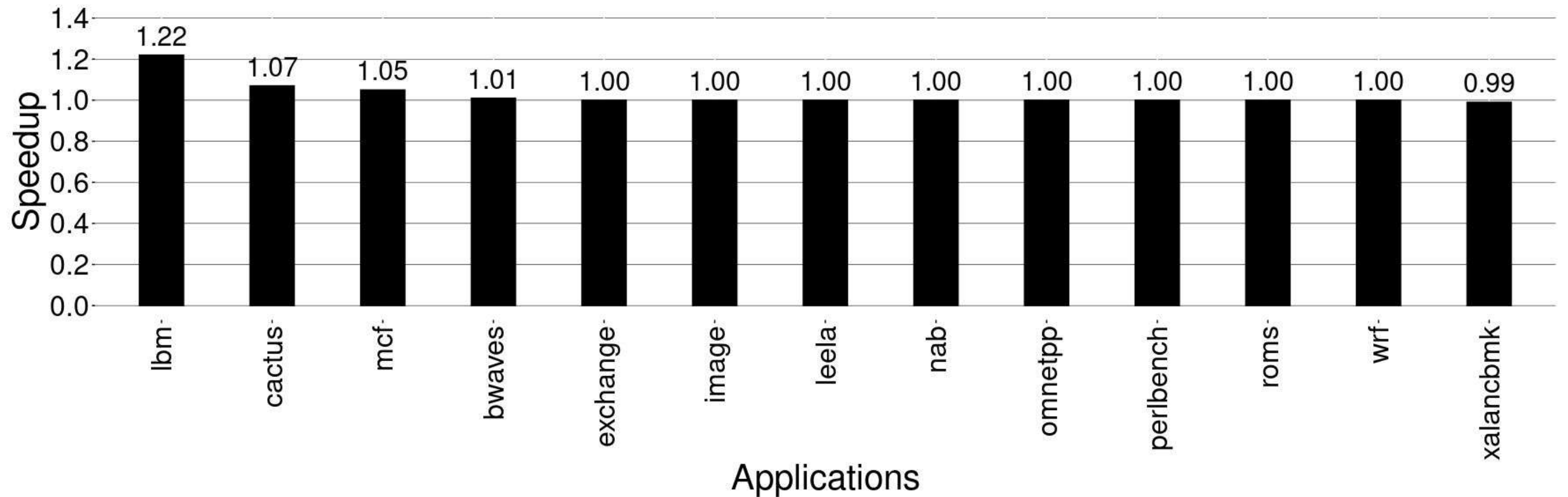


oráculo



46% de desempenho

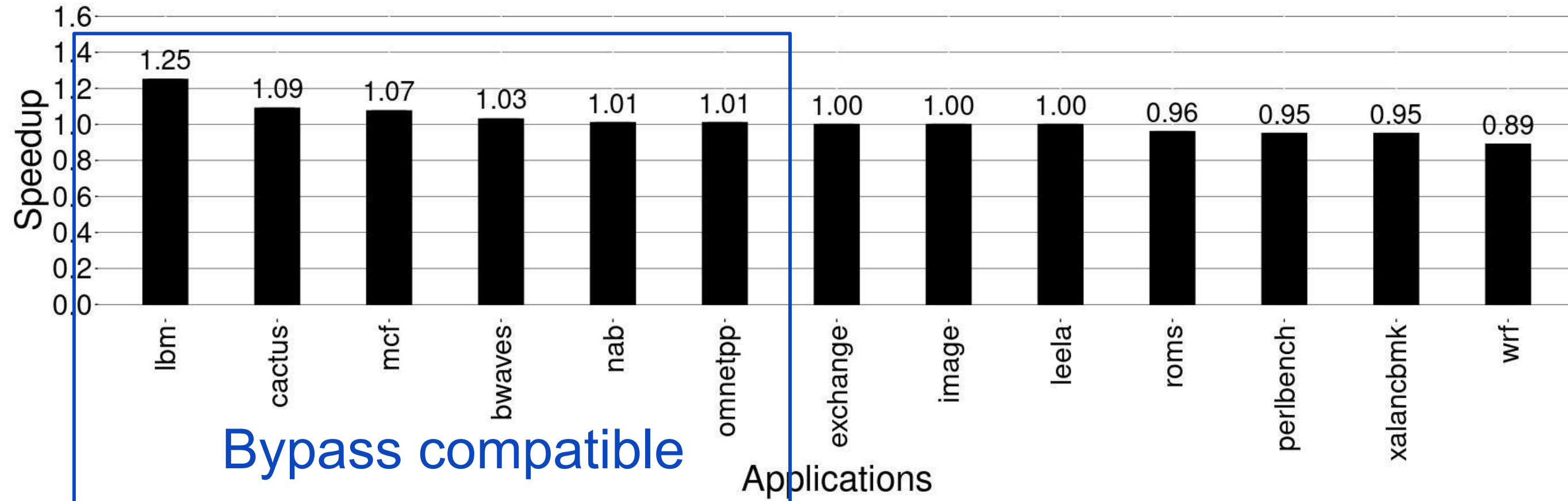
DYCA



Resultados - *Multiple applications*

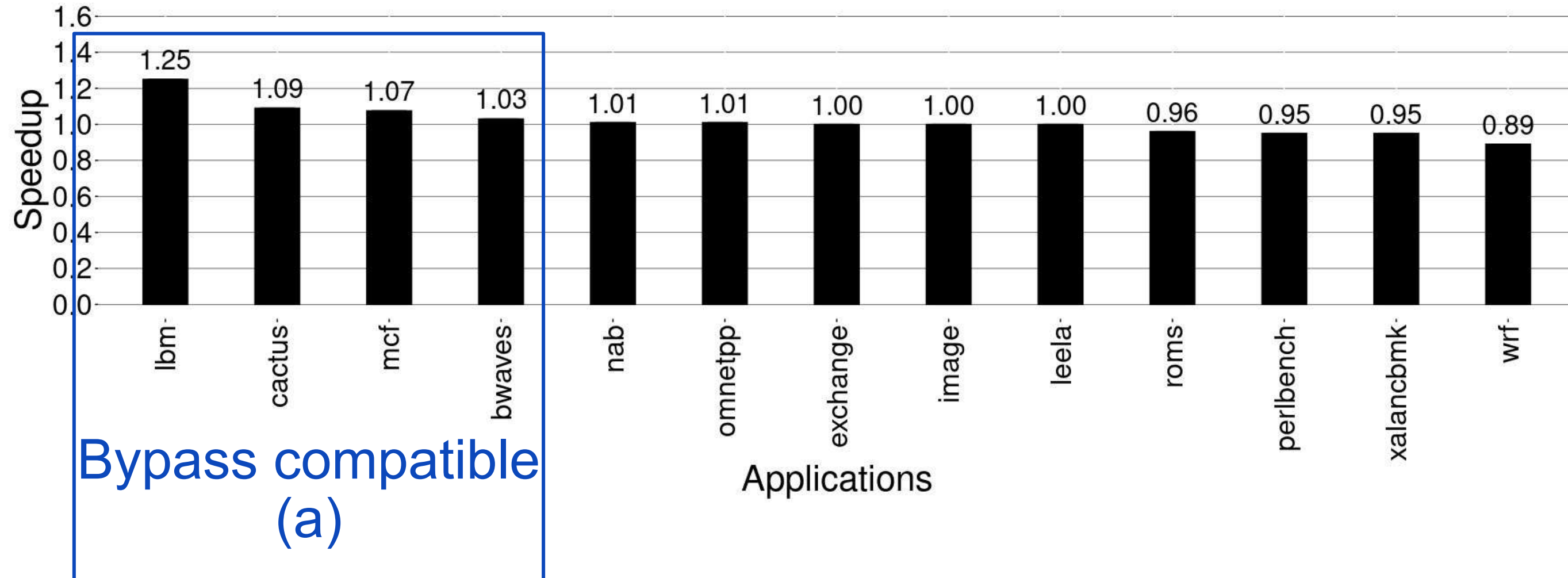
- Grupos de 4 aplicações
 - Baseado no comportamento
 - Ganham
 - Perdem
 - Misto
 - Randômico

