タスク構造体のカラーリングによるプロセススケジューラの高速化

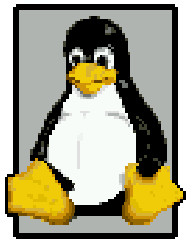
山村 周史　平井 聡　久門 耕一

株式会社　富士通研究所

内容

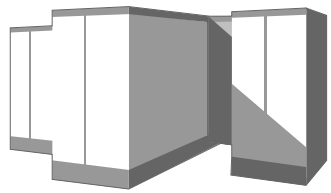
- 背景
- 性能ボトルネック
- Linuxカーネルの問題点
- タスク構造体のカラーリング
- 性能評価

背景 (1)　～安定性への要求～

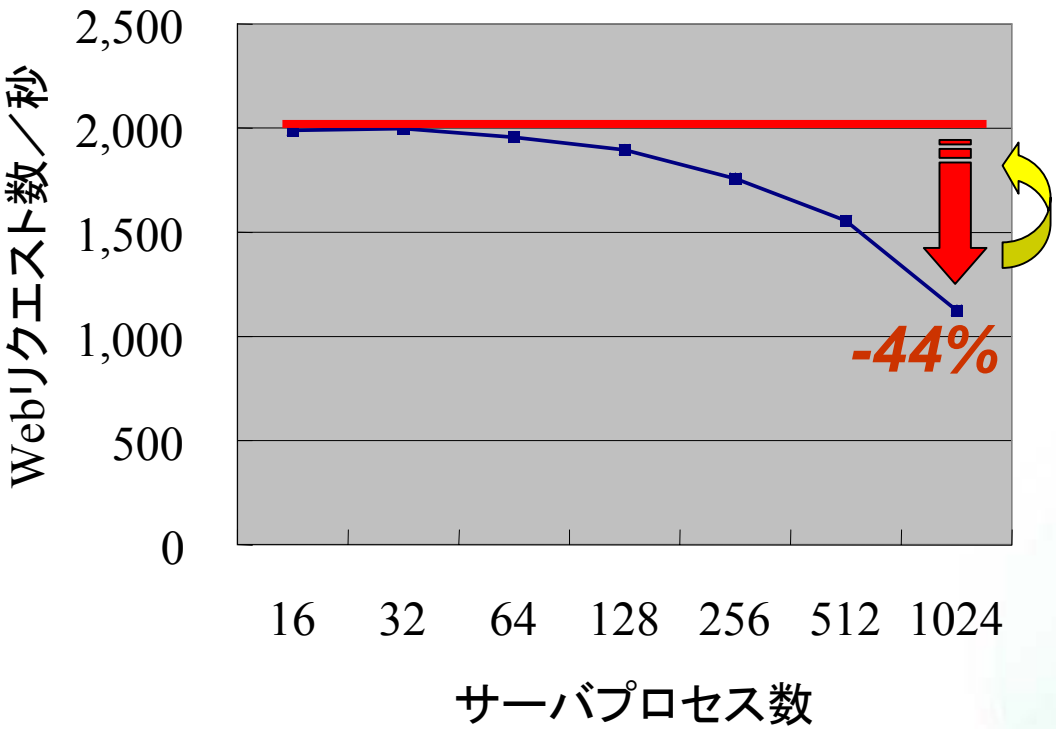


Linuxをエンタープライズ分野への適応が目標

その要件として

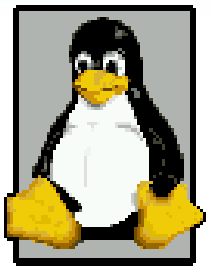


- 高負荷な状況でも安定した性能を発揮
- CPU数の増加に見合った高い性能スケーラビリティ



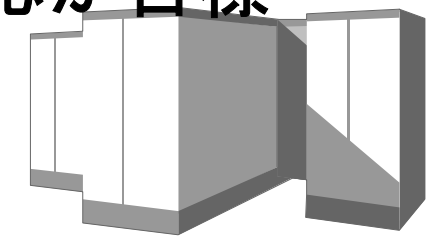
- **サーバ**
  4-way Pentium Pro 200MHz
  2次キャッシュサイズ 1,024KB
- **アプリケーション**
  Apache 1.3.19

