



エンタープライズ向け プロセススケジューラの評価および改良

山村 周史 平井 聡 久門 耕一
(株) 富士通研究所

内容

- **背景**

- 標準スケジューラの問題点と他スケジューラの開発動向

- **エンタープライズ向け評価・改良**

1. 2.4系標準スケジューラ内部で多発するキャッシュミスを削減するための改良 (メモリアーキテクチャ)
2. Hyper-Threading対応CPUシステムでの既存スケジューラの性能評価・改良 (CPUアーキテクチャ)

- **まとめ**

背景 ～ Linuxのエンタープライズ分野への対応 ～

• 要件

- 安定性:
高負荷時の性能劣化、システムダウンを避ける
- スケーラビリティ:
CPUなどハードウェア資源の追加に伴う性能スケーラビリティを向上させる

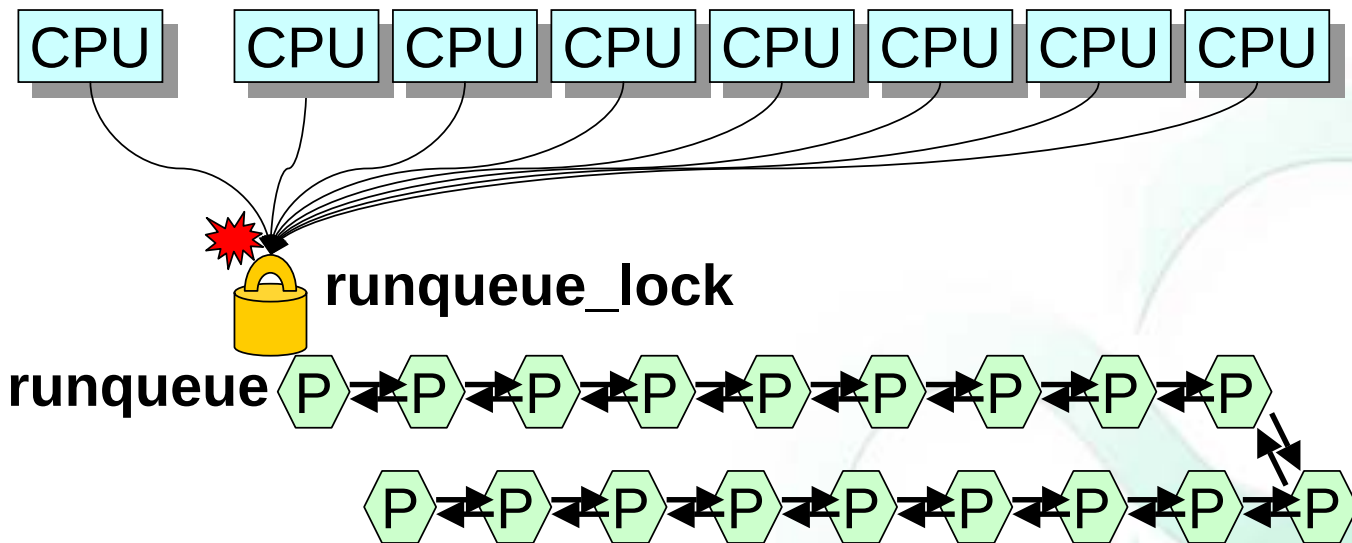
• 対応

- スケジューラ・メモリ管理・NUMAチューニング・I/O性能など...

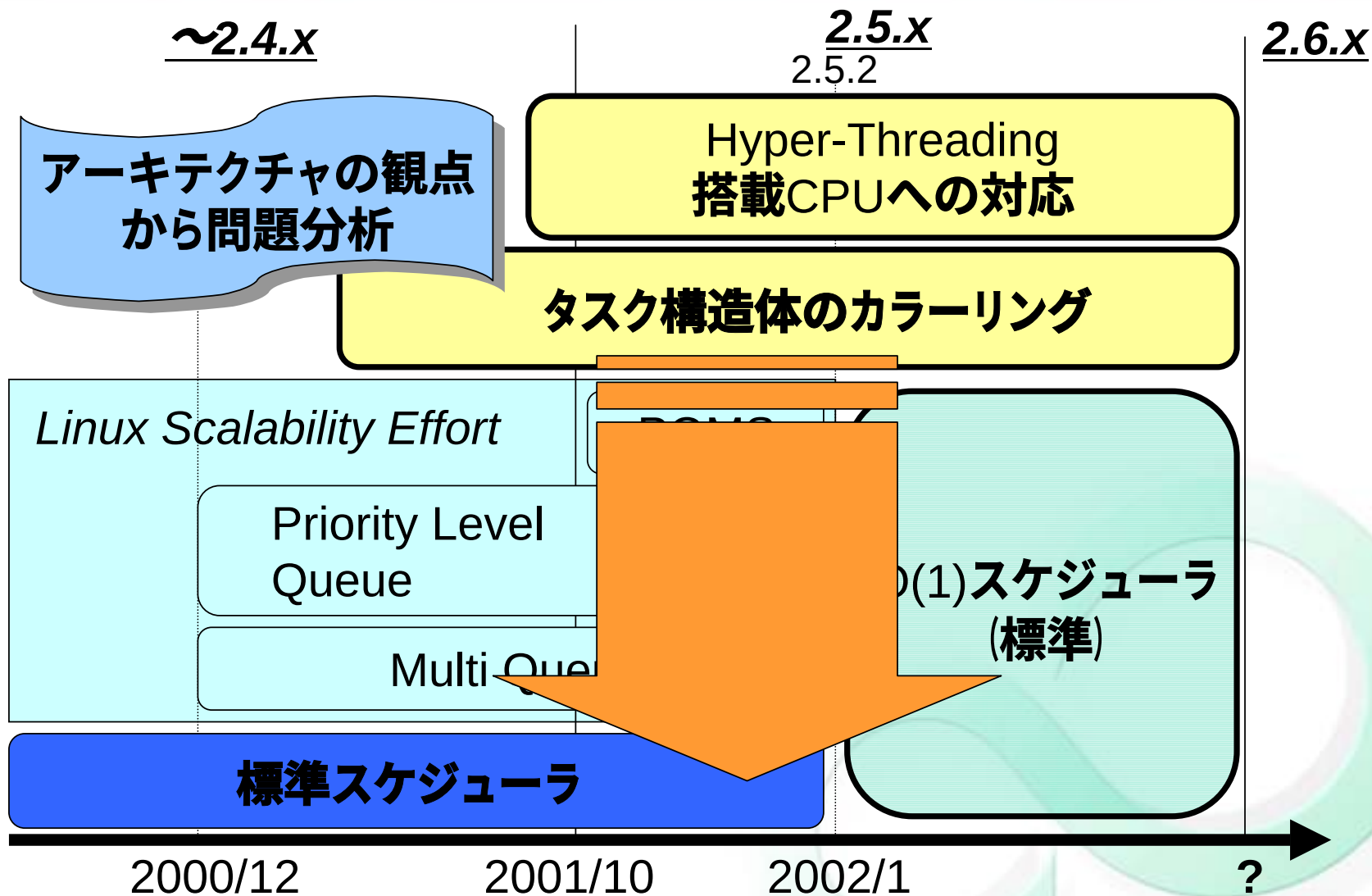
プロセススケジューラがボトルネック

背景 ～ 標準スケジューラの問題 ～

1. すべての実行可能タスクを1本のrunqueueで管理
→ ランキュー走査時間が増大
2. SMP環境では、1個のロック変数で排他制御
→ ロックの競合が増大



