

## 性能評価プログラムの開発技術：リバーストレータ

坂本, 真理子  
九州大学大学院システム情報科学府

<http://hdl.handle.net/2324/9171>

---

出版情報：SLRC プレゼンテーション, 2007-07-25  
バージョン：  
権利関係：



---

[Next Generation Architecture Forum 2007 ]

# 性能評価プログラムの開発技術 リバーストレーサ

2007.07.25

国立大学法人九州大学  
大学院システム情報科学府  
坂本 真理子

---

# 内容

---

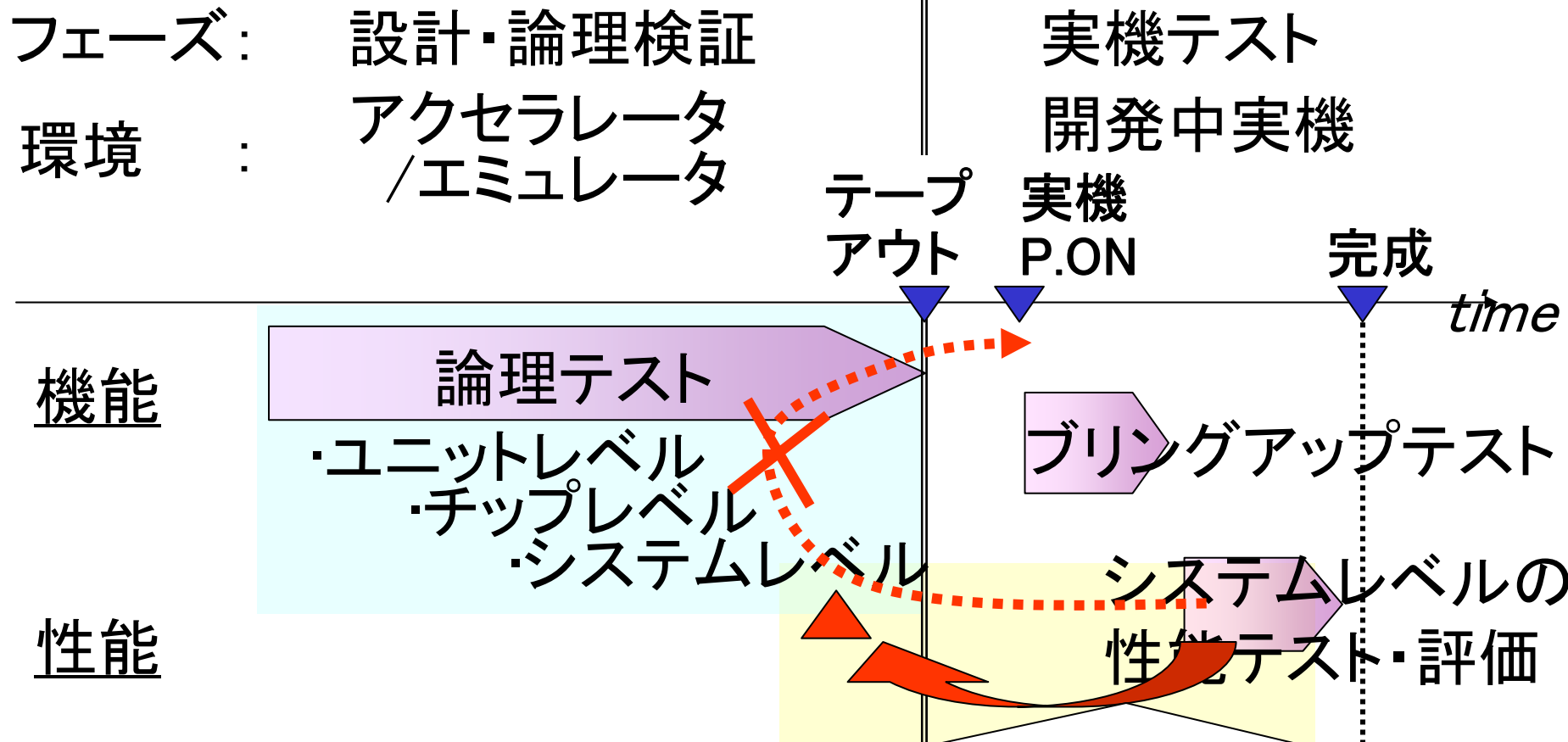
- 背景
- ゴール
- アプローチ
- 性能テストプログラム (PTP)
- 評価
- まとめ

# 背景

---

- 高性能プロセッサとメモリシステムの開発
- 取り巻く環境
  - オープン化による市場競争の激化で、開発期間が短縮
  - LSIチップに搭載可能なトランジスタ数の増加による、制御系の設計論理の複雑度が拡大
- 限られた期間、複雑なシステム、性能目標を達成
  - 早期に、システムレベルで、高い精度の性能テストを実施し、設計へのフィードバックを掛けたいが、
  - 実機が完成するまでできない

# 性能を早期に把握するメリット



テープアウト前にテストを行うメリットは大きい。  
変更のコストが安く、自由度が高い

しかし、早期テスト用の性能テストプログラムがない！

# なぜ、早期のテスト用プログラムが無い？

## ・・ 実行環境 ・・

- 早期性能テストに、アクセラレータ/エミュレータを使用
  - 完成後実機と同じ論理で組み立てられている(精度OK)
- 実機とアクセラレータ/エミュレータには規模に差がある

(例)

	実機	⇔	アクセラレータ
実行速度	2GHz	1/480	4.2MHz
メモリサイズ	64GB	1/256	256MB
周辺機器	有り		無し
実OS	稼動		非稼動

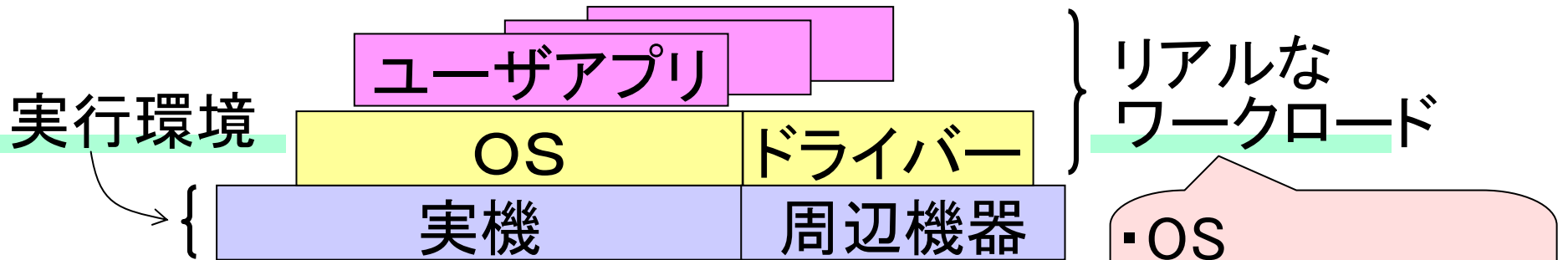
実機での性能テストは、OSとユーザアプリで構成する実ワークロード(リアル)で実施.