Resultados - *Multiple applications*



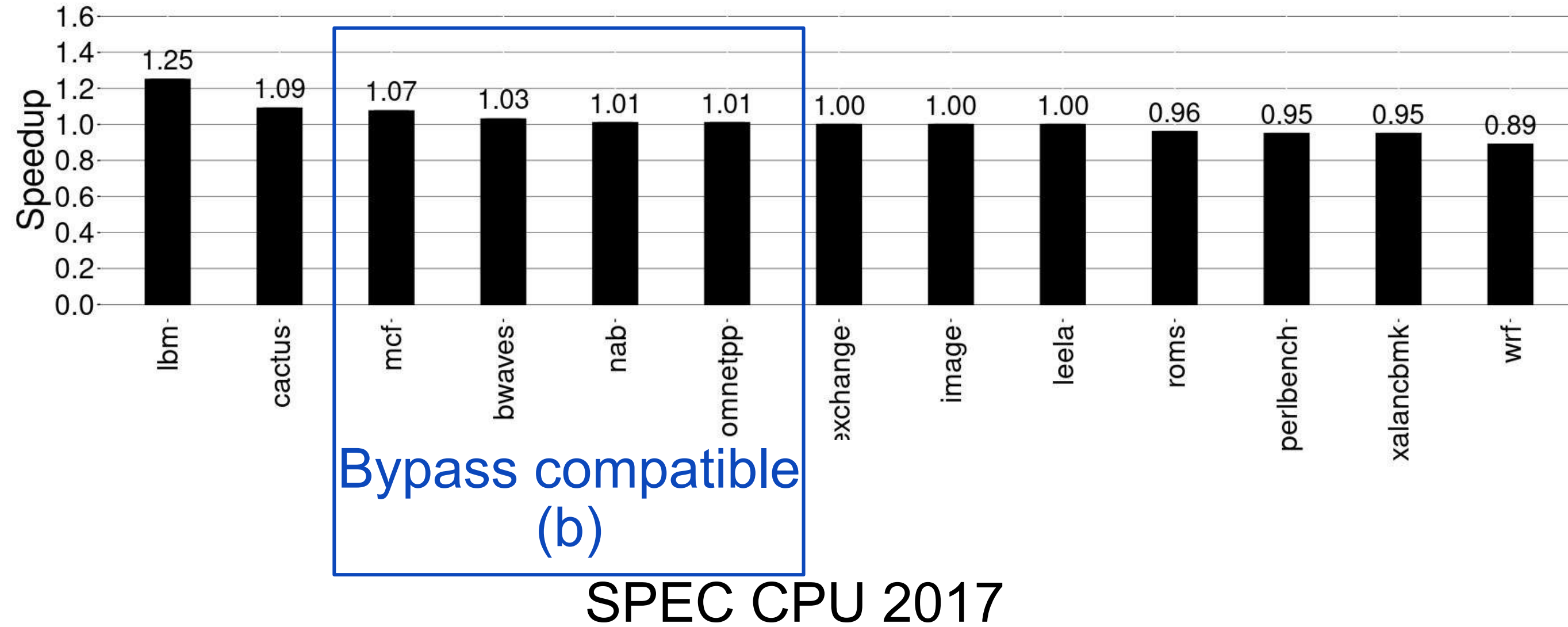
SPEC CPU 2017

Resultados - *Multiple applications*

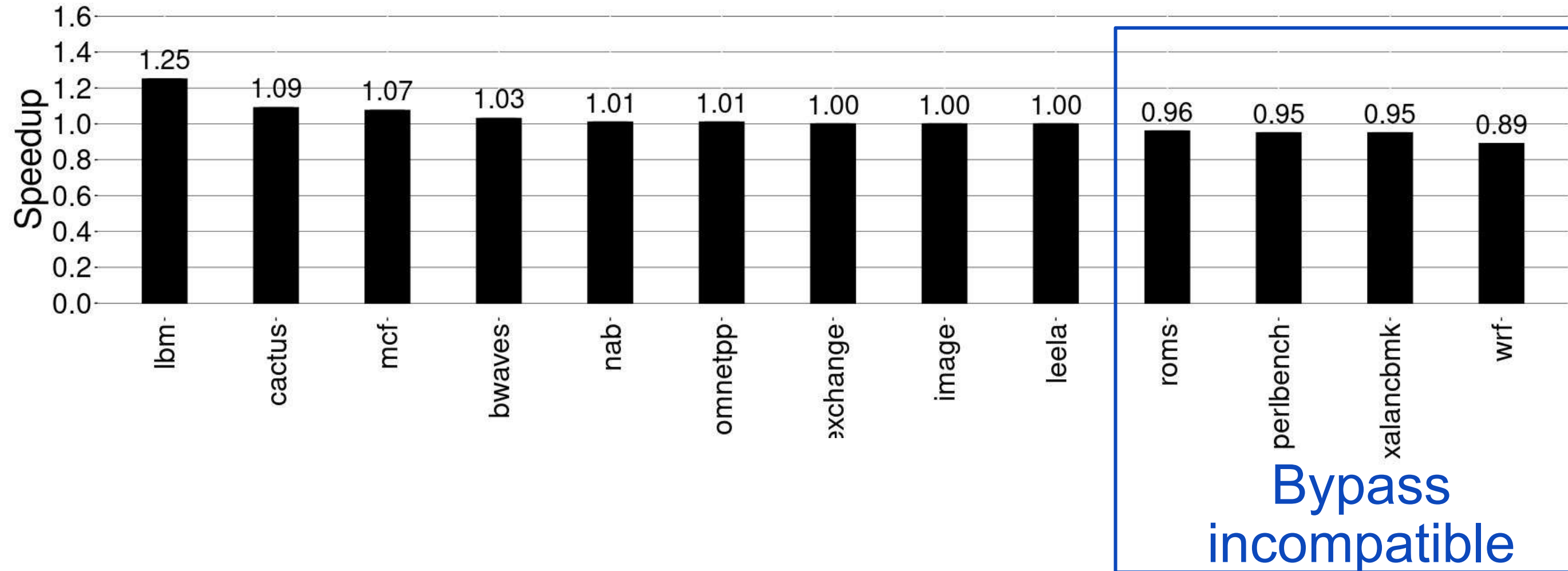


SPEC CPU 2017

Resultados - *Multiple applications*

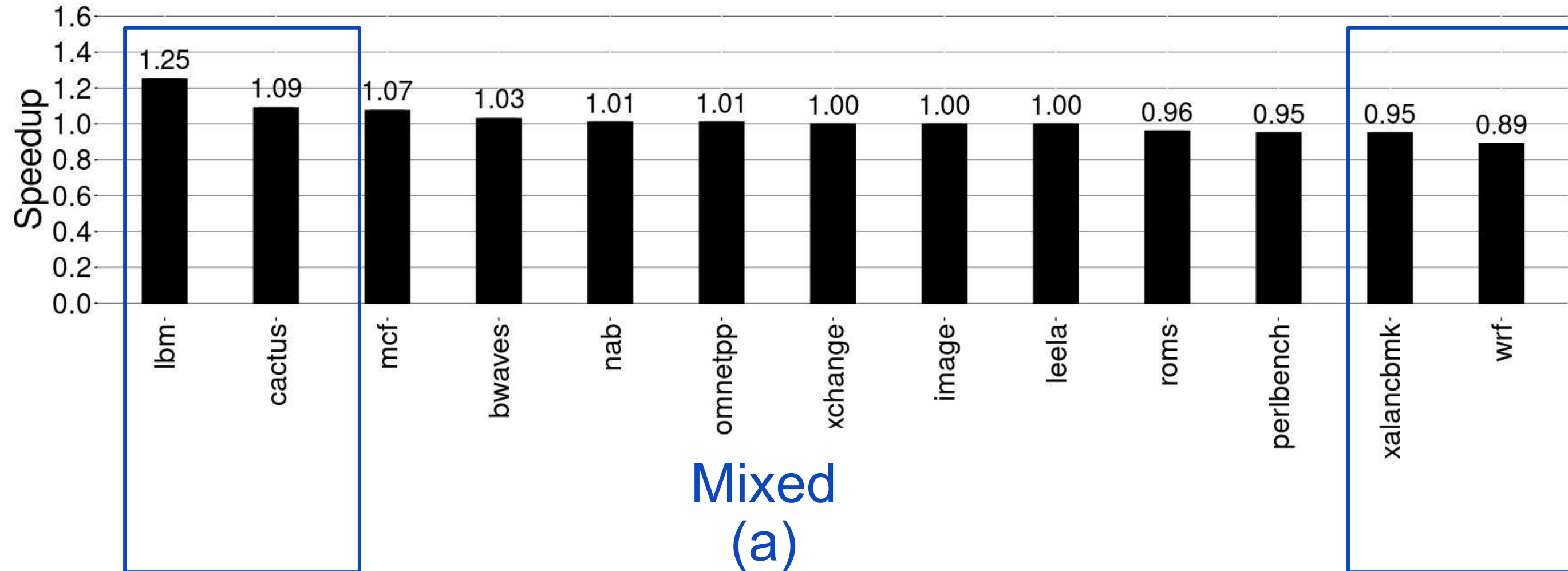


Resultados - *Multiple applications*



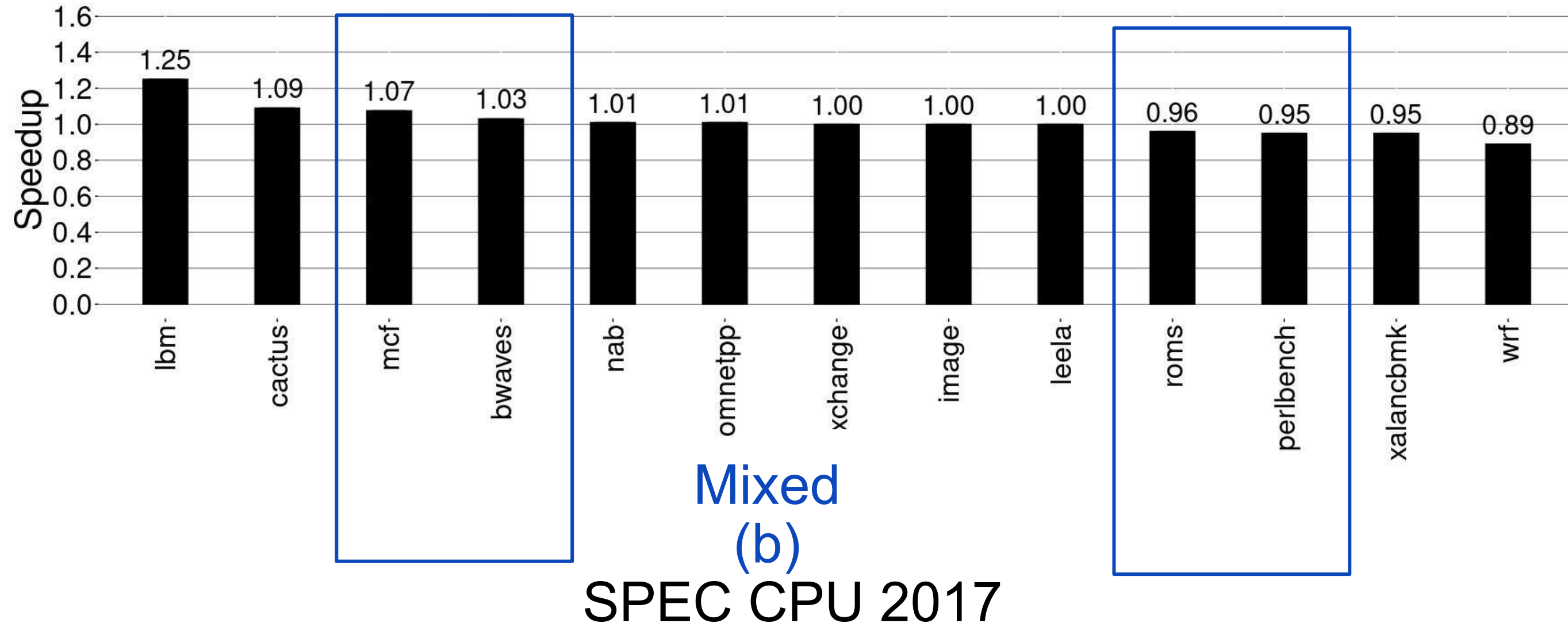