背景 ～ プロセススケジューラの開発動向 ～



内容

- 背景

- 標準スケジューラの問題点と他スケジューラの開発動向

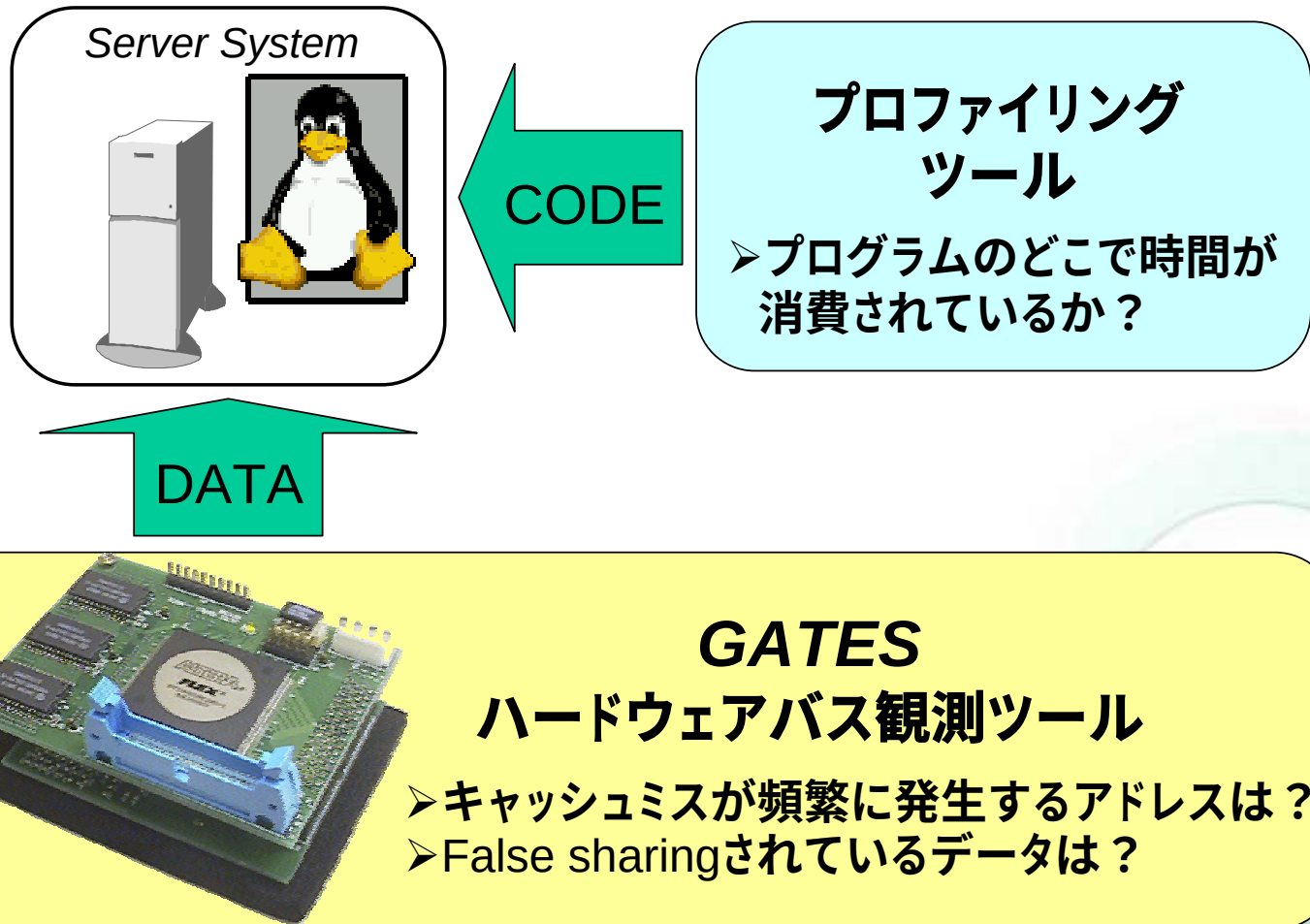
- エンタープライズ向け評価・改良

1. 2.4系標準スケジューラ内部で多発するキャッシュミスを削減するための改良 (メモリアーキテクチャ)
2. Hyper-Threading対応CPUシステムでの既存スケジューラの性能評価・改良 (CPUアーキテクチャ)