しかし、アクセラレータ/エミュレータの規模では、リアルの実行はムリ

# なぜ、早期テスト用プログラムが無い？

.. ワークロード ..

ユーザ環境



- 早期テスト時に実機は存在しない
- アクセラレータ/エミュレータに、OSを乗せられない

そのため、リアリスティックなワークロードの作成

- 命令の混在，並び，非同期割り込みのタイミング
- メモリアクセスの時間的・空間的パターン

解析に基づく作成は難しい

# ゴール

---

テープアウトの前の高精度性能評価に用いる,  
性能テストプログラムの作成技術の確立

以降, 性能テストプログラムをPTPとよぶ



PTPの要件:

- リアリスティックである
- アクセラレータ／エミュレータで動く
- 実機完成後の性能を, 高い精度に予測できる



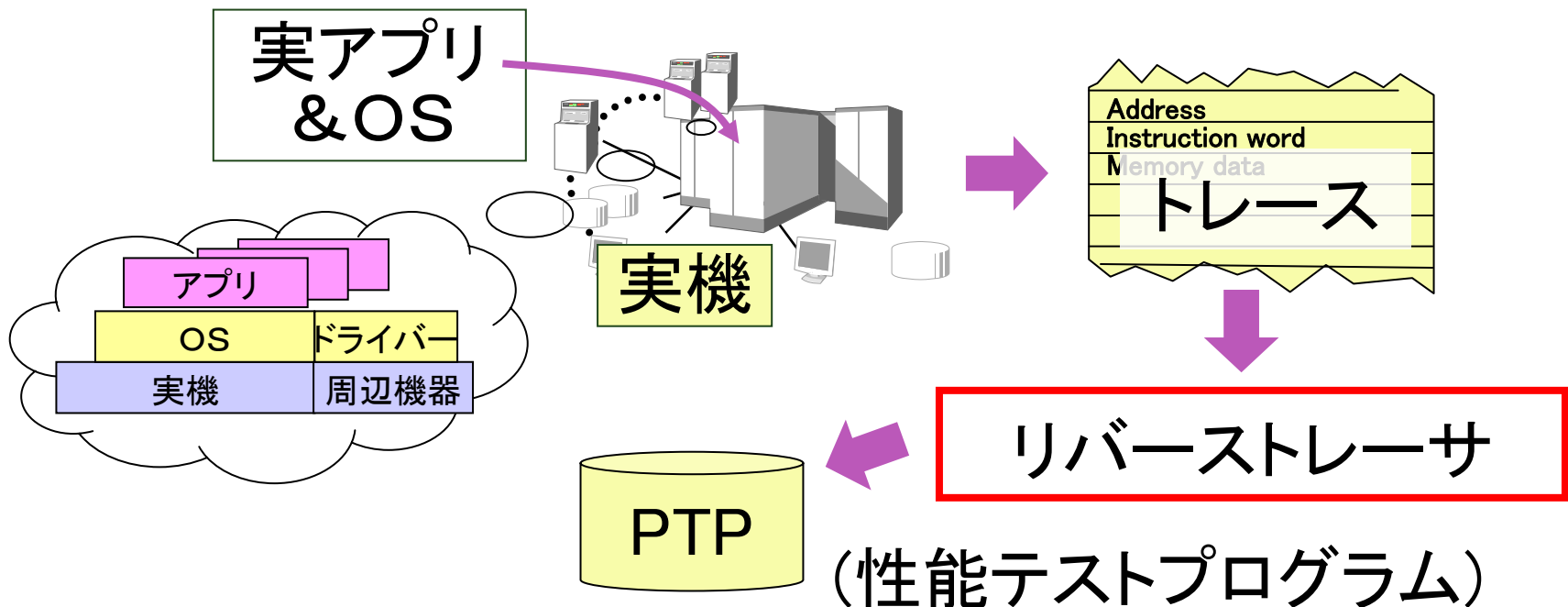
# アプローチ

---

- リアリスティックさの確保
- 規模の差の克服 と 高精度性能予測
  - ・ 実行速度, ハードウェア資源
  - ・ ウォームアップ

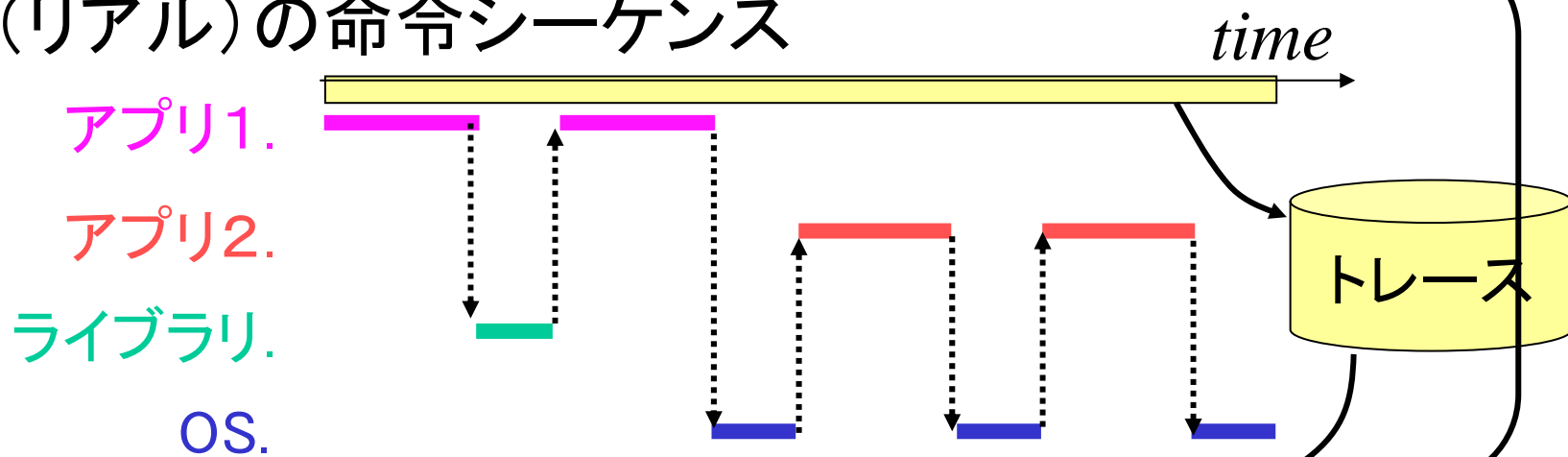
# リアリスティックさの確保

- 実機で作成したトレース(リアル)ベースで, PTP作成
- トレースにある命令シーケンス, メモリのアクセスを忠実に再現することで, リアリスティックさを確保



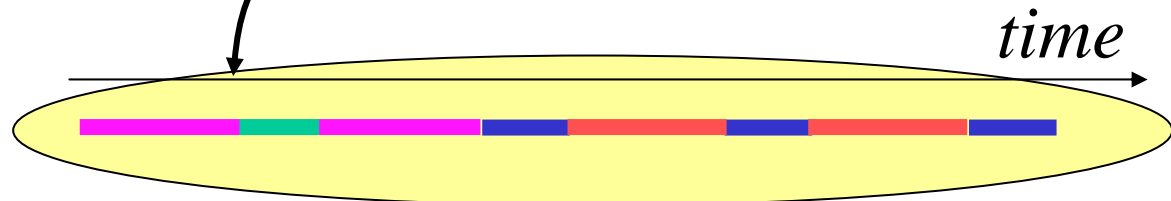
# リアリスティックさの確保

## 実機(リアル)の命令シーケンス



上記の複数の独立した処理からの命令フローを, PTP は忠実に再現しなければならない

## PTPの実行



# リアリスティックさ確保:リバーストレーサの概念

“コンピュータの基本的な振る舞いは, メモリから命令やデータを参照し, 演算を行い, 結果をメモリに書き戻す”