SPEC CPU 2017

Resultados - *Multiple applications*

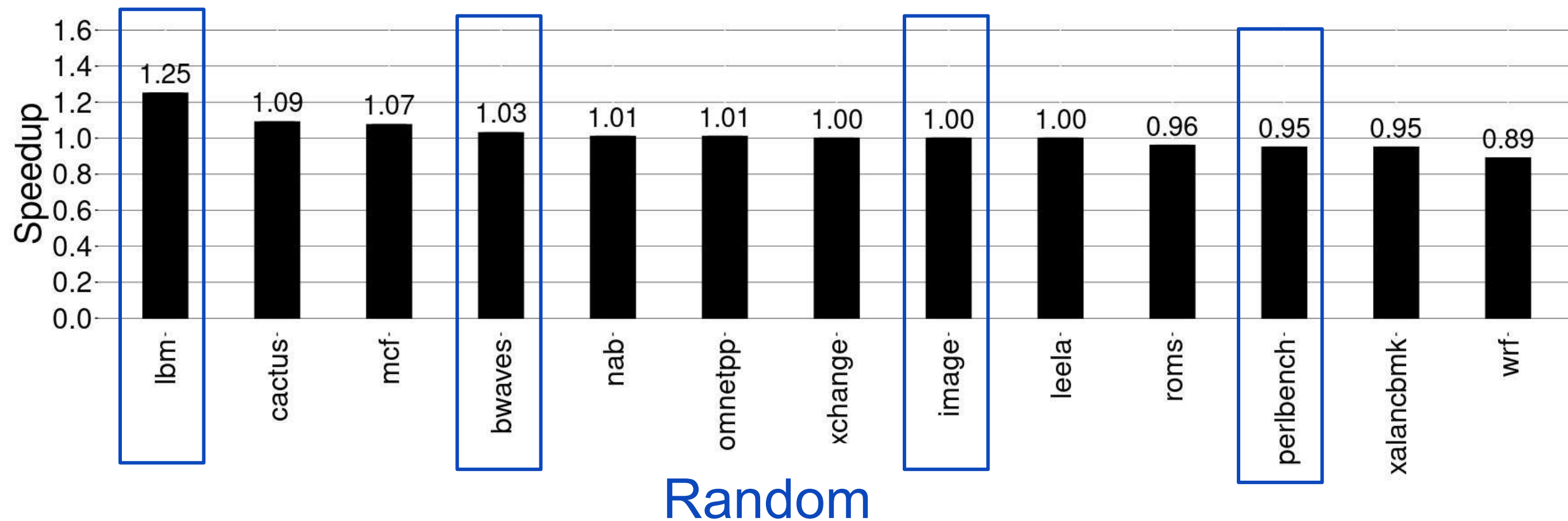


Mixed
(a)
SPEC CPU 2017

Resultados - *Multiple applications*

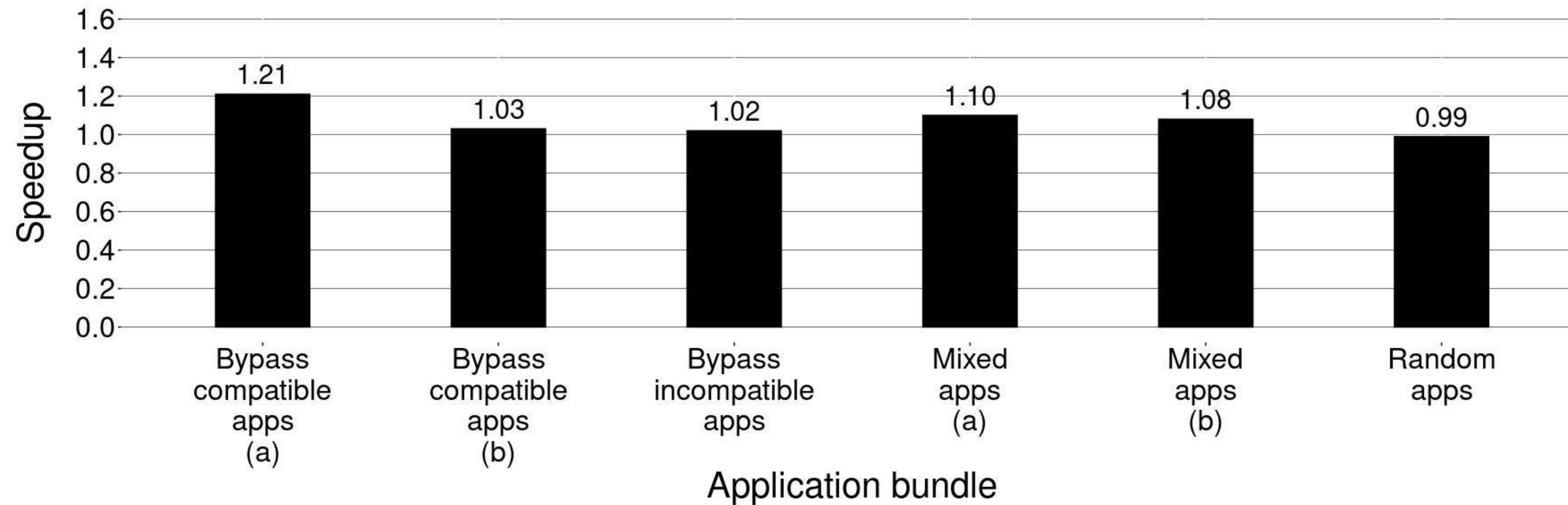


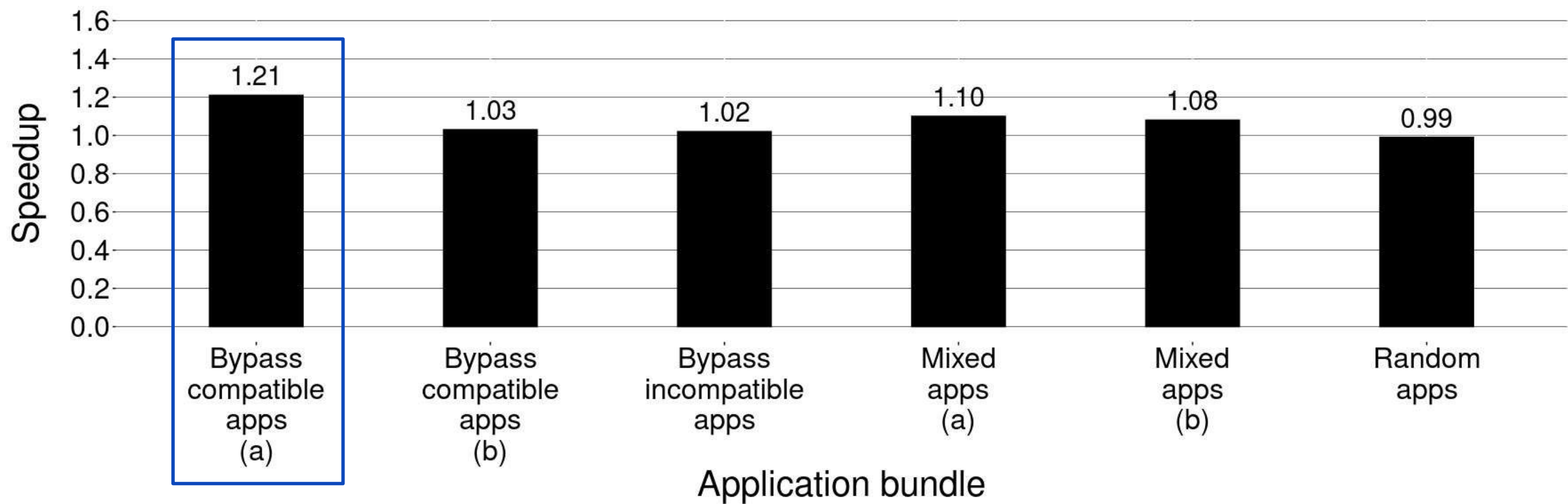
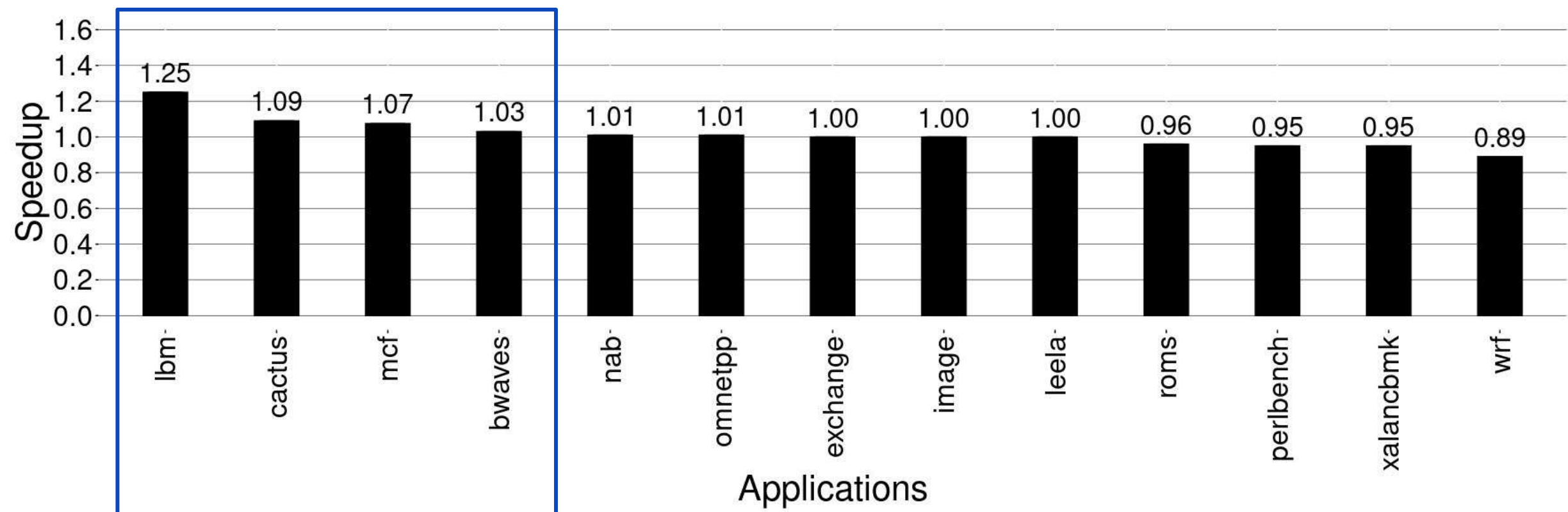
Resultados - *Multiple applications*