- まとめ

問題分析のアプローチ

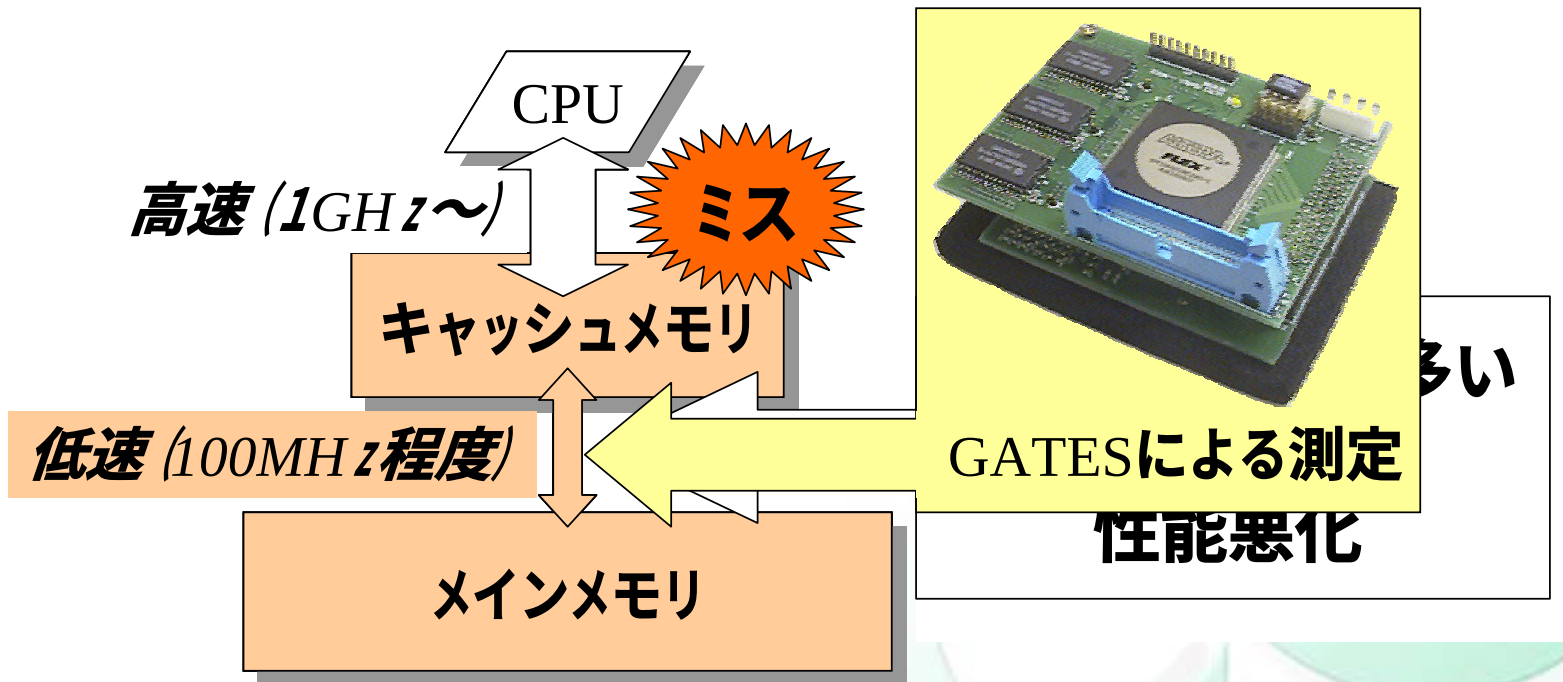
■ メモリアーキテクチャの観点からの定量的な評価



メモリシステムの分析

- 現在のコンピュータシステムの性能は、メモリシステムの性能が大きく左右する

➡ 高負荷時の挙動をメモリシステムから調査



高負荷時におけるバストランザクション

4-way Pentium Proサーバ上で256 httpdプロセスが走行中

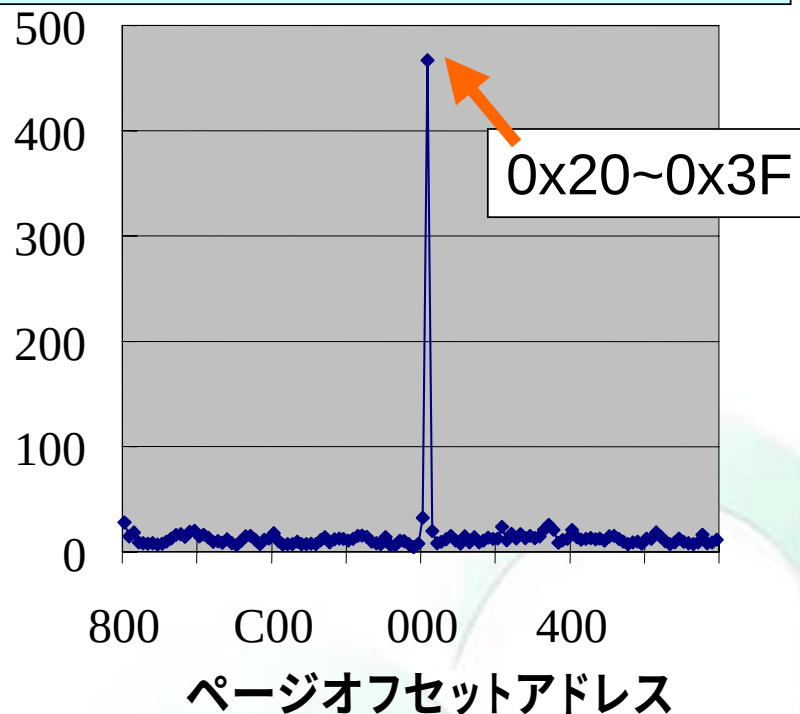
メモリバストランザクション の分析

ページオフセットアドレス0x030
で多数のバストランザクションが
発生

カーネルプロファイリング

スケジューラ内部でOS実行時
間の10%~20%を消費

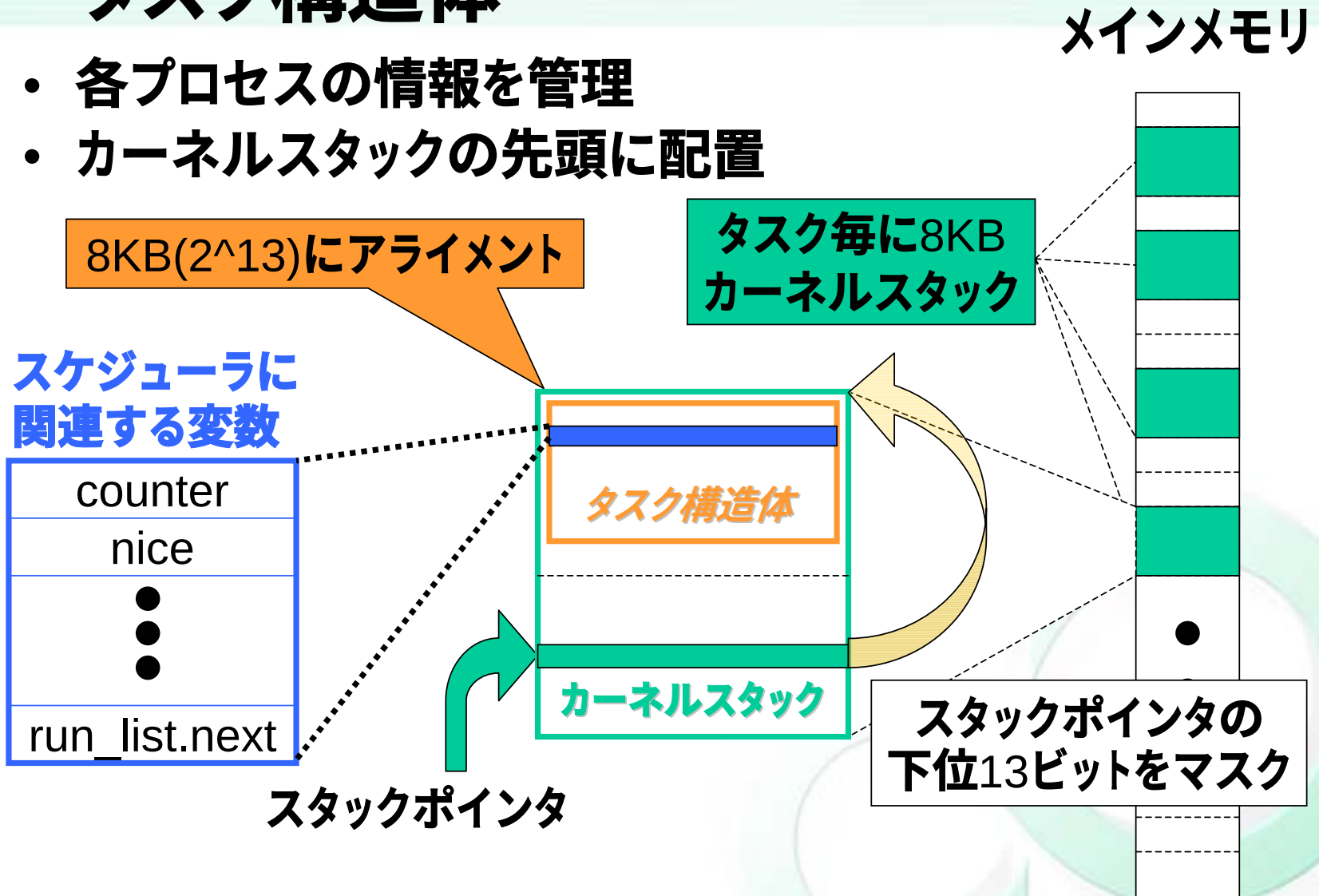
1リクエストあたりのメモリアクセス回数



**プロセススケジューラ内部でランキュー探索中に
多数のキャッシュミスが発生**

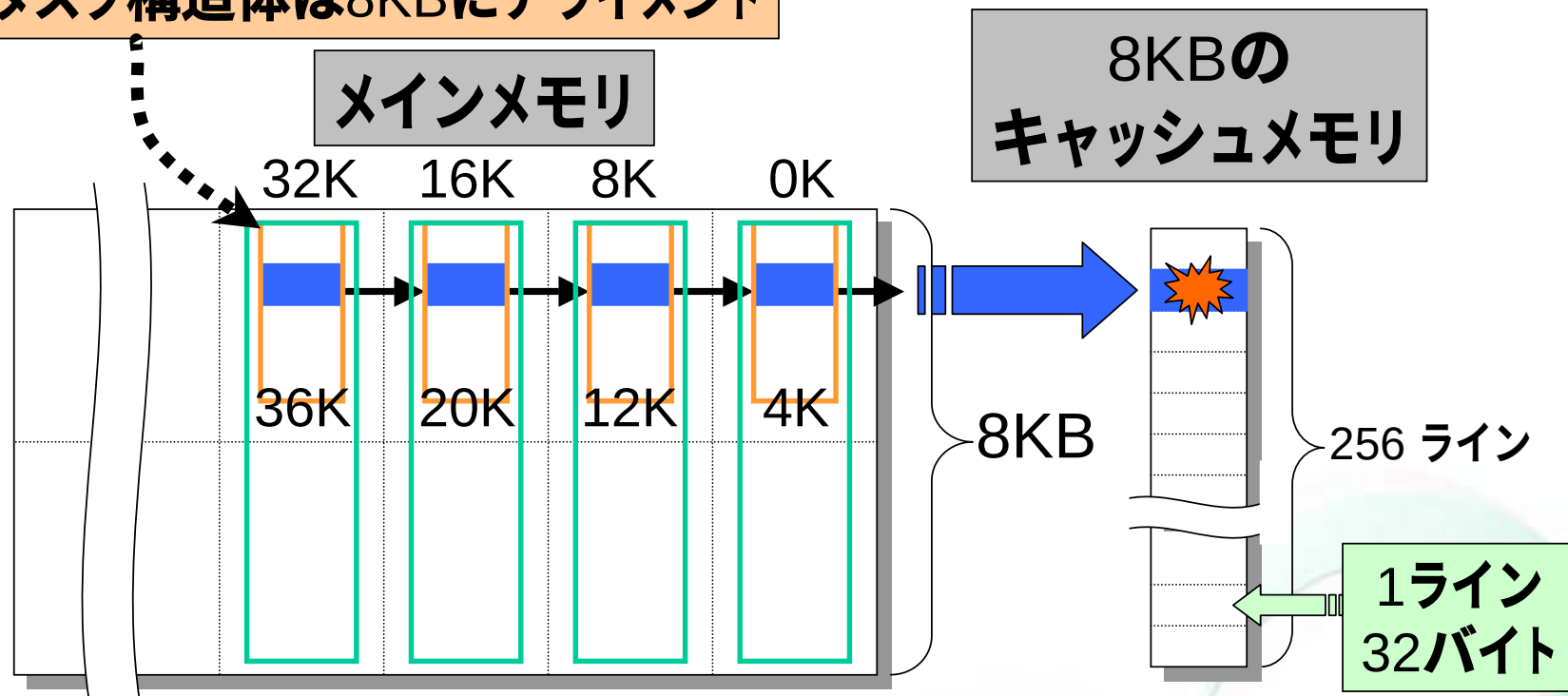
タスク構造体

- 各プロセスの情報を管理
- カーネルスタックの先頭に配置



スケジューラ内部で発生するキャッシュライン競合

タスク構造体は8KBにアライメント

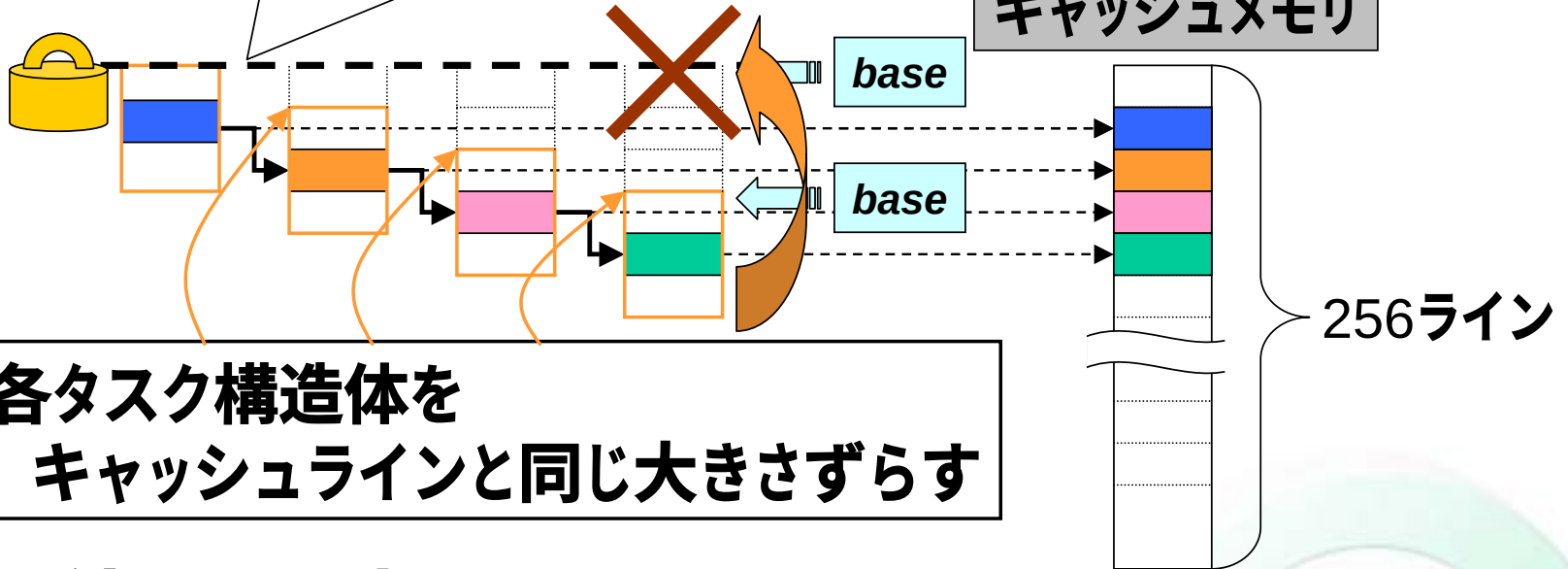


実測値：2次キャッシュミス率 > 90%
通常は、ミス率1%未満

解決法 ーキャッシュカラーリングー