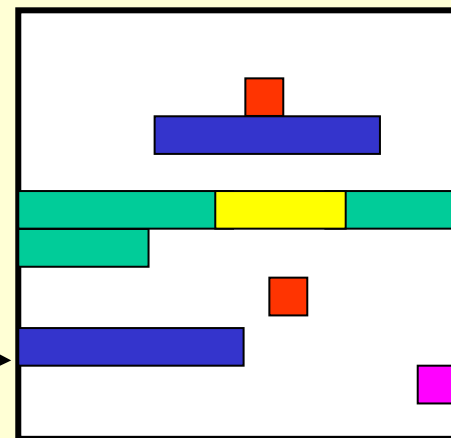
アプリケーション実行

time

- 実行フローを色わけ
- 同じ色で, 時間範囲と参照・更新されたメモリ領域を対応づけ

命令A

メモリ空間



すなわち,

赤いメモリ領域を命令A開始時の状態にし, 命令AのアドレスをPCに設定すれば, 赤の命令フローを再現できる<sup>11</sup>

# 規模の差の克服・高精度性能予測

---

- 規模の差の克服
  - 実行速度 → サンプルングで実行箇所絞る
  - 実行資源 → PTPに必要なメモリデータだけを持つ  
→ 周辺機器による処理を代替する機能
- 高精度性能予測
  - リアルな入力を用いる
  - 命令フロー, メモリアクセスパターンを忠実に再現
  - キャッシュとメモリシステムを暖める(ウォームアップ)
  - サンプルング技術の活用
  - 性能情報採取範囲の厳密化, 等

# 処理の流れ：実機（リアル） vs. PTP

## 実機実行



## RT: リバーストレーサ



RT: 実行可能  
プログラム生成

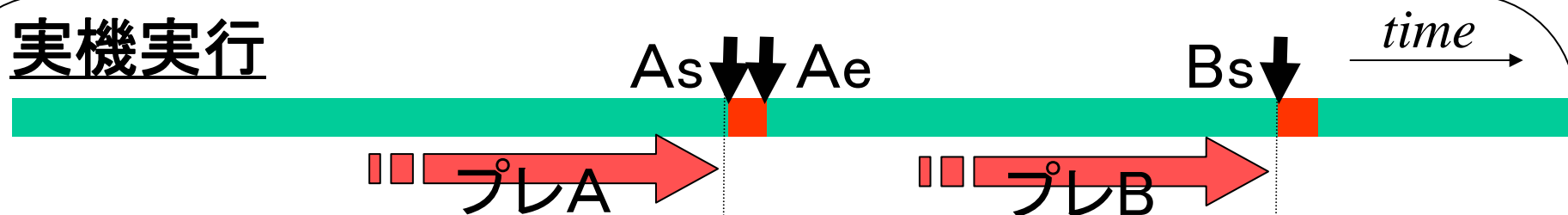
- オリジナルコード
- ウォームアップ処理
- 複数点をつなぐ処理

## 性能テストプログラム(PTP)



# ウォームアップ(1)

## 実機実行



Asでのキャッシュ, メモリシステムのバスやバッファの状態  
⇒ プレAのメモリアクセスの処理に依存  
⇒ As-Ae間の性能は, キャッシュミス率やキャッシュミスのペナルティ(サイクル)で変わる

ミス率やペナルティを整えなければいけない

実機実行とPTPの違いは,

- 実機では, AsやBsは連続した処理フローの1点
- PTPでは, AsとBsはサンプリング箇所先頭で非連続

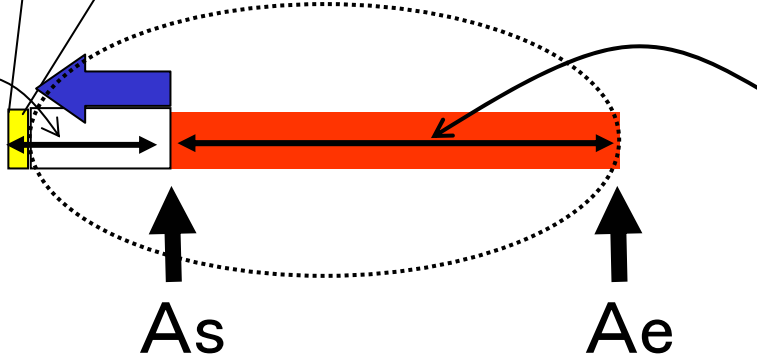
# ウォームアップ(2)