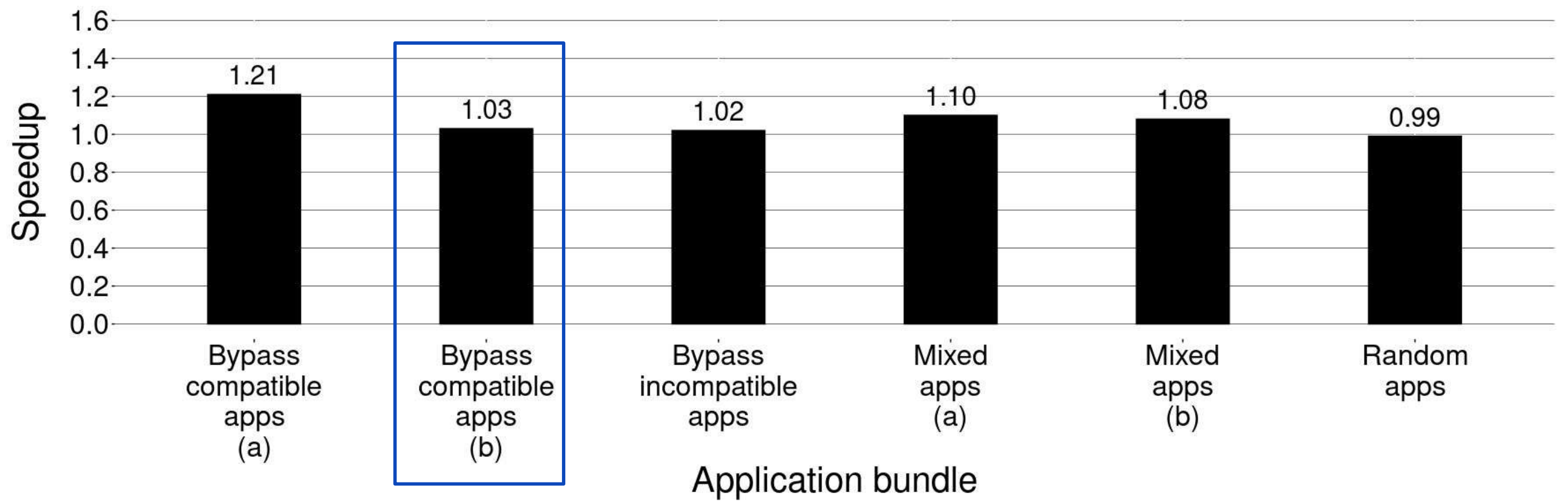
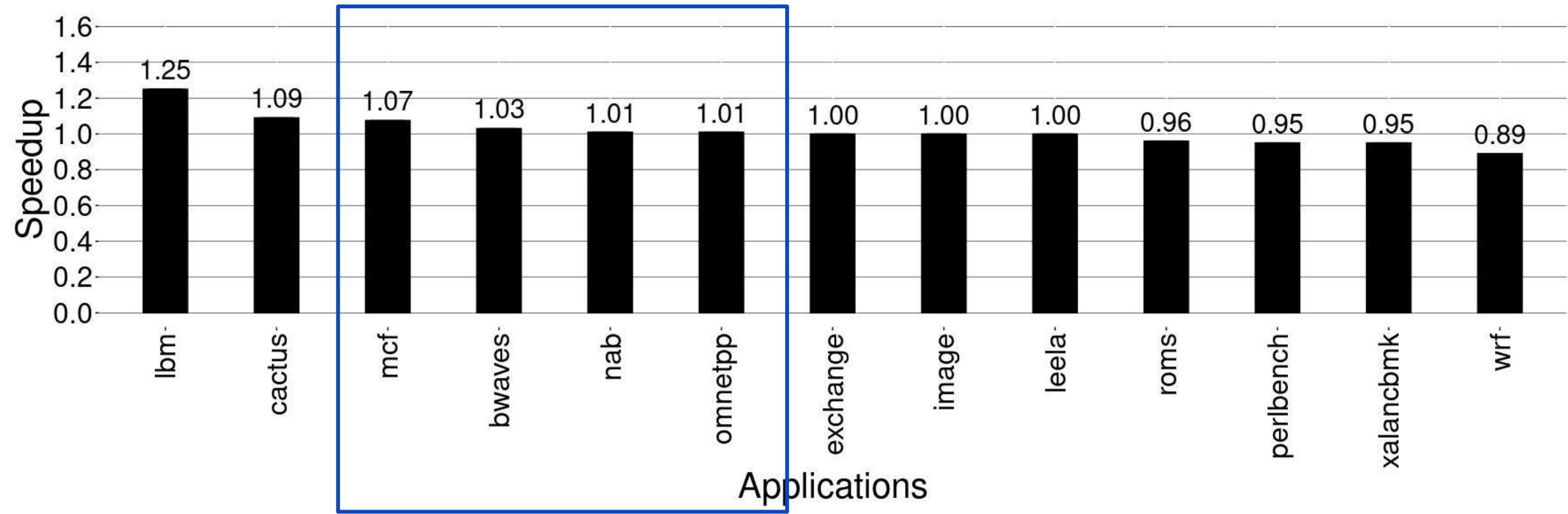