8KB境界にアライメント

8KBの
キャッシュメモリ



各タスク構造体を
キャッシュラインと同じ大きさずらす

期待できる効果

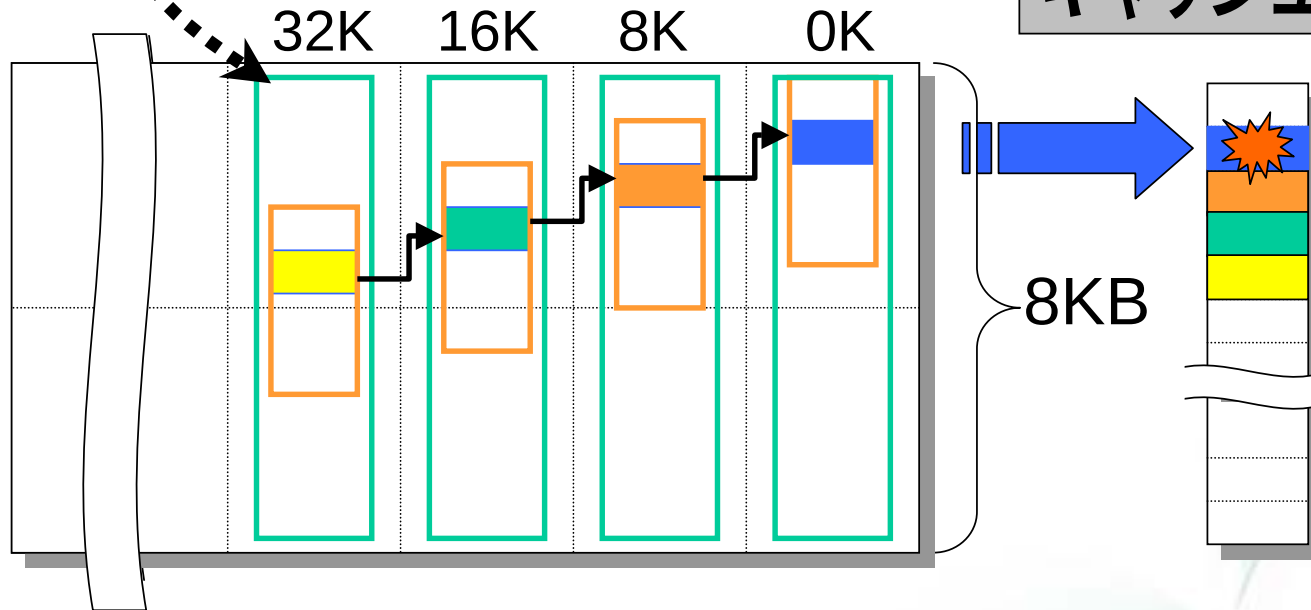
- キャッシュミスが減少しrunqueueの走査時間が短縮
- SMPシステムでは, ロック競合率が減少
- ✗ 移動したタスク構造体が参照できない

キャッシュカラーリングの実装

タスク構造体は8KBにアライメント

メインメモリ

8KBの
キャッシュメモリ

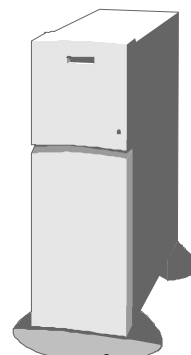


アドレスの上位ビットを使ってハッシュしてベースアドレスを決定

評価環境

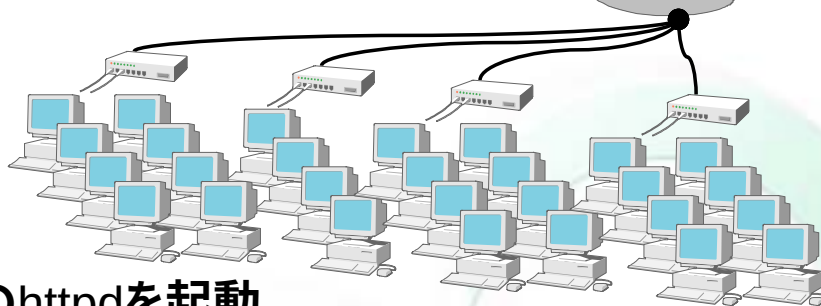
サーバ

CPU	Pentium III Xeon 550 MHz × 8
2次キャッシュサイズ	1MB (4-way S.A)
メインメモリ	1GB
ディストリビューション	RedHat 7.1 (US)
カーネル	2.4.4

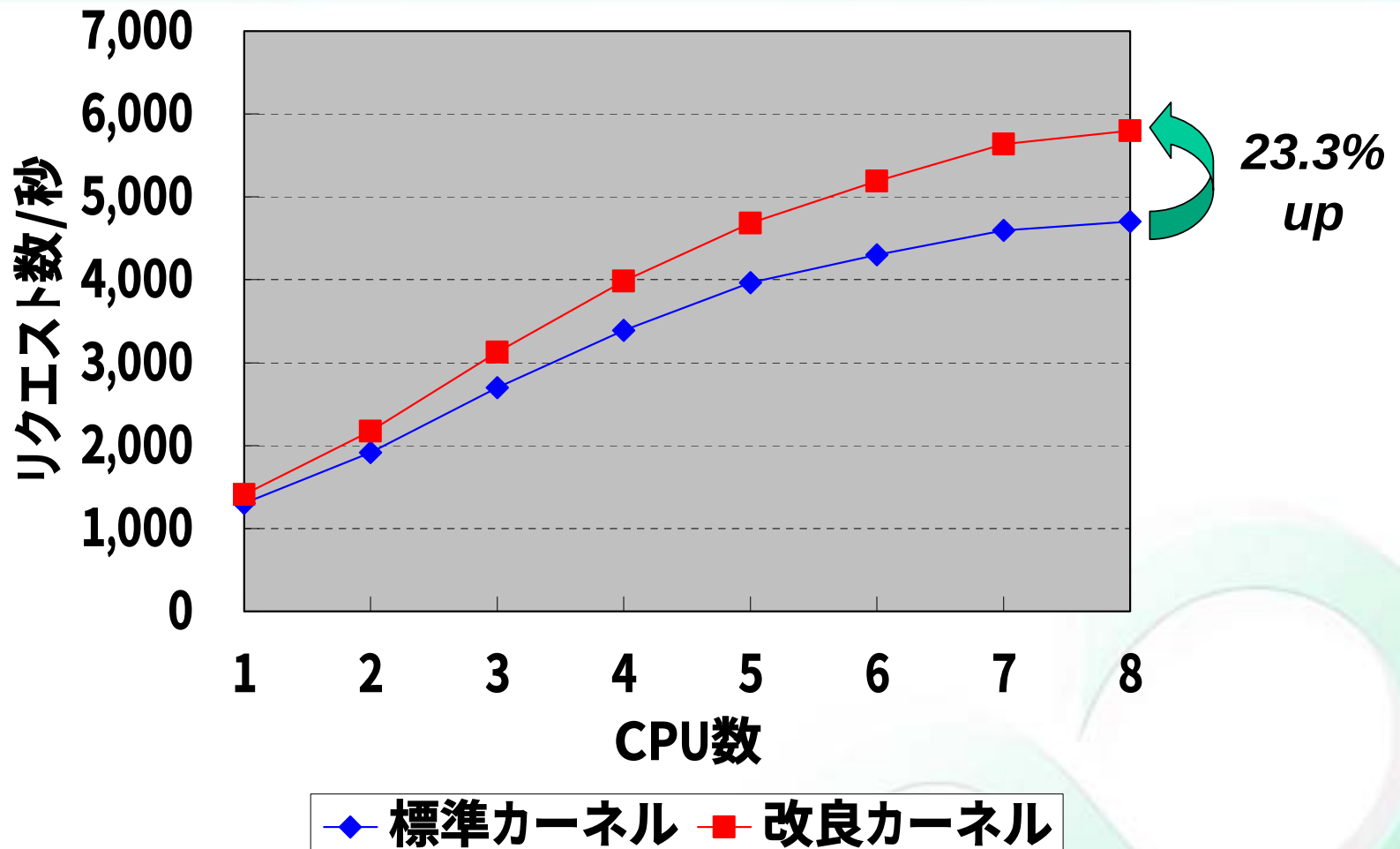


アプリケーション

- WebBench 3.0
サーバ: Apache 1.3.19. 256個のhttpdを起動.
クライアント: 28 PC/AT. 合計256個のクライアントスレッド.
- Chat micro benchmark
チャットルームでのメッセージ交換をシミュレート.
2400のスレッドが起動 (WebBenchよりも多い) .