As開始前に実行されたロード命令・ストア命令

- システムのウォームアップのコードをトレースに含める
- 実機では,  $A_s$ の時点のデータキャッシュは, 開始からのロード命令・ストア命令のアクセスを反映
- データキャッシュのウォームアップは, このロード・ストアだけを集めた処理列で実施

ウォームアップ



PTPのターゲット区間の性能情報を採取



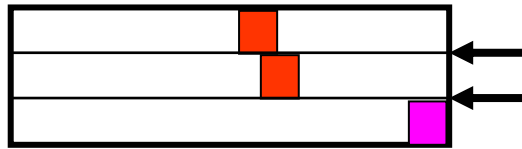
# 必要なメモリデータだけ持つ

アプリケーション実行

時間

処理フローと色で対応づけた、  
参照・更新対象の領域

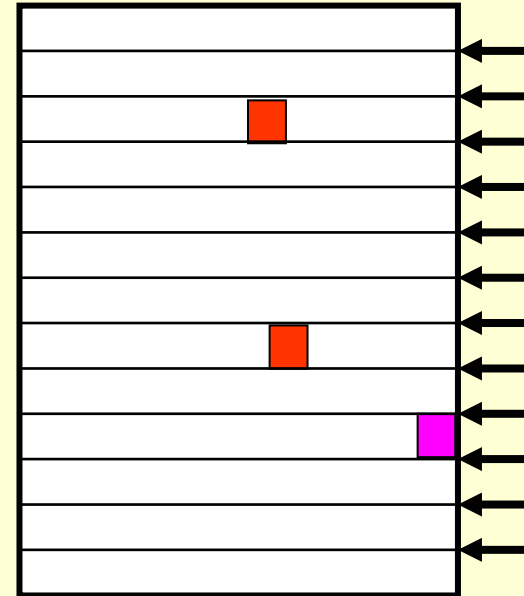
- 論理アドレスを変えないで、メモリサイズの縮小処理
- 縮小にあわせて、アドレス変換テーブルを変更