SPEC CPU 2017

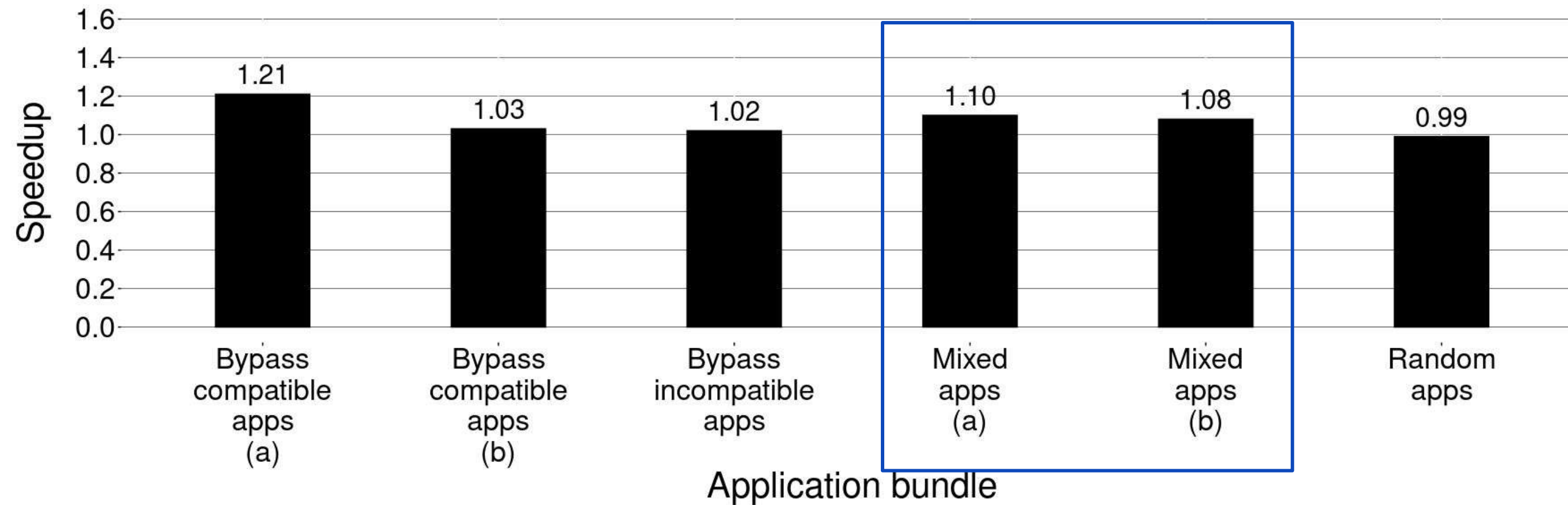
Resultados - *Multiple applications*



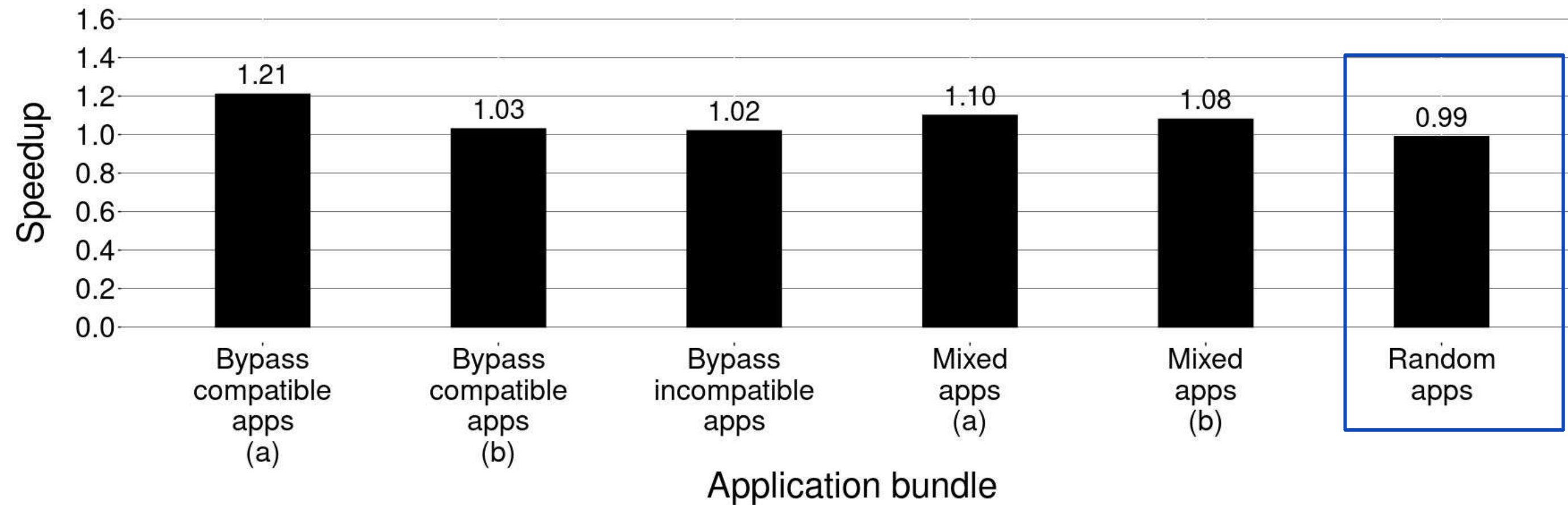




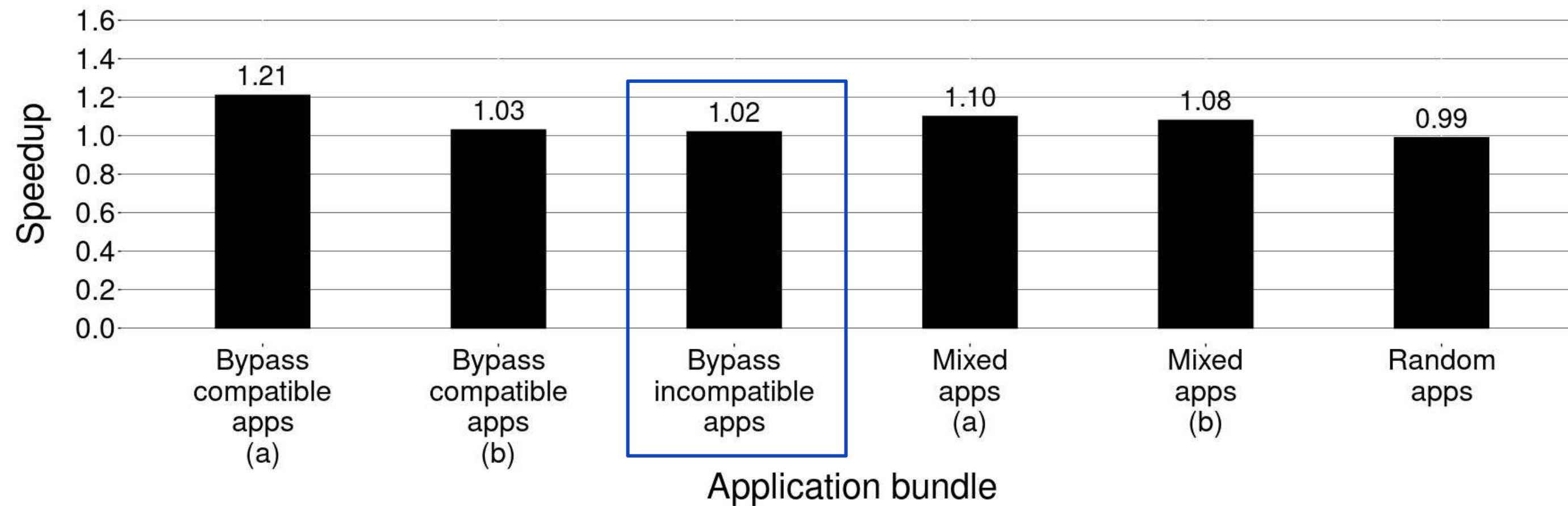
Resultados - *Multiple applications*



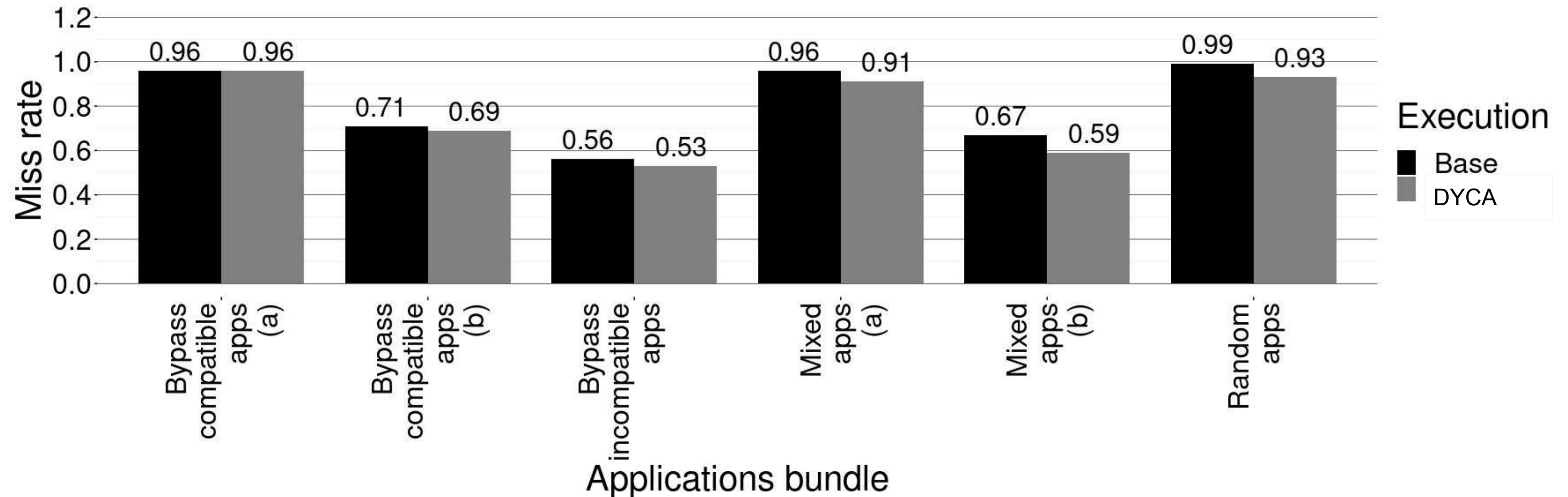
Resultados - *Multiple applications*



Resultados - *Multiple applications*



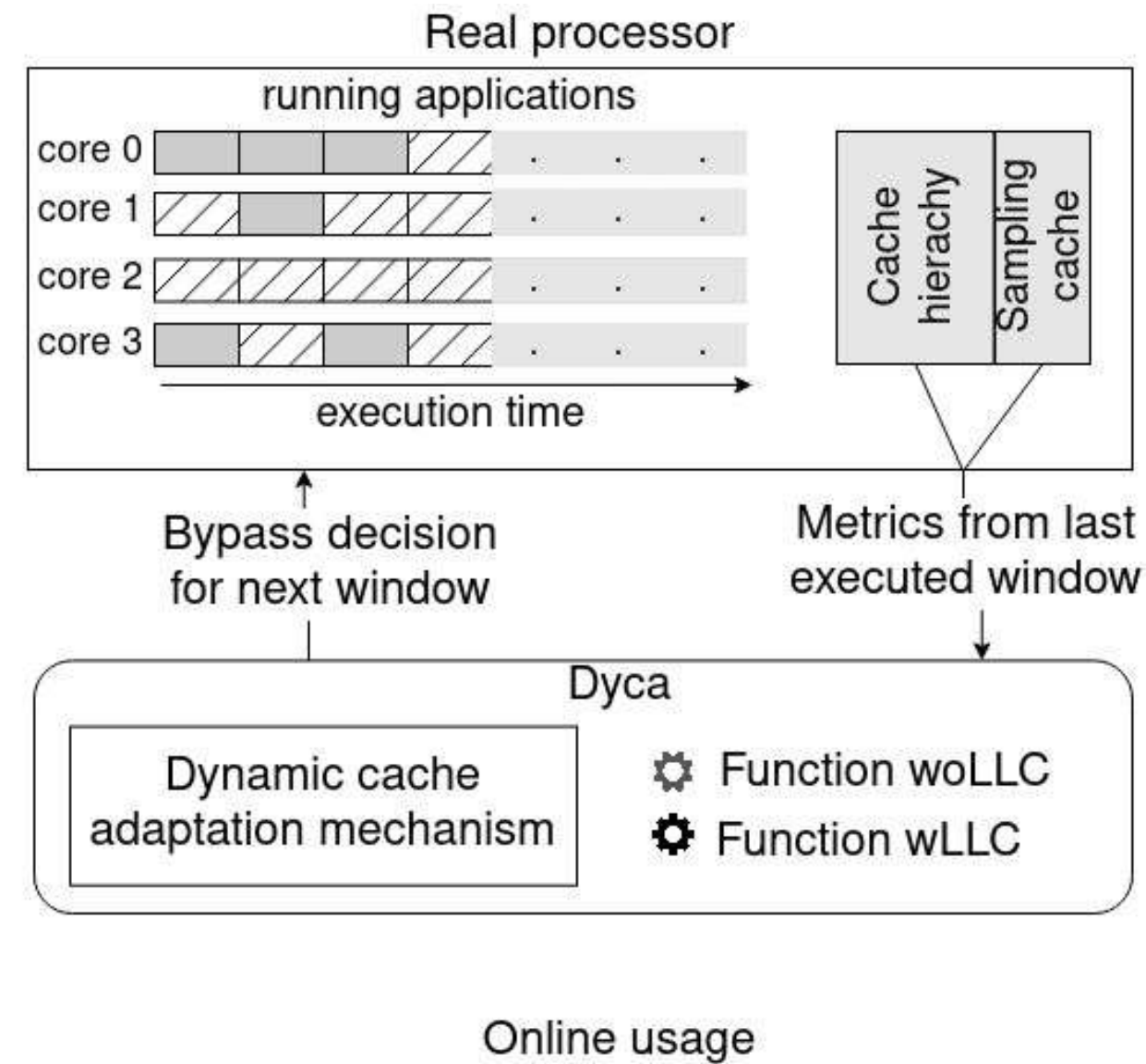
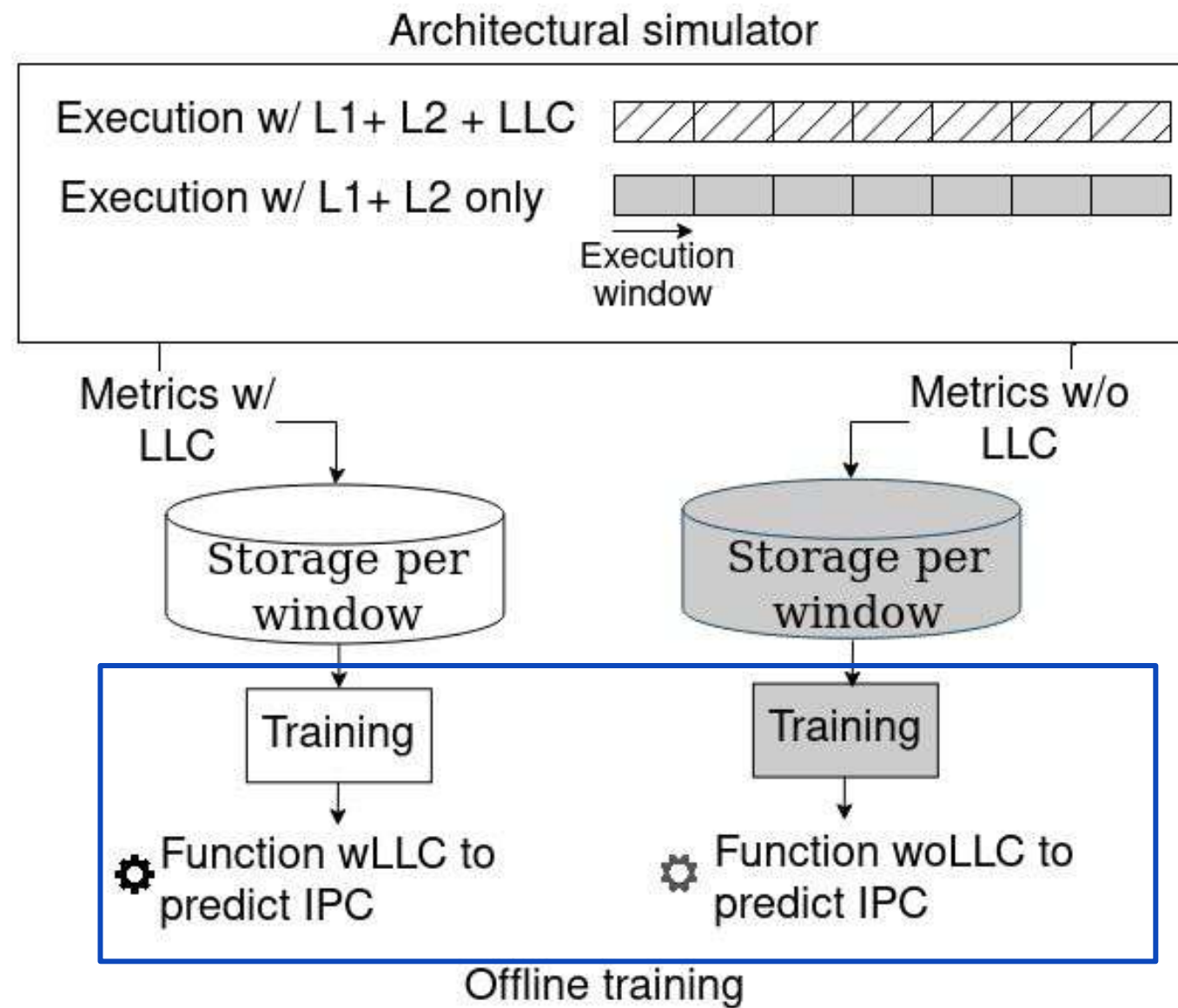
Resultados - *Multiple applications*