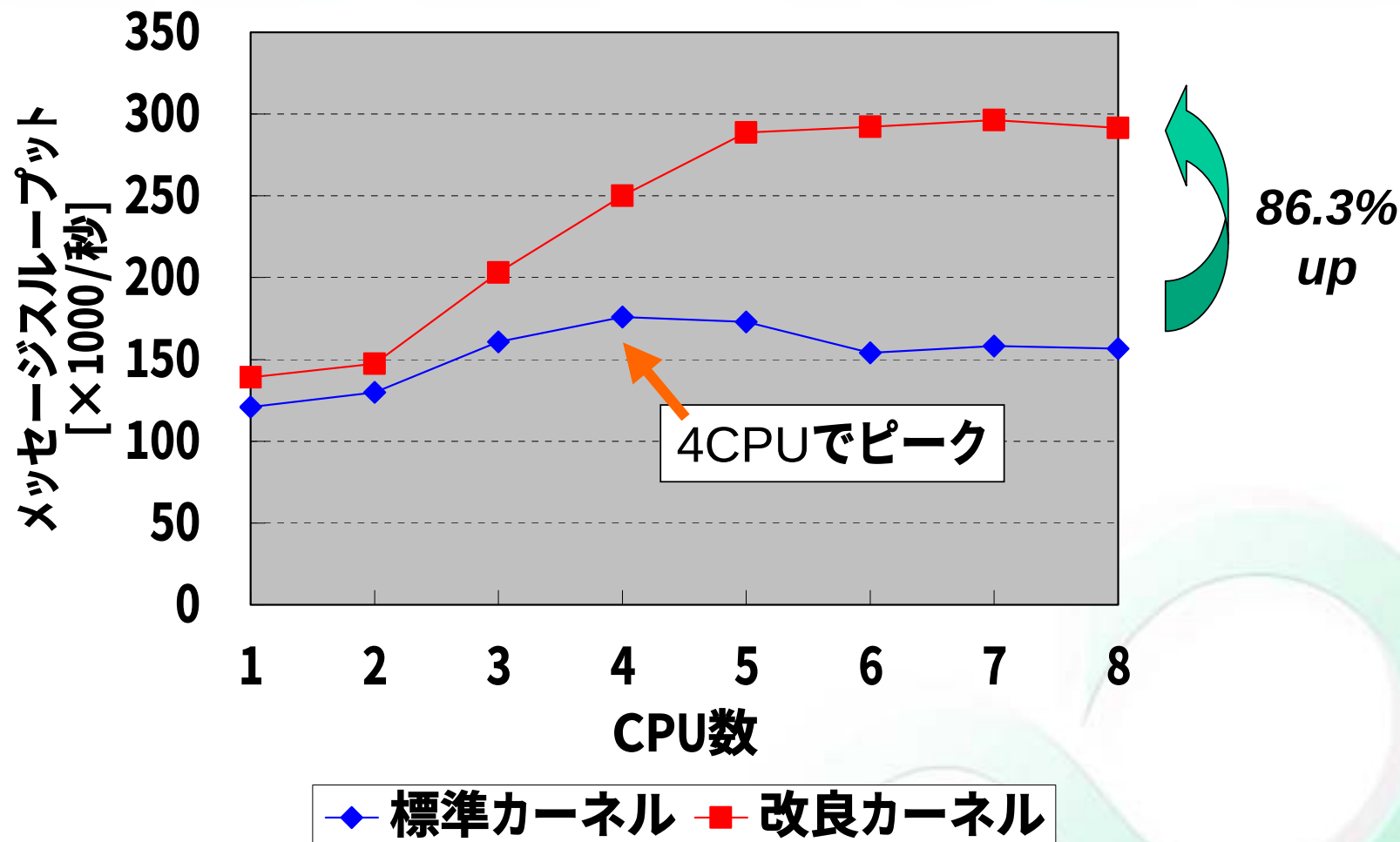


評価結果【WebBench】



8CPUの場合に最大23.3% の性能向上

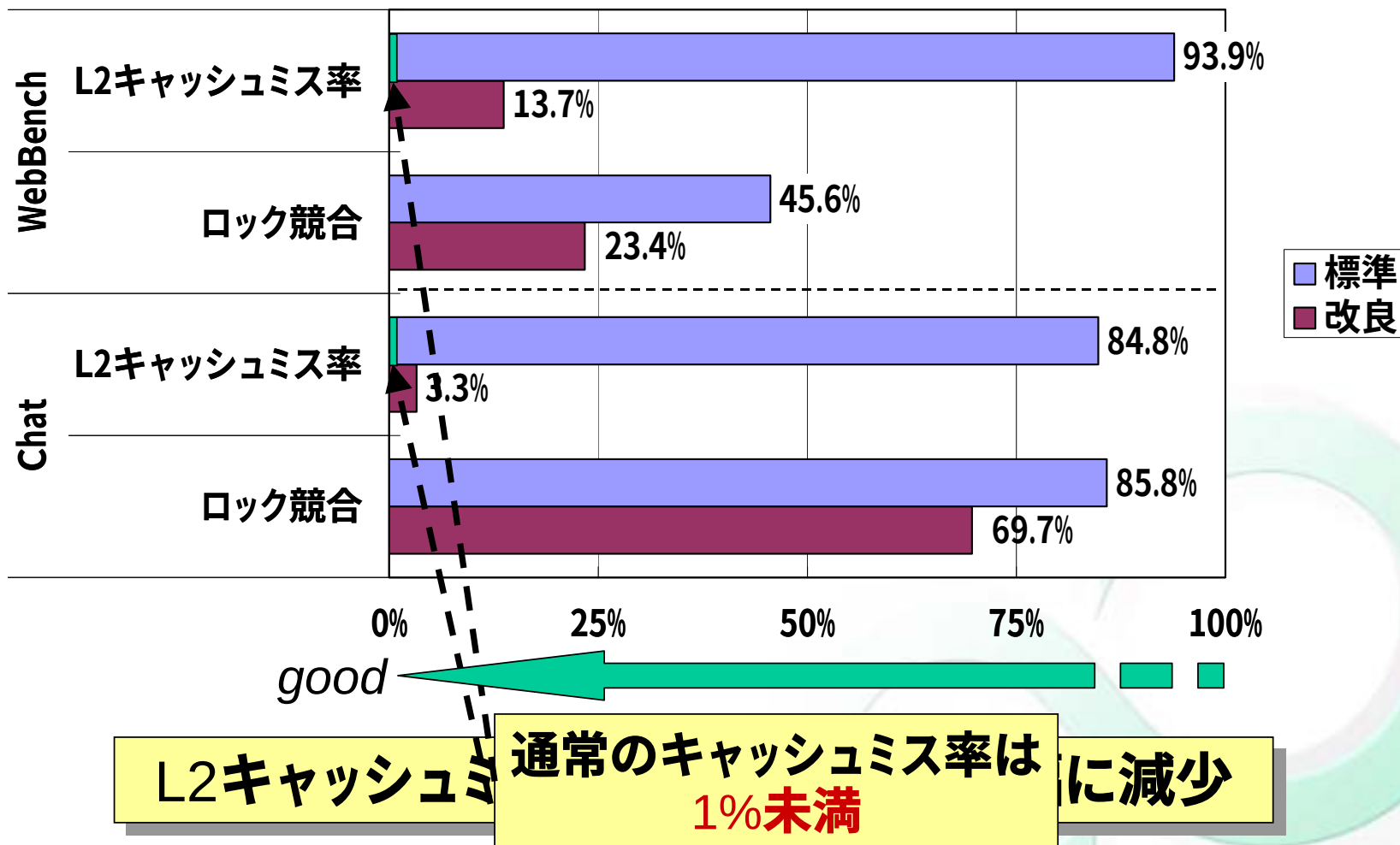
評価結果【Chat】



8CPUの場合に86.3% の性能向上

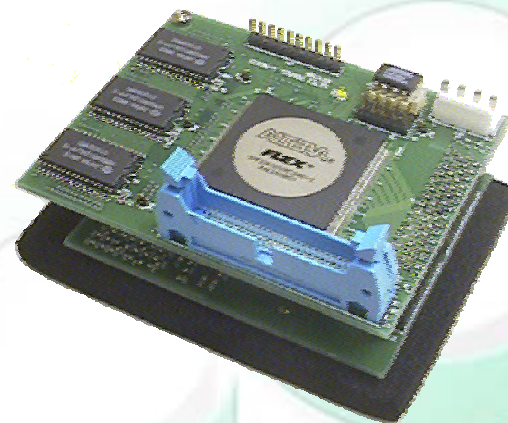
L2キャッシュミス率とロック競合(8CPUの場合)

L2キャッシュミス率は、runqueue探索中のみにについて測定



タスク構造体カラーリングのまとめ

- プロセススケジューラ内部で多発するキャッシュミスを確認.
- タスク構造体のカラーリングにより解決.
わずかな修正でスケーラビリティが向上.
 - Webサーバ性能: 23.3%UP (8CPU時) .
 - Chatサーバ性能: 86.3%UP (8CPU時) .
- メモリアーキテクチャからのチューニングが有効.



内容

- 背景

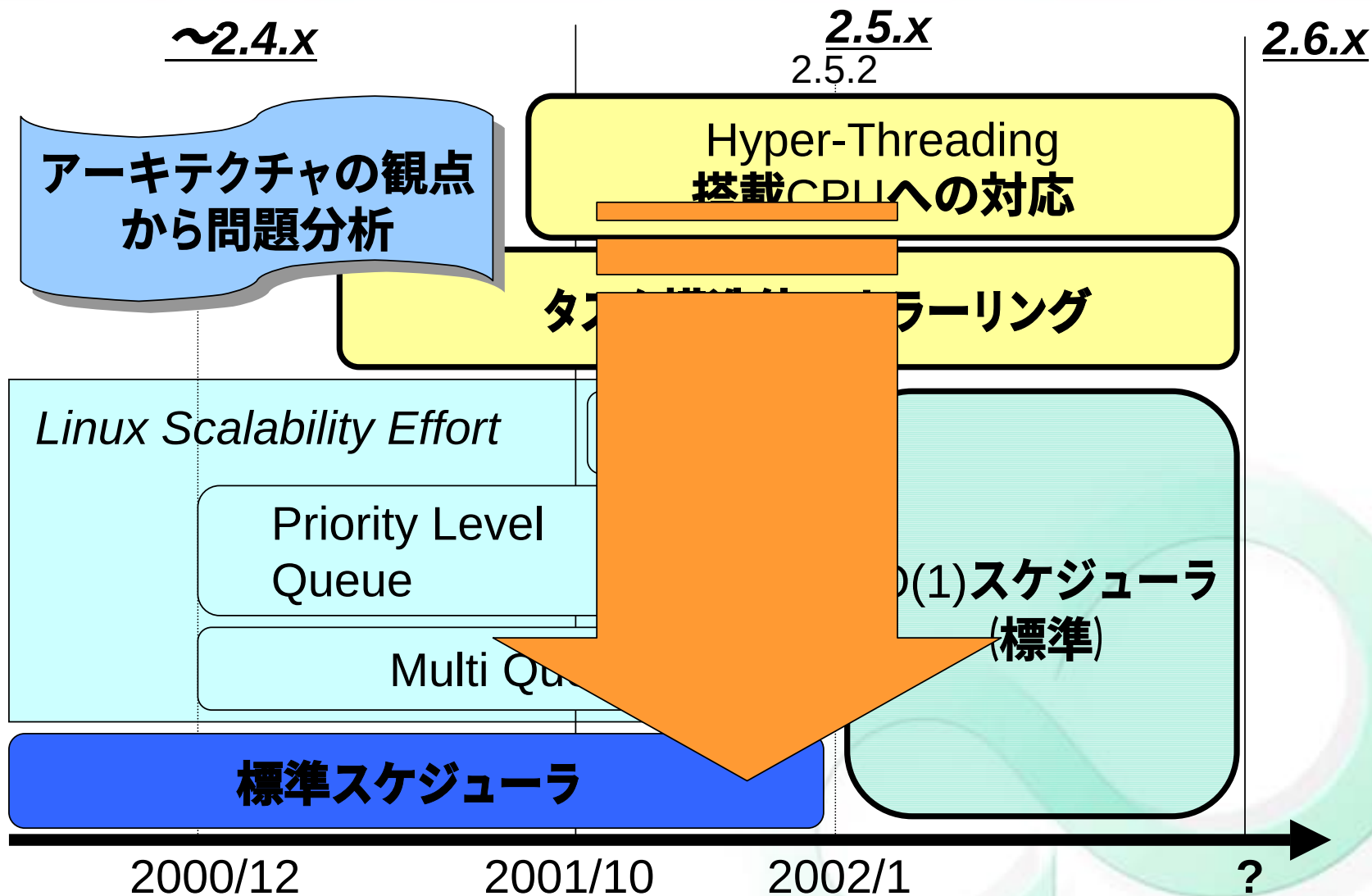
- 標準スケジューラの問題点と他スケジューラの開発動向

- エンタープライズ向け評価・改良

1. 2.4系標準スケジューラ内部で多発するキャッシュミスを削減するための改良 (メモリアーキテクチャ)
2. Hyper-Threading対応CPUシステムでの既存スケジューラの性能評価・改良 (CPUアーキテクチャ)