メモリ空間

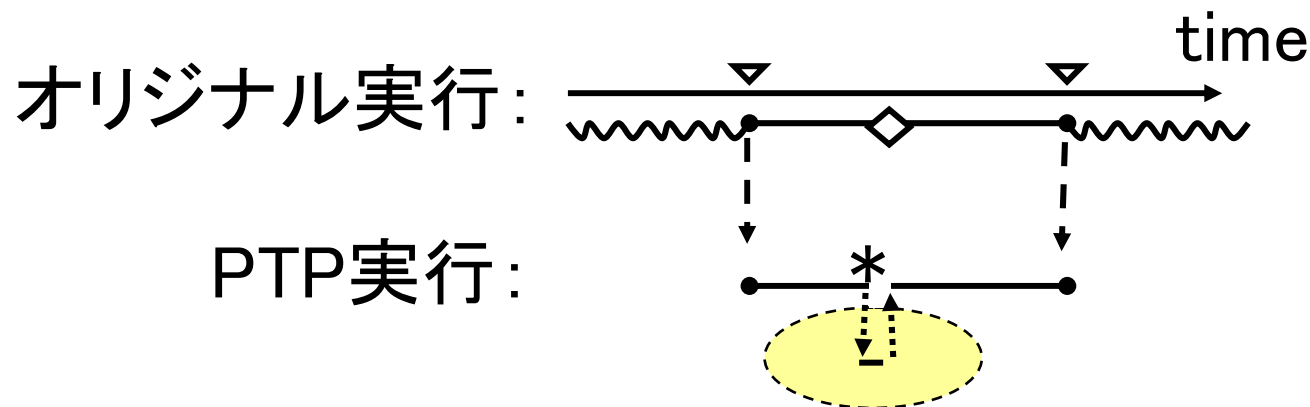
ページ境界

PTPに含まれる処理  
の為のメモリ領域



# 周辺機器の処理の代替機能

## システムコール処理に対する補正処理



- ▽ : レコード書き出しのタイミング
- : オリジナルコード
- ~~~~~ : ウォームアップコード採取区間
- ◇ : システムコール
- \* : 補正処理への分岐命令

# 評価内容

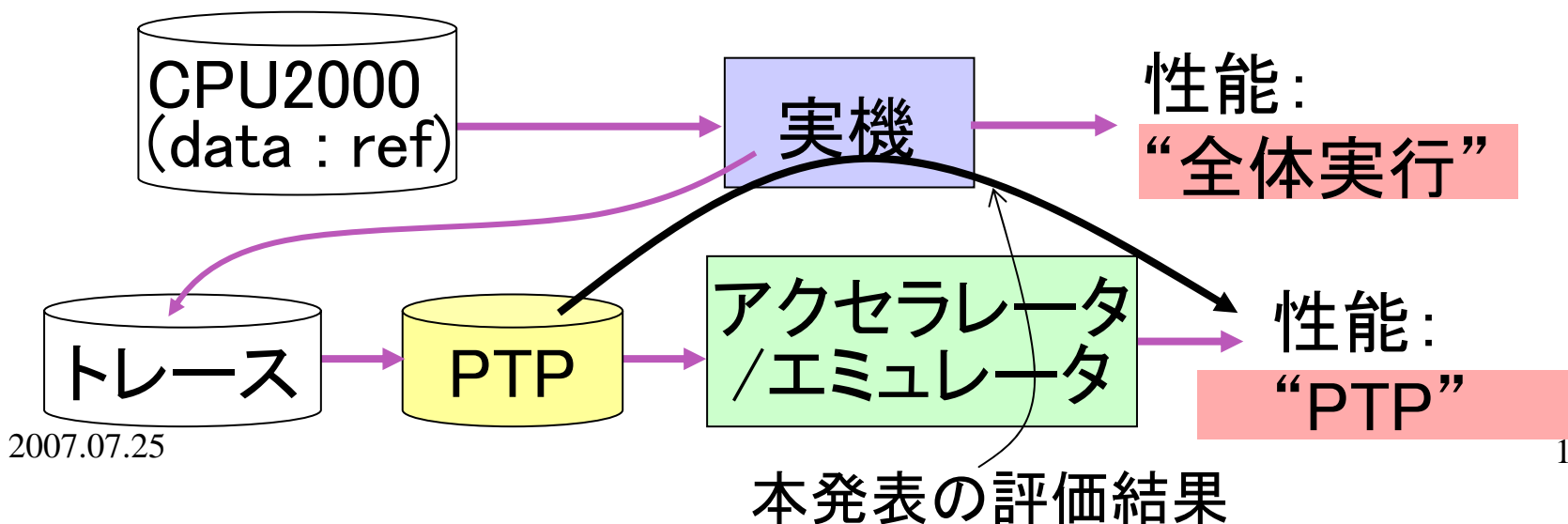
---

- 評価の環境
- 性能
  - 精度
  - 主要なテーブル類のミス率
  - ウォームアップ効果
- 規模について

# 評価環境と比較対象

- 評価環境

- ベンチマーク: SPEC CPU2000 (INT 10本、FP 14本)
- コンパイラ: SUN Studio11
- PRIMEPOWER650(CPU:2.1GHz), UNIX OS Solaris™
- RTバイナリの規模: 実行命令数 60M~100M
- 性能の比較: 全体実行 vs. PTP