Conclusões

- É possível ganhar desempenho usando um mecanismo dinâmico de adaptação do uso da LLC
- O mecanismo proposto apresenta ganhos de desempenho na maior parte dos casos
 - Identificou as janelas onde é possível adaptar o uso da LLC
 - Possui baixo *overhead* de *hardware* - 8kB + 8kB por núcleo
 - Mostrou ganhos em diferentes cenários (*Bypass compatible*, *bypass incompatible* e *Mixed*)
 - Dependente da acurácia do modelo

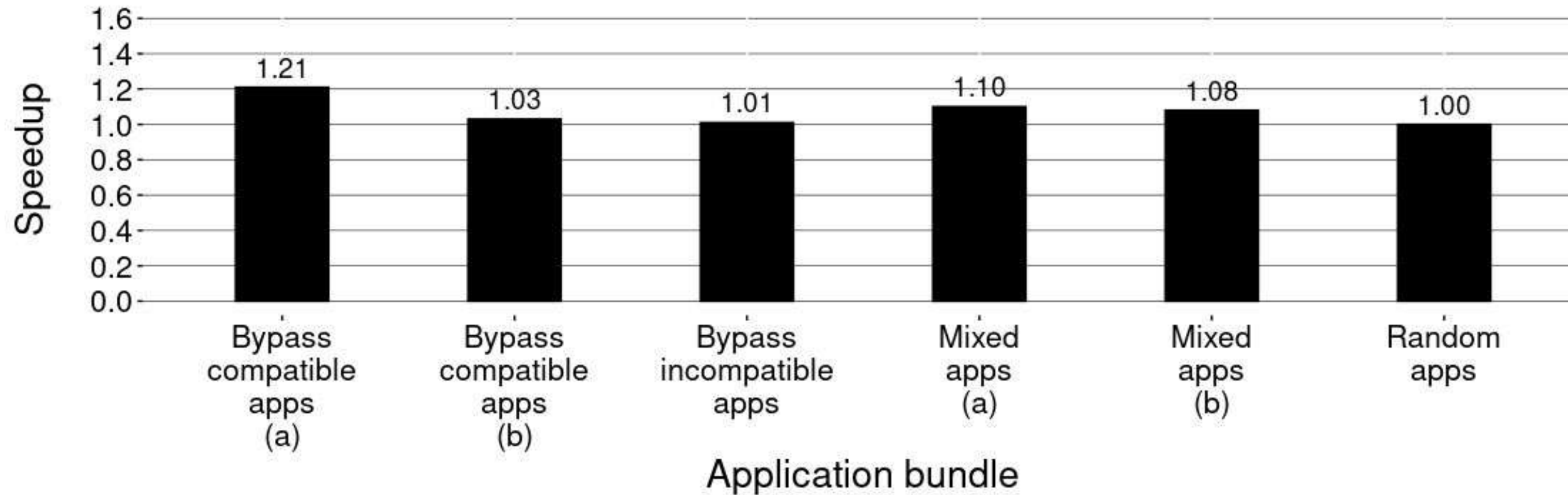
Outras abordagens



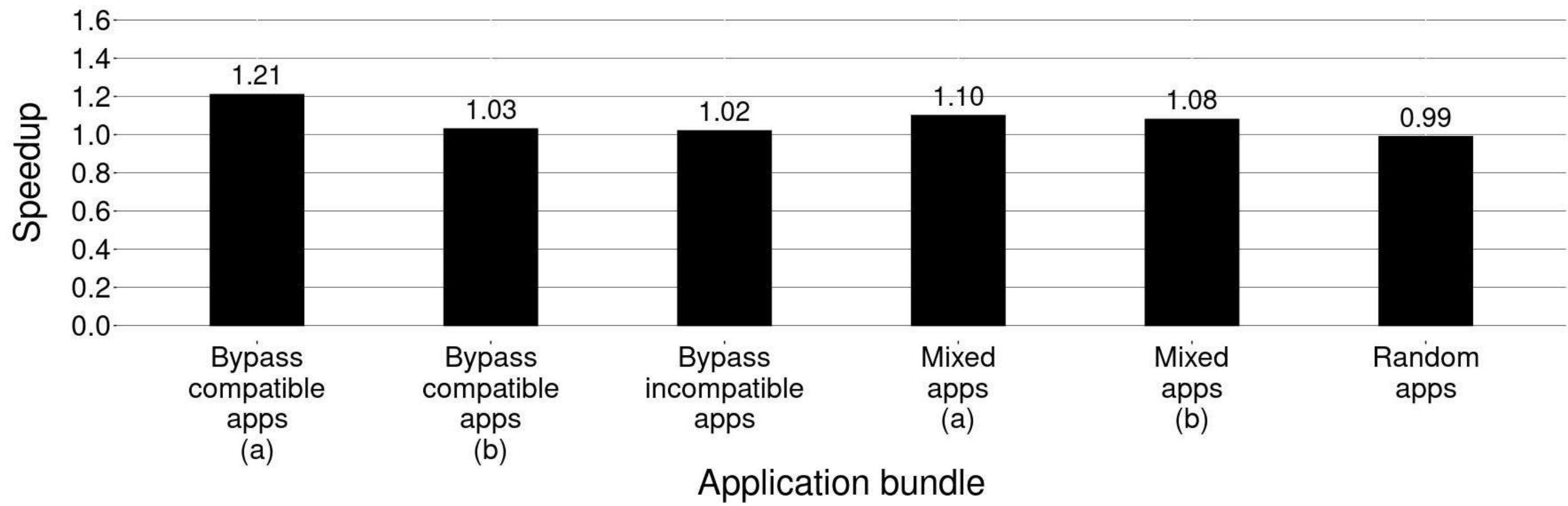
Aprendizado de máquinas

- Testes com Support Vector Machine (SVM), Naive Bayes (NB), Perceptron, Linear Discriminant Analysis (LDA), Classification and Regression Tree (CART) e **Random Forest (RF)**
 - n_estimators igual a 200
- Uso do SULOVA e Recursive XGBoost para seleção das *features*