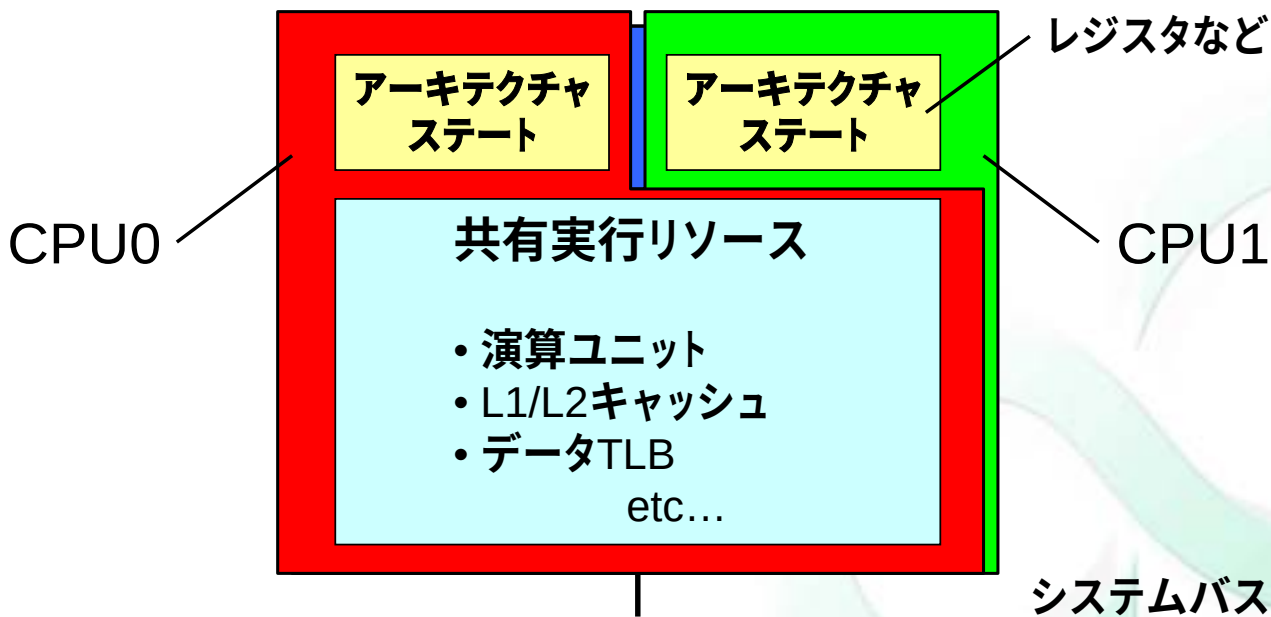
- まとめ

背景



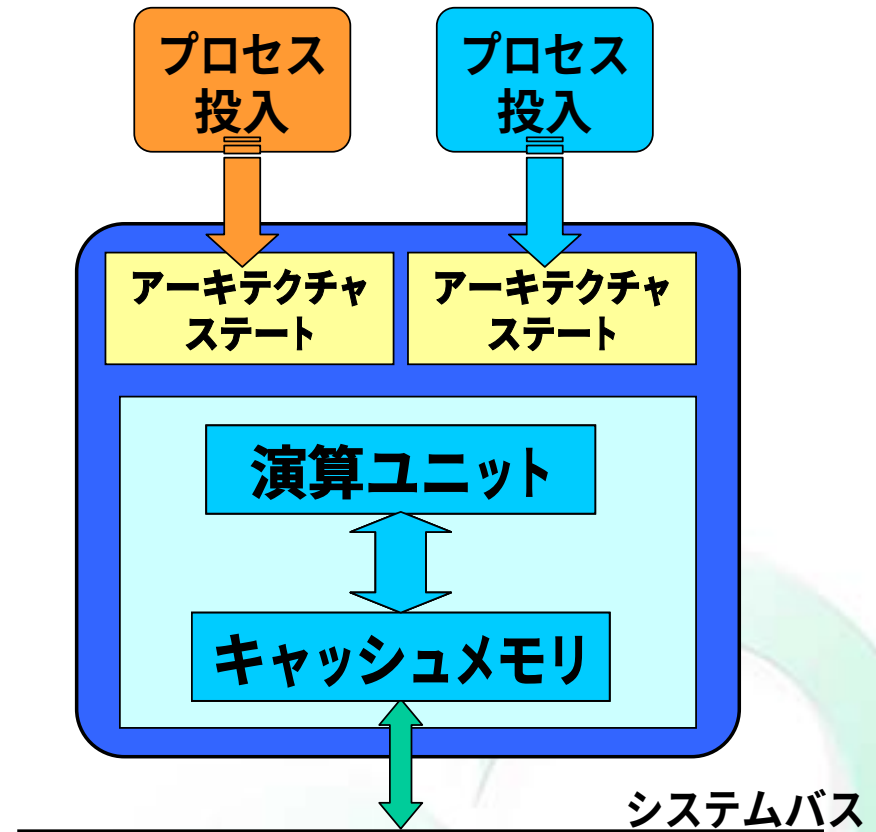
Hyper-Threading機能の概要

- Intel Xeon **プロセッサに搭載された新機能**
 - 一般には, SMT(Simultaneous Multi-Threading)と呼ばれる
- **1つの物理CPU の中に2つの論理CPU**
 - レジスタなどのアーキテクチャステートを2つ保持
 - 論理CPU間で実行リソースを共有



Hyper-Threading機能の特徴

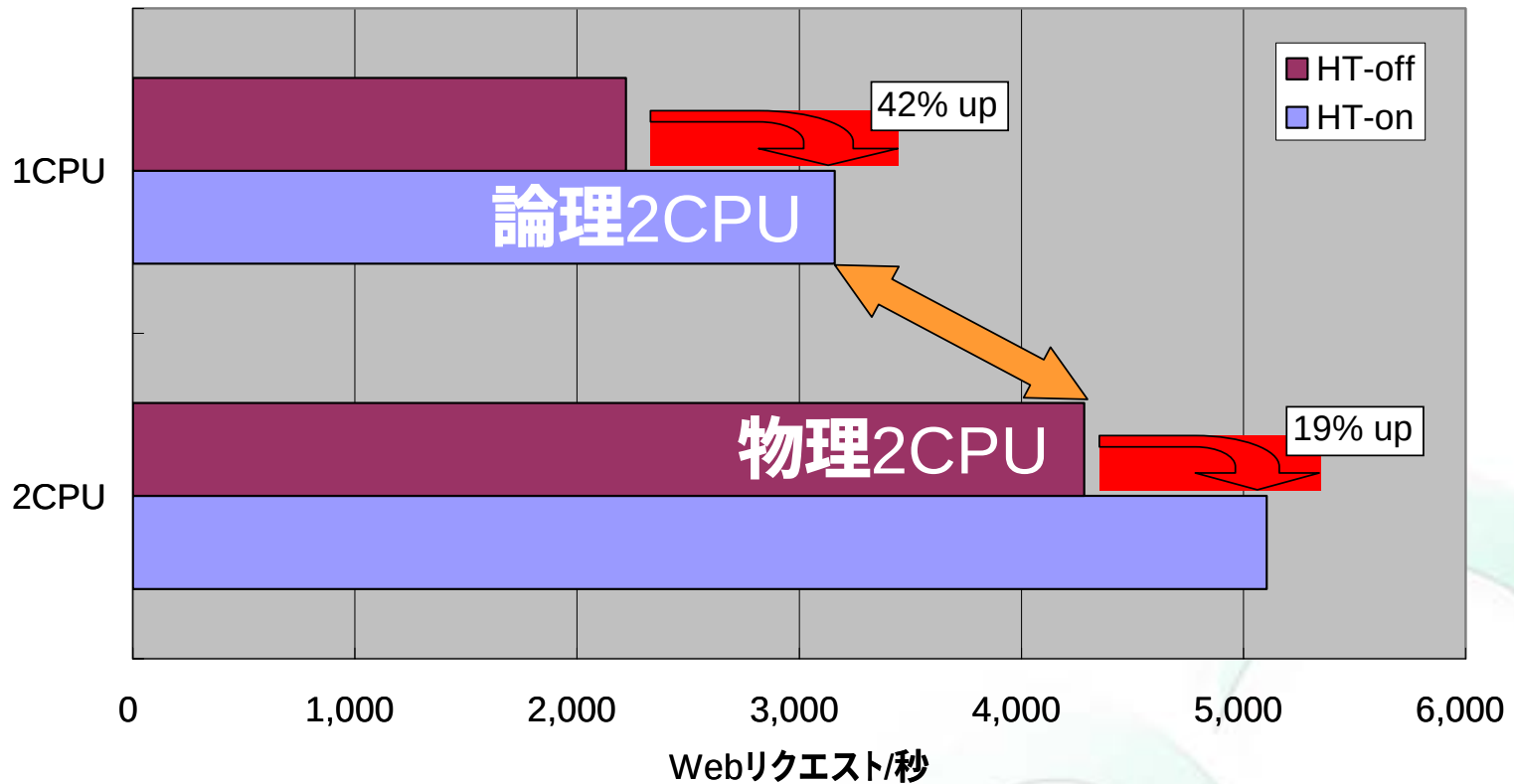
- 演算リソースのメモリアクセス遅延による「空き」を利用して、スループットを向上
- ソフトウェア (カーネル) から見ると2CPUのSMPシステム