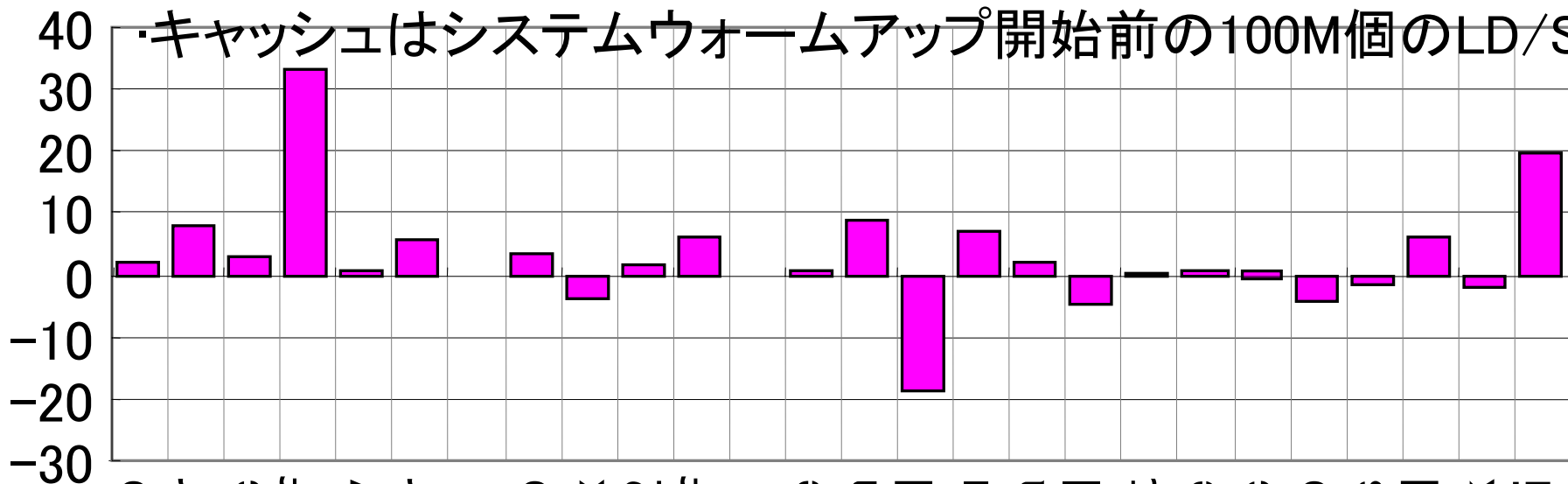


# 性能誤差 (CPI比)

サンプル点あたり10M命令  
ウォームアップ

CPI比 =  $1 - \frac{\text{PTP性能}}{\text{全体実行性能}}$

(%)  
・システムは代表点の直近の2M命令  
・キャッシュはシステムウォームアップ開始前の100M個のLD/ST



サンプル点数

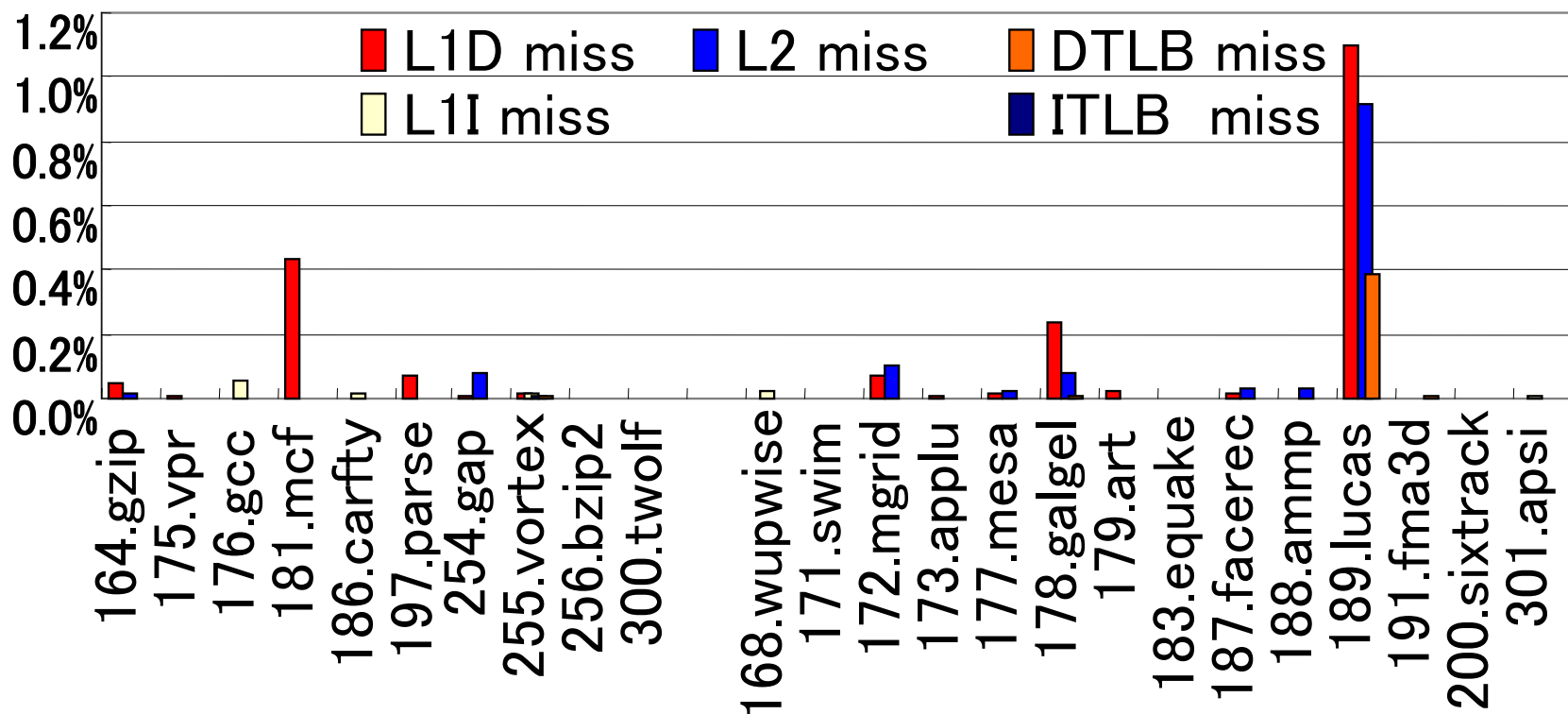
8 8 8 7 7 8 6 8 8 5 8 7 7 7 8 8 8 6 7 5 8 8 8 8

サンプリングは、UCSDのSimPointを利用。入力(BBV)は独自に作成

# 主要なキャッシュのミス率

## 主要なキャッシュのミス率の差 PTP(wup) - 全体実行

1%以下の世界での比表示は  
ミスリードのため、差で表示



L1キャッシュ, L2キャッシュ, TLBのミス率  
189.lucasを除き, ほぼ一致

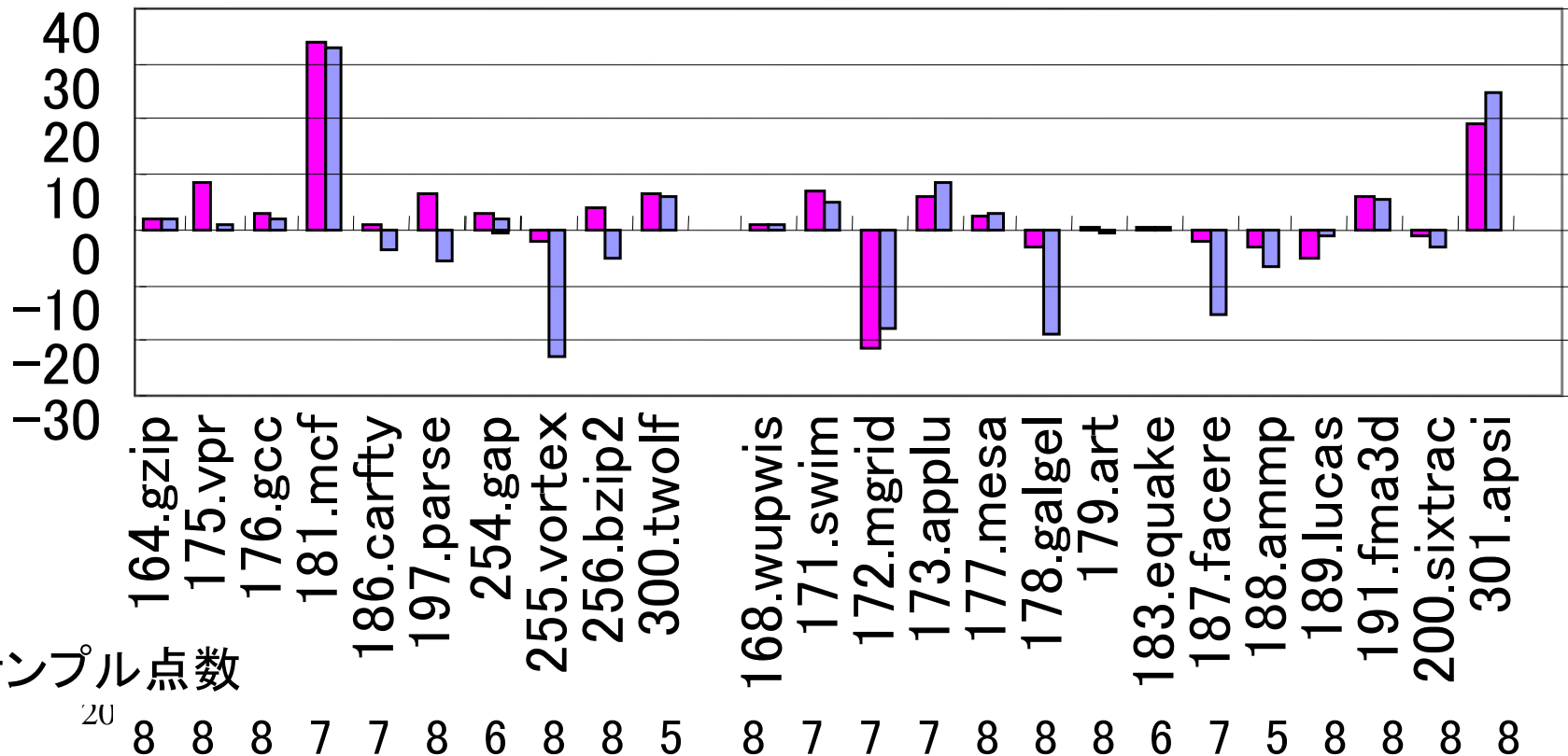