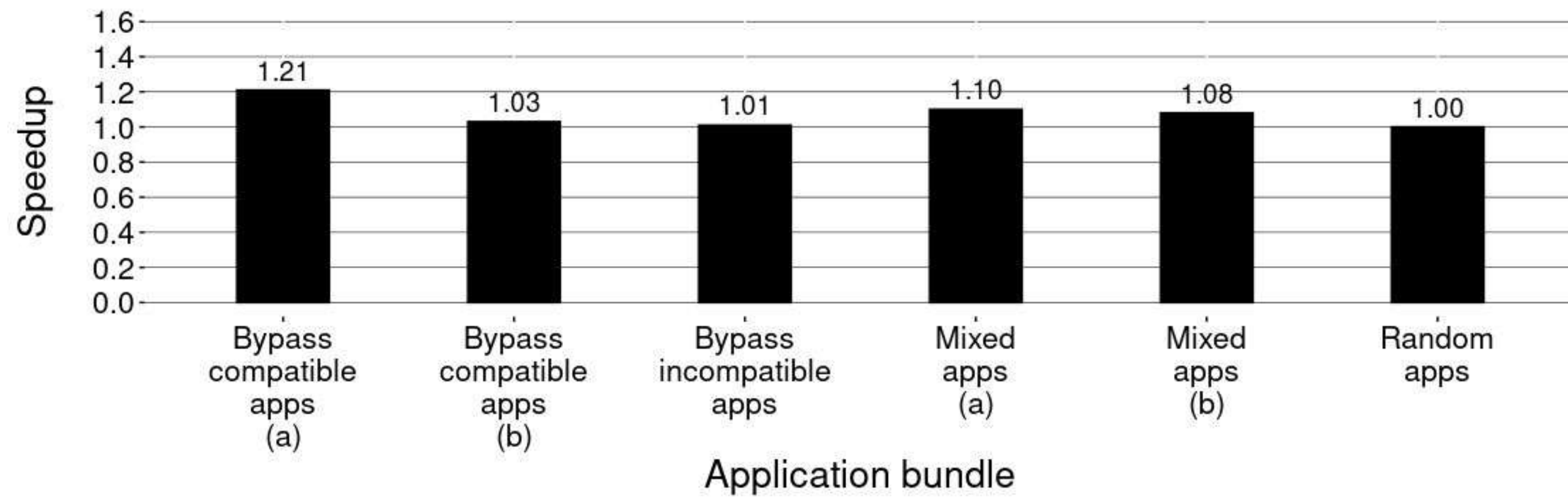
Resultado *multiple applications*



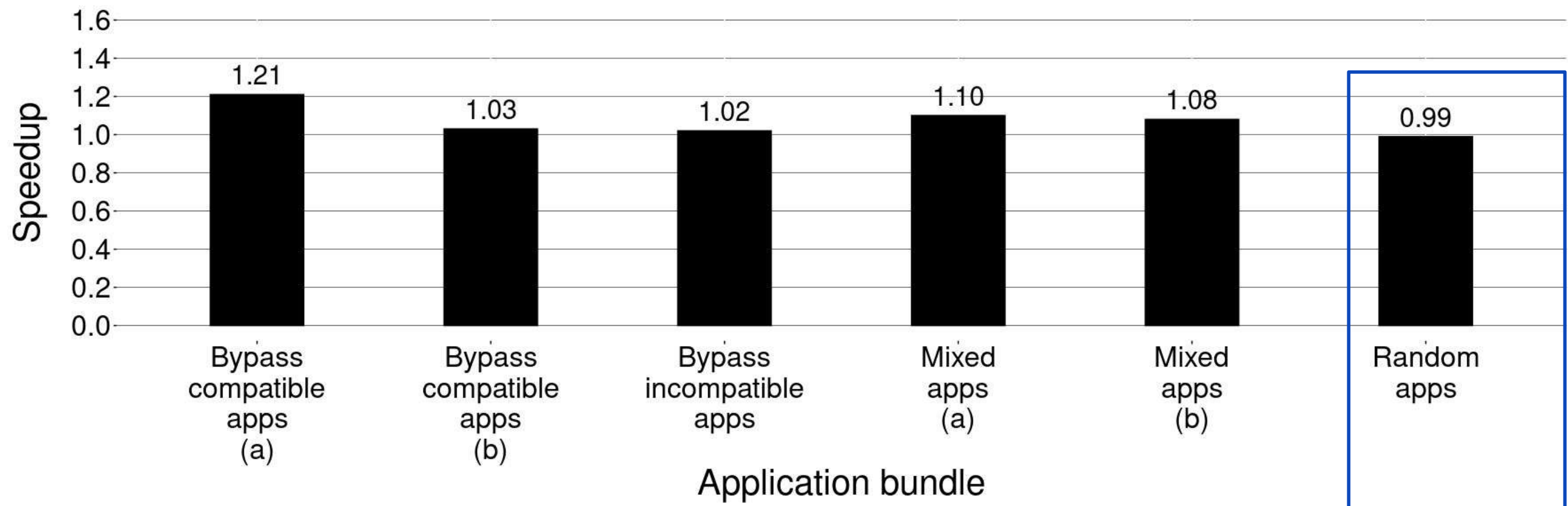
Regressão



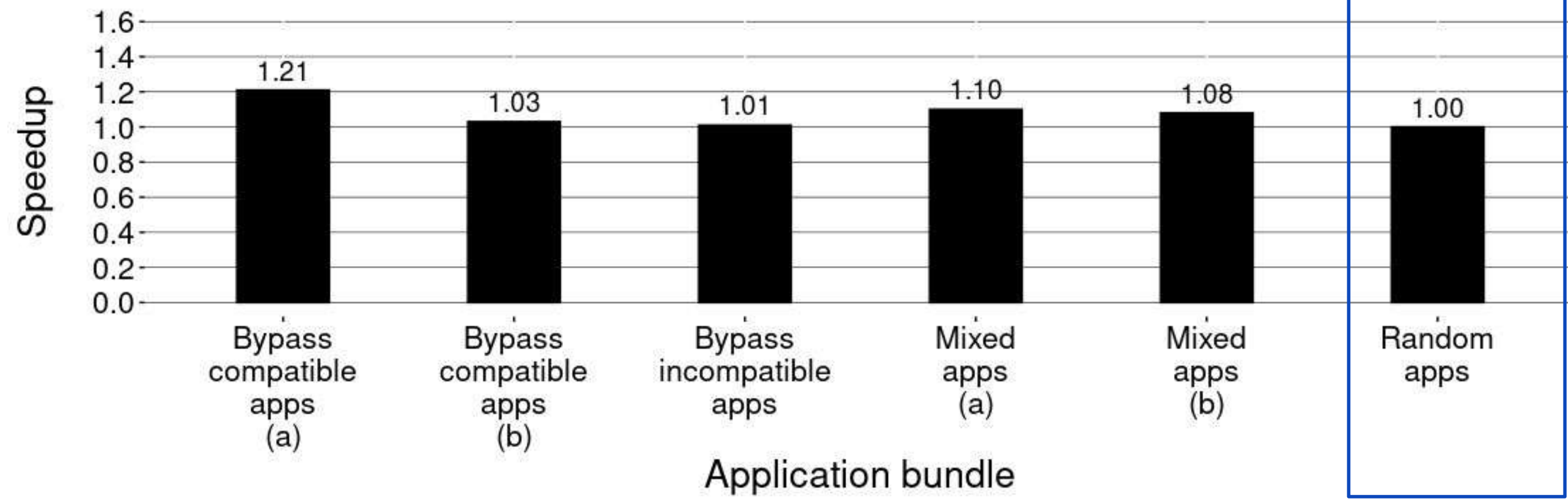
ML



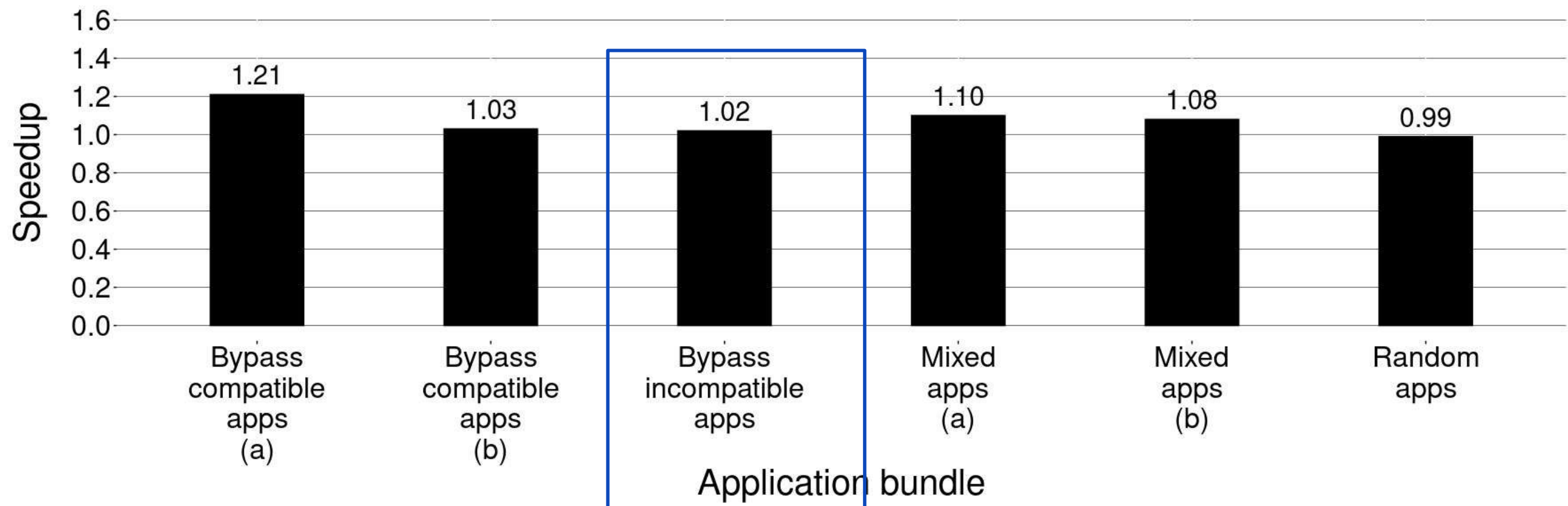
Regressão



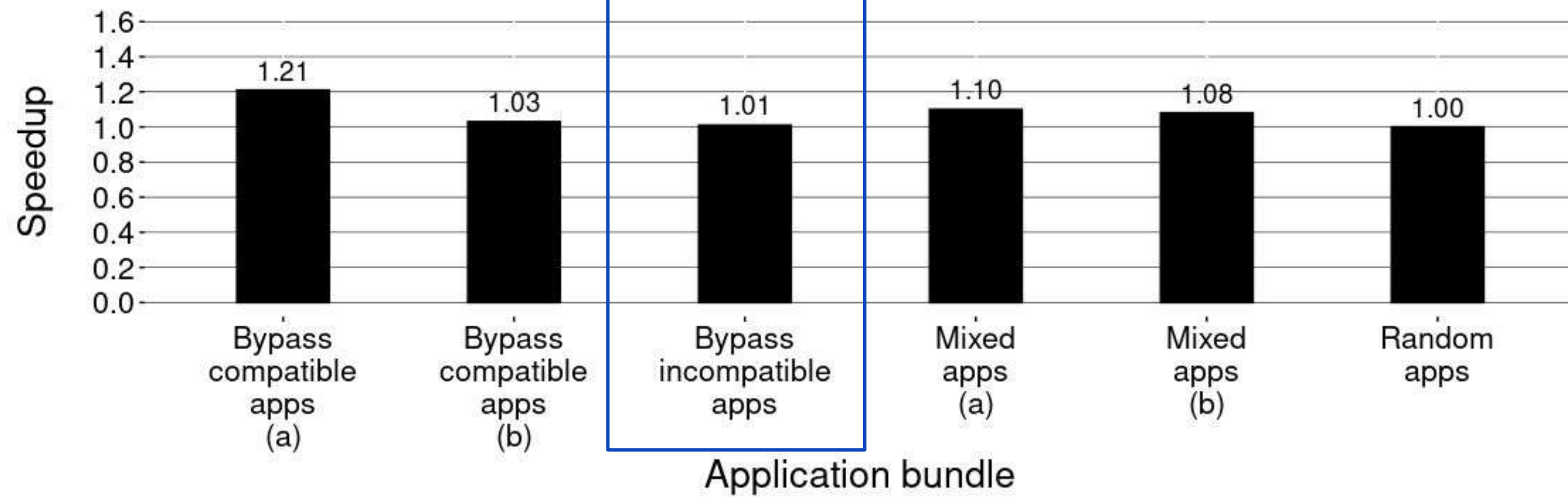
ML



Regressão



ML



Trabalhos futuros

- Análise de possíveis economias de consumo de energia
 - Bypass associado a técnicas de desligamento da LLC
 - *Gated-Vdd*
- Experimentos com diferentes configurações da LLC
- Ir além do SPEC CPU e aumentar o número de aplicações
- Aprofundar a discussão sobre a implementação em um sistema real
- Variar o tamanho da janela de execução

Artigos relacionados à dissertação

- Carmin, M., Santos, P.C. and Alves, M.A.Z. (2022). DyCa: Dynamically Adaptable Cache Bypassing Mechanism. In *Anais do XXIII Simpósio em Sistemas Computacionais de Alto Desempenho* (pp. 109-120). SBC.
- Carmin, M., Moreira, F.B. and Alves, M.A.Z. (2022). Mecanismo oráculo dinâmico para predição do uso da LLC. In *Anais da XXII Escola Regional de Alto Desempenho da Região Sul* (pp. 103-104). SBC.
- Carmin, M., Ensina, L.A. and Alves, M.A.Z. (2022). Comparison of Different Adaptable Cache Bypassing Approaches. In *2022 XII Brazilian symposium on computing systems engineering (SBESC)*. IEEE.

Projetos paralelos

- **Participação em artigos:**
 - Clementino, D., Carmin, M., Ditter, A. and Alves, M., 2021, April. Proposta de Mecanismo para Replicação de Comportamento de Nuvem para Pesquisa e Reprodutibilidade. In *Anais da XXI Escola Regional de Alto Desempenho da Região Sul* (pp. 13-16). SBC.
- **Projeto WHPC**
 - Palestra ERAD - Apresentando o Women in HPC Affiliate Brasil Região Sul
- **Parcerias com o projeto Brazilians in Tech**
- **Minicursos ministrados:**
 - Meninas nas exatas - Descobrimo a verdade por trás do DNA dos dinossauros e outras habilidades dentro da Informática Biomédica”
 - WSCAD - “From sequence assembly to ancestry testing: HPC challenges for bioinformatics”
- **Apresentação em eventos:**
 - II Fórum dos Programas de Pós-Graduação em Computação do Paraná - Adaptação dinâmica de cache em arquiteturas heterogêneas baseada no perfil de aplicações.

Projetos paralelos

- Textos revista SBC Horizontes:
 - CARMIN, M.; ANJOS, M. . Por que não me sinto suficiente no mundo da tecnologia? Uma discussão psicológica sobre a autoexigência feminina.. Horizontes SBC, 29 set. 2022.