従来のSMPカーネルがそのまま動作

Webサーバ性能測定結果

Xeon 2GHz／カーネル2.4.18(標準スケジューラ)／Apache 1.3.19

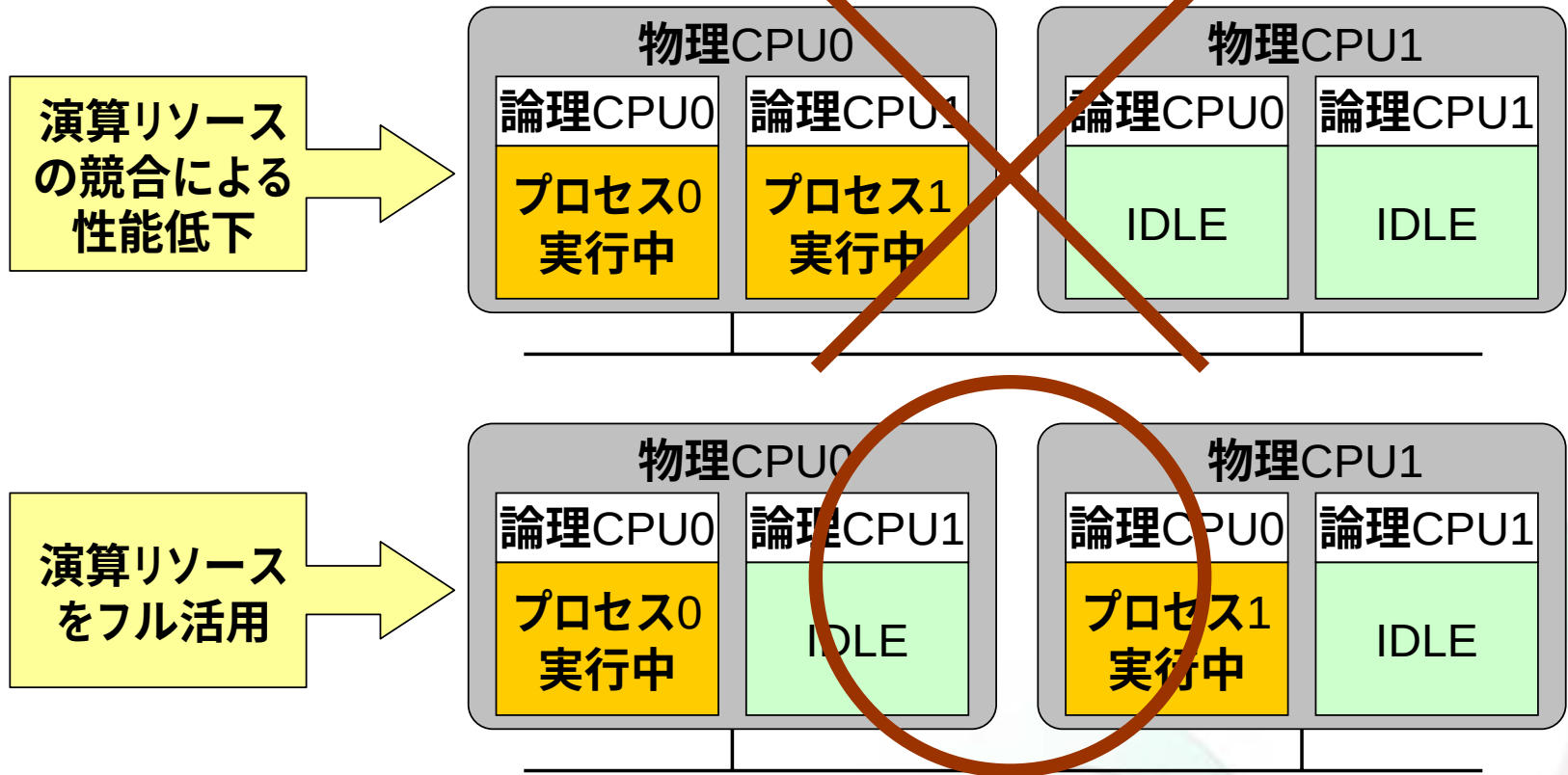


Hyper-Threading機能によりWebサーバ性能が向上

共有リソースの競合によって、論理2CPU < 物理2CPU

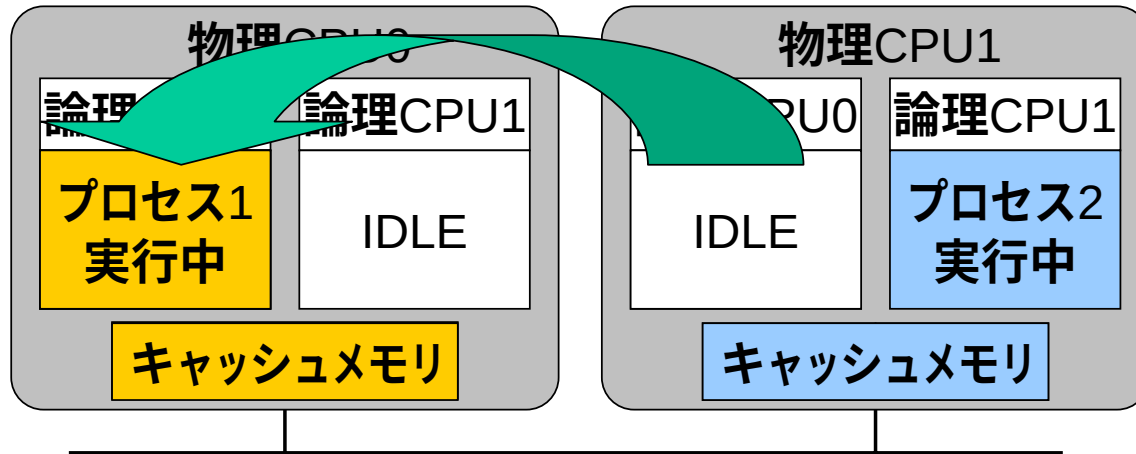
SMPシステム上でのスケジューラの問題

2SMPシステム上で2つのプロセスが走行している場合



**原則：可能な限り1つの物理CPUにつき
1プロセスが走行するようにスケジューリング**

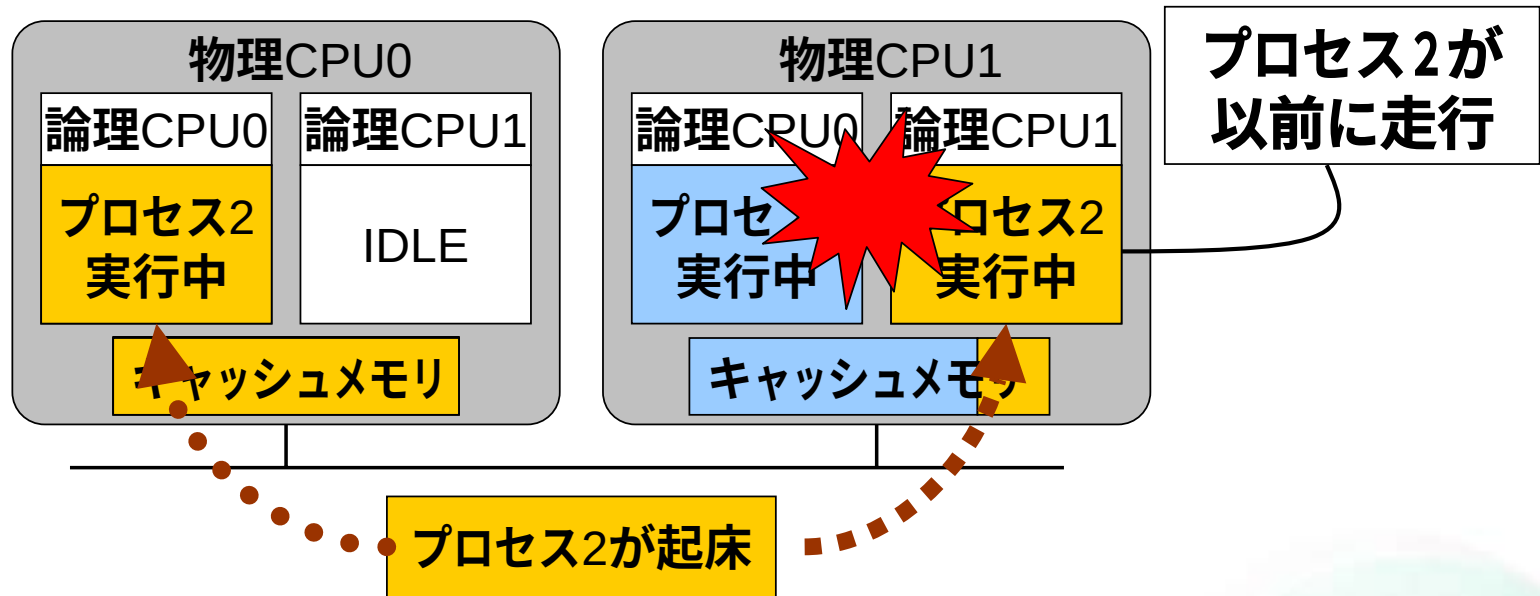
改良 (1) 実行中のプロセスが減った場合



□ schedule()関数の修正

- Runqueueに並んだプロセスがすべてクアンタムを使い切った
- Idle状態にある物理CPUが存在
を満たす場合、idle状態の物理CPUにプロセスを移動