# PTPウォームアップの効果

$$\text{CPI比} = 1 - \frac{\text{PTP性能}}{\text{全体実行性能}}$$

■ PTP(10M命令\*X点)WUPあり  
■ PTP(10M命令\*X点)WUPなし

・システムは代表点の直近の2M命令

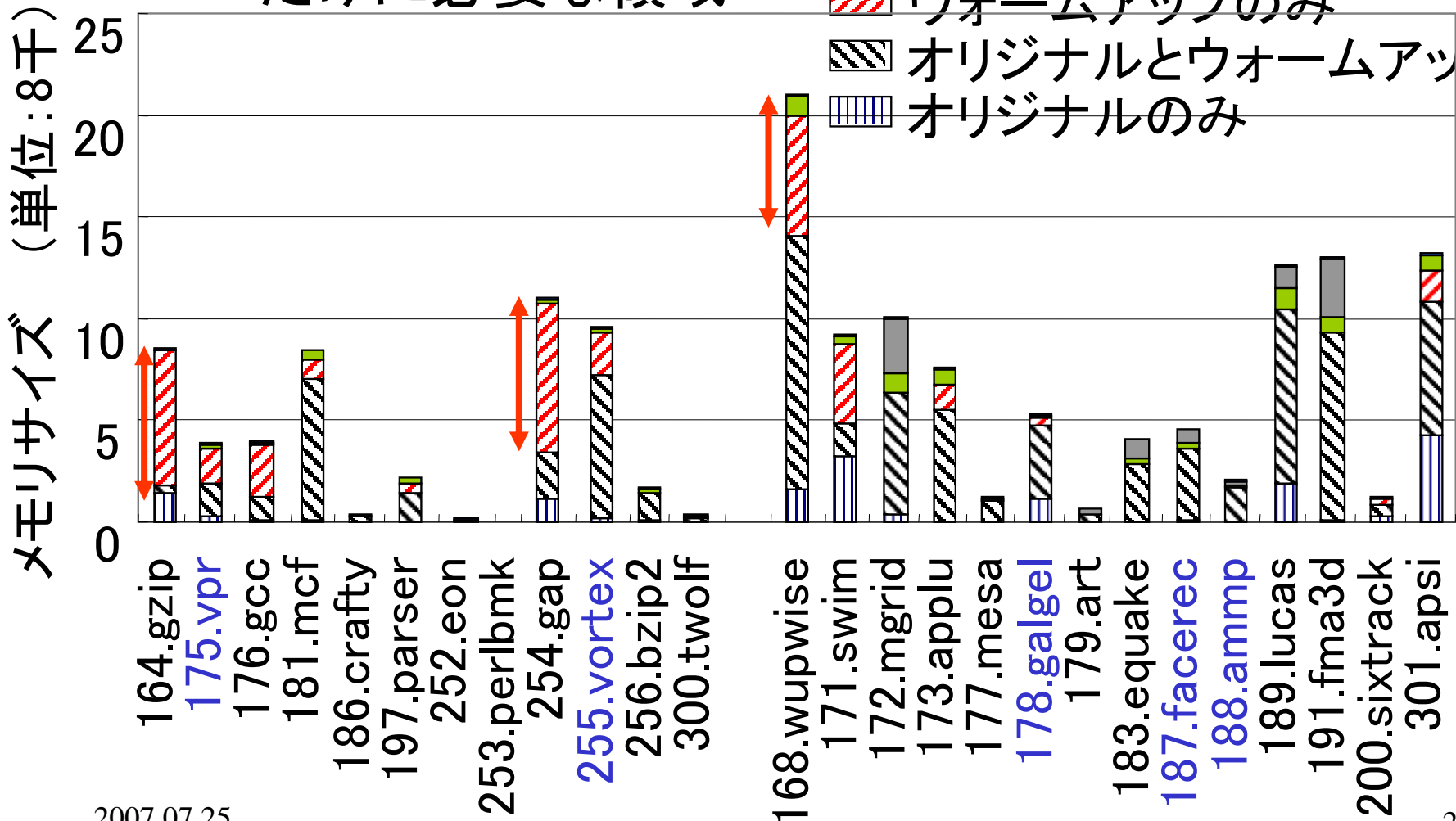
(%) ・キャッシュはシステムウォームアップ開始前の100M個のLD/ST



# PTPメモリ領域の内訳

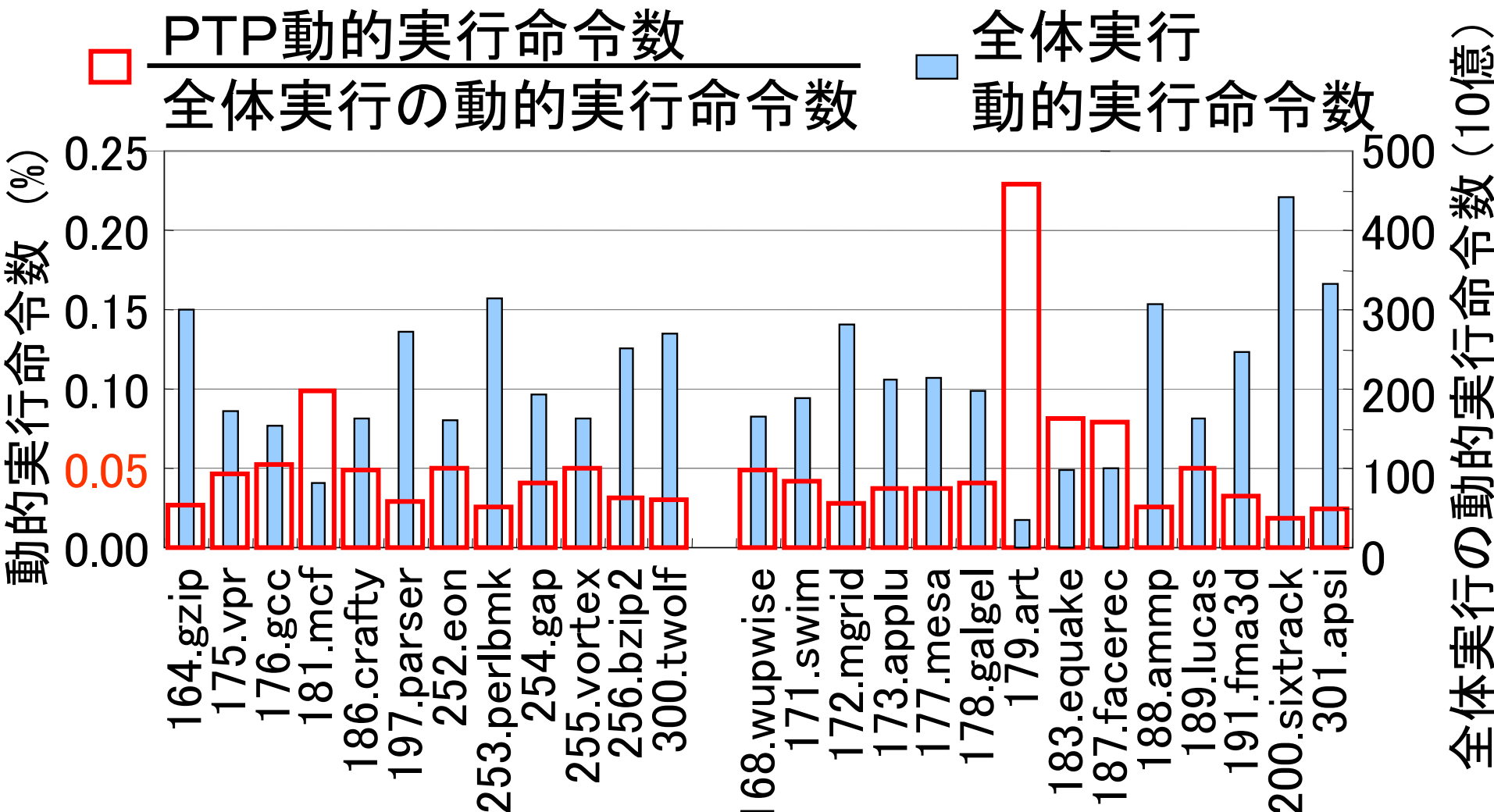
ウォームアップのためだけに必要な領域

- その他
- 複数点を結ぶ補正
- ウォームアップ制御
- ▨ ウォームアップのみ
- ▨ オリジナルとウォームアップ
- ▨ オリジナルのみ



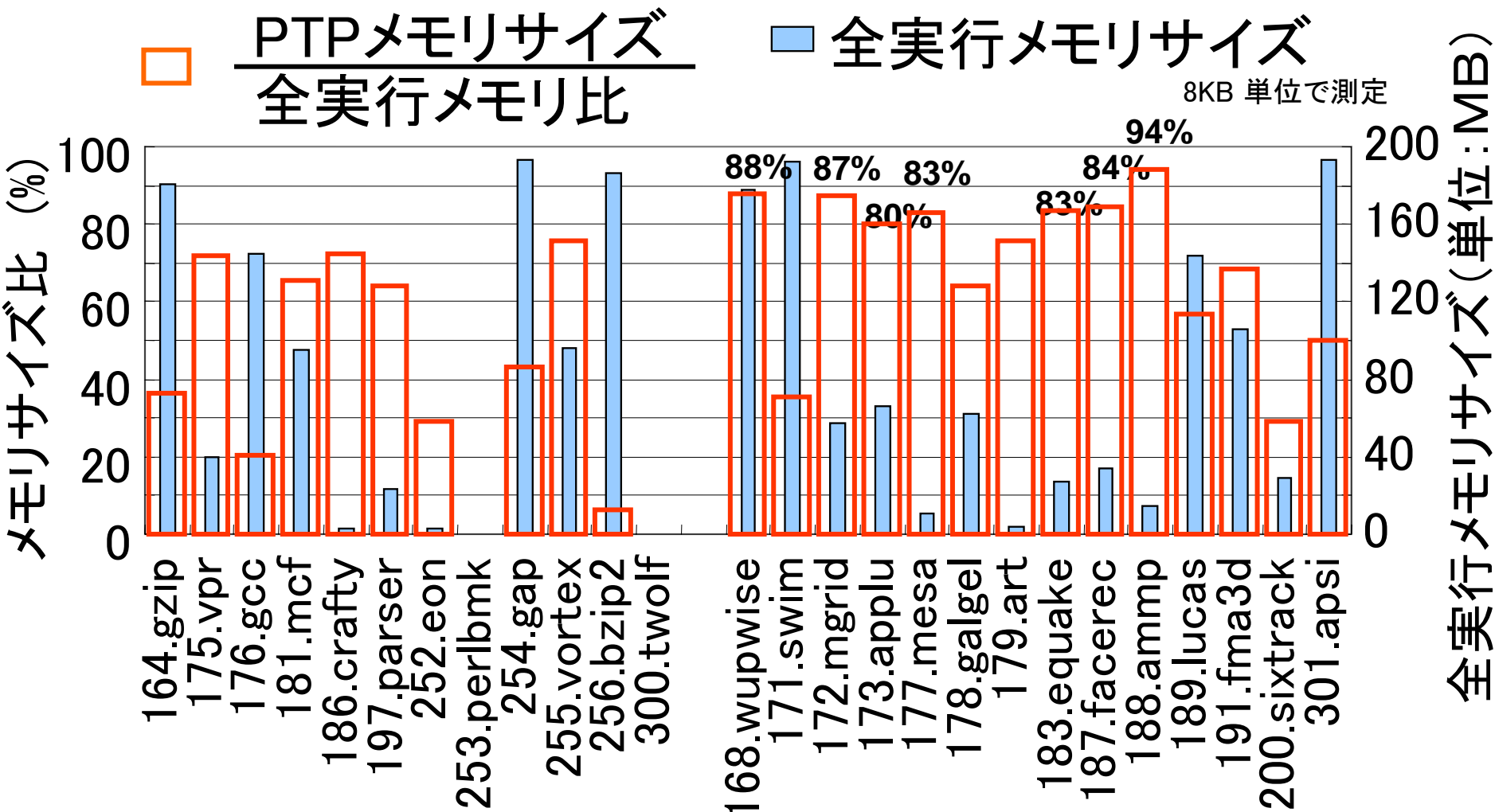


# 動的実行命令数の比較



- 0.05%以下の場合 (実行時間換算で, 1/2000以下)
- もっとも圧縮率の低い179.artは0.25% (1/400)

# メモリサイズ比較



メモリサイズの縮小率は小さい。

全体実行の最大: 約190MB. PTP: 1MB ~ 156MB.

# まとめ

---

- リバーストレーサ
  - リバーストレーサの特徴は、複数アプリやOSの処理が混在する複雑な処理状態を、リアルさを保ちながら1本のテストプログラムで表現する点
  - トレースから実行プログラムを作れることを示した
- SPEC CPU2000でPTPを作成
  - 実行プログラムの作成技術はほぼ確立
  - 性能誤差に課題がある。今後は、サンプリングの精度をあげる
- 非同期割り込みの扱いHPCA8で発表

# 発展

---

- MPをターゲットに，高精度のPTP作成技術の確立
  - 複数本のトレースを入力とし，複数コア間の同期処理をリアリスティックに表現
- まったく新規に設計するシステムのテスト基盤
  - まだ存在しない実機とまだ存在しない周辺チップ
  - 例えば，インターコネクト＋プロセッサコア＋メモリシステムのインターフェース部のリアリスティックかつ広範なテスト

---

Thank you.