 - CARMIN, M.. Conheça o Women in HPC Affiliate Região Sul do Brasil e sua participação no ERAD/RS 2022. Horizontes SBC, 02 maio 2022.
 - CARMIN, M.. Afghan Dreamers ? Como é a realidade das meninas afegãs na computação. SBC Horizontes, Curitiba, 15 set. 2021.
 - CARMIN, M.; RIBEIRO, K. . Por que as mulheres acadêmicas estão produzindo menos durante a quarentena?. SBC Horizontes, Curitiba, 14 maio 2020.
 - CARMIN, M.. Um pequeno recorte do caminho para a vacina: mulheres fundamentais para o desenvolvimento de um imunizante em tempo recorde. SBC Horizontes, Curitiba, 16 jan. 2020.
- Bancas de Trabalho de graduação:
 - Participação em banca de Odair Mario Ditzkun Junior. Modelo de Escalonamento de Tarefas para Gerenciadores de Bancos de Dados em Arquitetura Numa. 2021.
 - Participação em banca de Cláudio Torres Júnios. Desafios na correção de sequências genômicas. 2022.
 - Participação em banca de Eduarda Moresco Farias. Modelando a admissão de pacientes à leitos hospitalares. 2022.
 - Banca Avaliadora de Projetos - Disciplina Inovação Tecnológica e Gestão de Projetos (CI1204).

DYCA: Multi-core Dynamically Adaptable Cache Bypassing Mechanism

Mariana Carmin

Orientador: Marco Antonio Zanata Alves

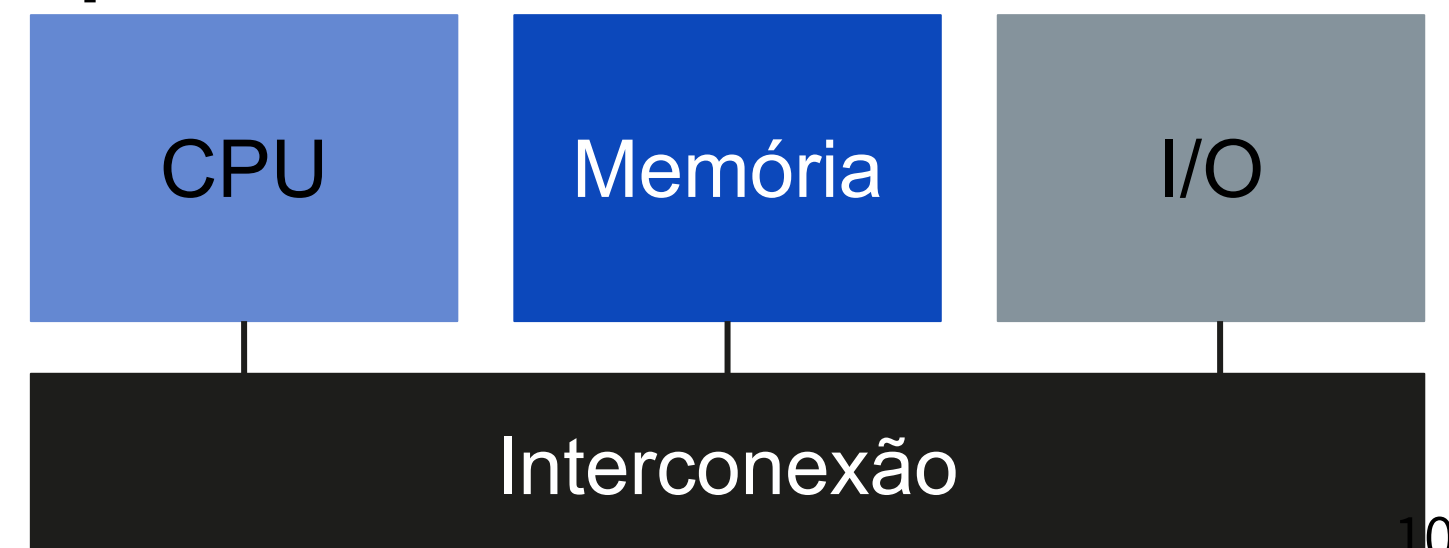


serrapilheira



Arquitetura de Von Neumann

- Criada em 1946
- Define unidades de armazenamento e processamento como unidades dissociadas
- Problema inerente:
 - Transferência de dados entre o processador e a memória



Memory wall

- Limitação do ganho de desempenho associado à latência de obtenção de dados
- Problema se intensificou ao longo dos anos
 - 2004 à 2011:
 - Desempenho processadores: 4.6x
 - Redução latência memórias: 1.3x

Aprendizado de máquina

- **Features selecionadas:**
 - Taxa de acessos na L2
 - Taxa de hit na L2
 - HPKI L2
 - Número de writes na L1

DYCA - Modelo de regressão

- Preparando a base:
 - União de ambas as execuções (com e sem LLC)
 - Deslocamento do IPC

```
application, core, IPC_LLC, IPC_SLLC, miss_llc, hit_llc, miss_sampling_cache, hit_sampling_cache
BWAVES, 0, 3.0167530, 3.2490450, 0.008434, 0.991566, 0.000573, 0.999427
BWAVES, 0, 3.4028880, 3.4396540, 0.004816, 0.995184, 0.001621, 0.998379
CACTUS, 0, 0.8727450, 0.9431940, 0.170157, 0.829843, 0.003642, 0.996358
CACTUS, 0, 0.8735980, 0.9434470, 0.170158, 0.829842, 0.996204, 0.003796
```