改良 (2) 実行中のプロセスが増えた場合



□ reschedule_idle()関数の修正

- ・ プロセス起床時／フォーク時に呼び出される
- ・ 【修正前】 当該プロセスが以前走行していたCPUがidleであれば、そのCPUにディスパッチ
- 【修正後】 Idle状態にある物理CPUが存在するか他の条件より優先してチェック

改良スケジューラの効果

【実験方法】

- Intel Xeon 2.0 GHz×2CPU／カーネル2.4.18を使用
- 4つのプロセスを起動し、すべてのプロセスが終了するまでの時間を測定 (10回測定)
 - 1CPUで実行時間 2秒のループ処理 × 2プロセス
 - 1CPUで実行時間10秒のループ処理 × 2プロセス

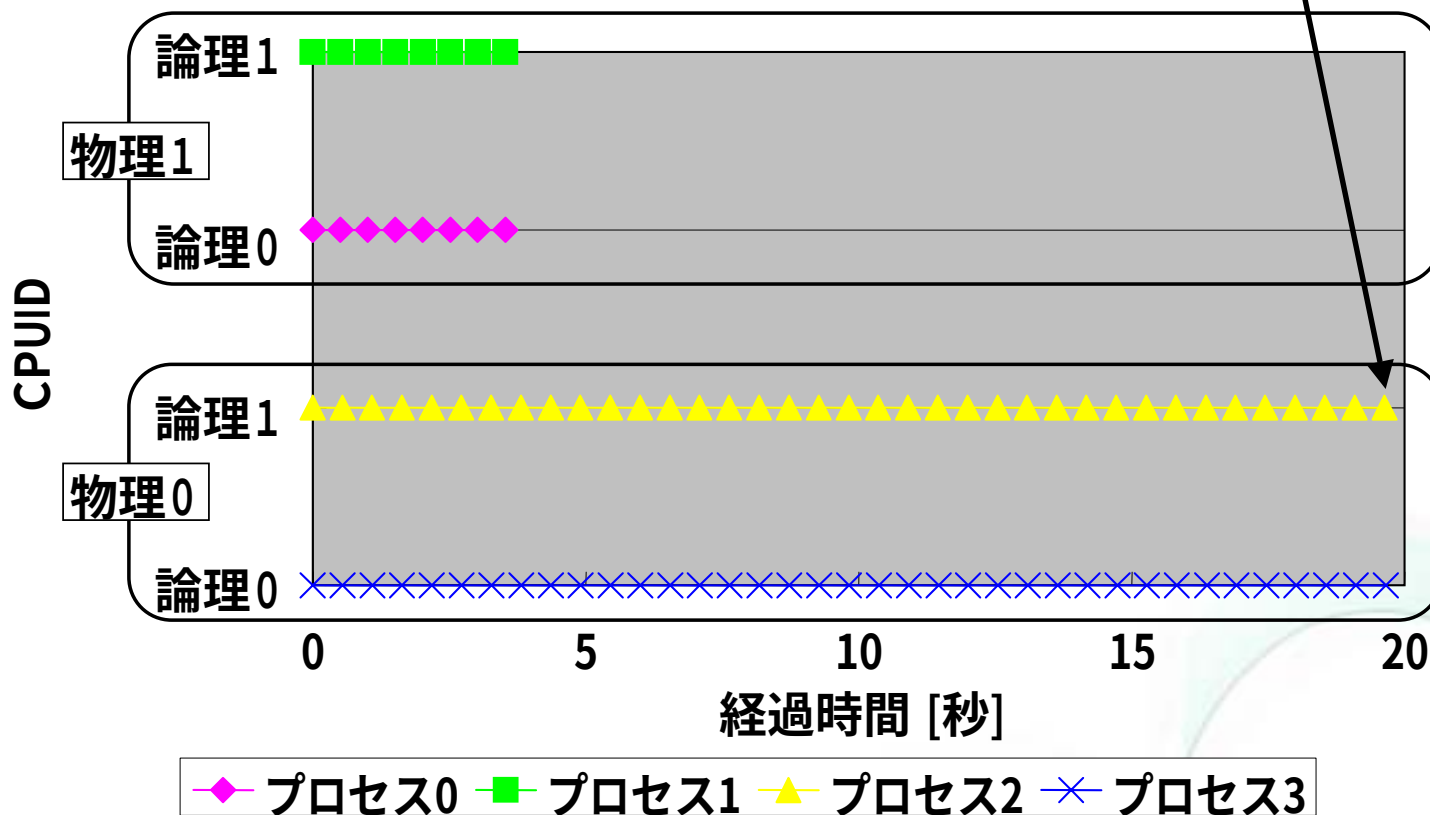
【結果】

	最短 [秒]	最長 [秒]	平均 [秒]
標準スケジューラ	12.6	19.8	15.5
改良スケジューラ	12.6	12.7	12.6

- 標準スケジューラでは、実行時間に大きなばらつき
- 最大で1.57倍の性能差

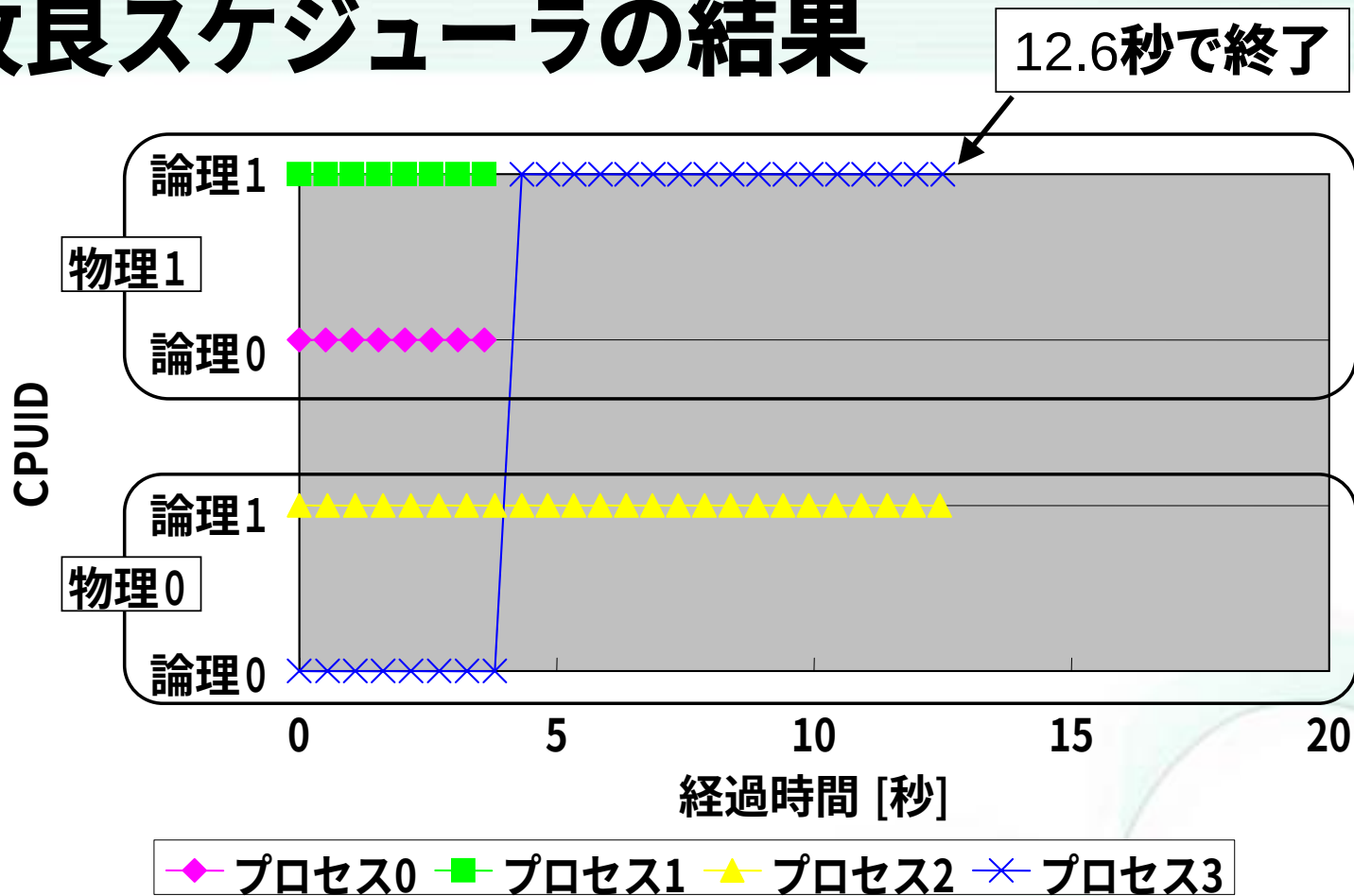
標準スケジューラの結果

19.8秒で終了



2つのプロセスが同じ物理CPU上で動作しつづける

改良スケジューラの結果



他方の物理CPUがidleになったときプロセスが移動

Hyper-Threading向けスケジューラ改良のまとめ

■ CPUアーキテクチャを意識したスケジューリング

- テストプログラムによる評価で1.5倍以上の性能差

■ エンタープライズ分野での

Hyper-Threadingの利用

- プロセスが少ない場合に誤ったスケジューリング
- 他プロセスの有無で見かけ上のCPUの性能が大きく変動

timeコマンドの結果が変動 → 課金システムで問題

まとめ

- **アーキテクチャの観点からスケジューラを評価・改良**

- タスク構造体のカラーリング (メモリアーキテクチャ)
- Hyper-Threading機能への対応 (CPUアーキテクチャ)

アーキテクチャ的観点からの分析と定量的な評価が有効

- **今後のエンタープライズ向けスケジューラ開発**

- **Hyper-Threading対応**

- ✓ 開発カーネル標準のO(1)スケジューラへの展開
- ✓ O(1)スケジューラへのパッチの投稿があった

- **エンタープライズ分野において我々の分析手法を活用**

- ✓ カーネル・アプリケーションを含めたチューニング