背景 (2)　～性能スケーラビリティへの要求～

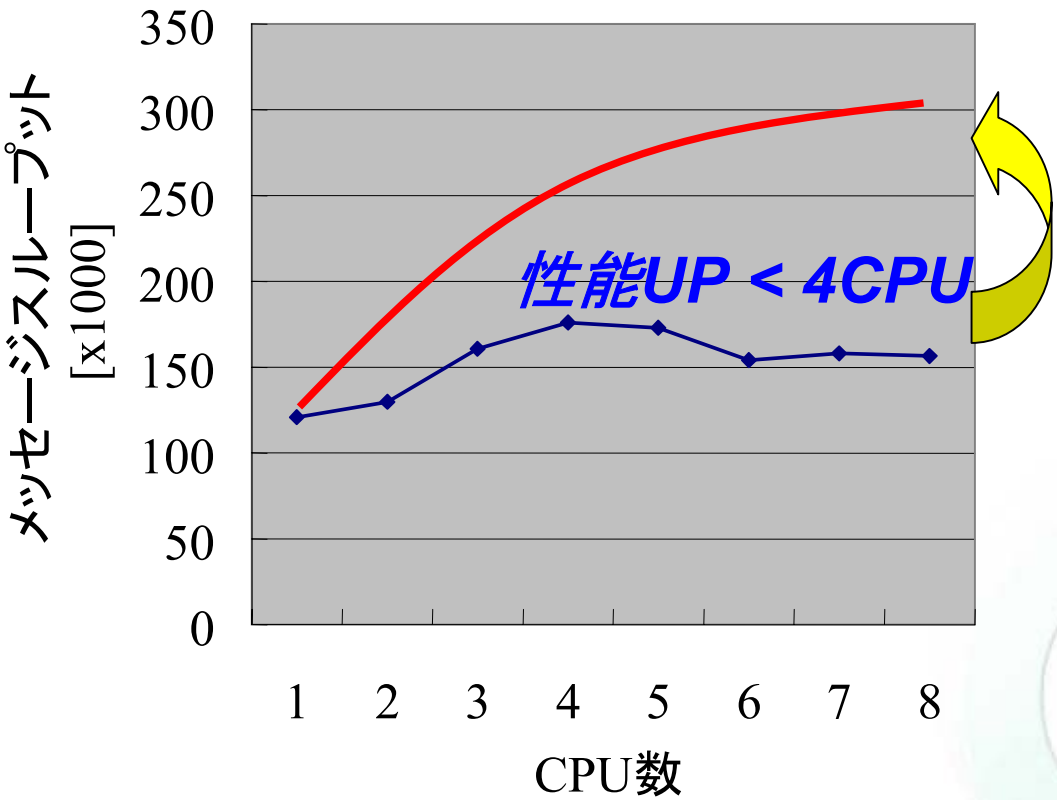


Linuxをエンタープライズ分野への適応が目標

その要件として

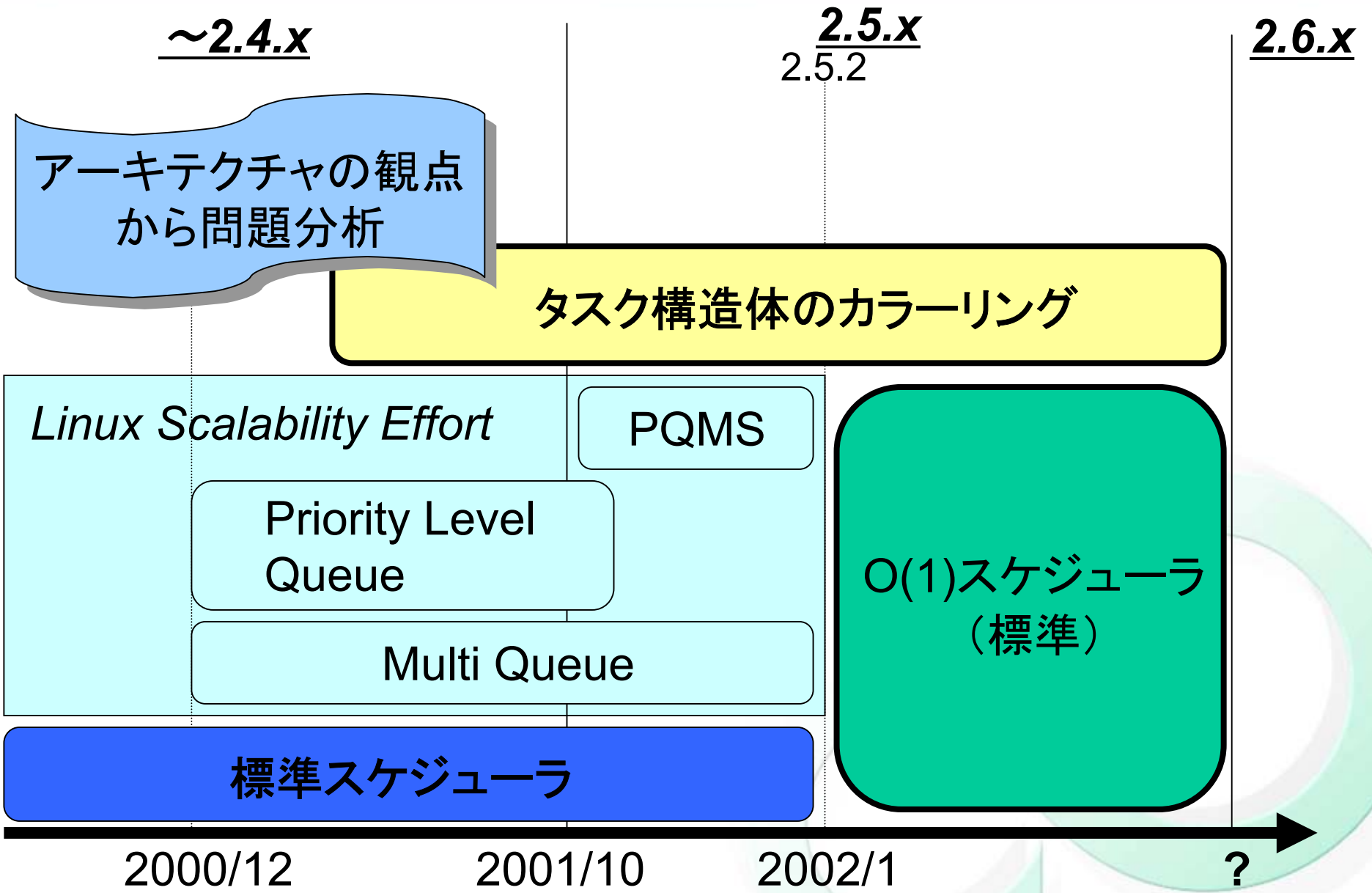


- 高負荷な状況でも安定した性能を発揮
- CPU数の増加に見合った高い性能スケーラビリティ



- サーバ
  8-way Pentium III 550MHz
  2次キャッシュサイズ1,024KB
- アプリケーション
  Chat micro benchmark

背景 ~プロセススケジューラの開発動向~

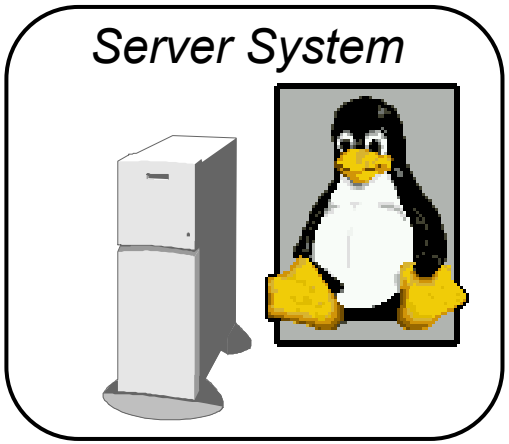


内容

- 背景
- 性能ボトルネック
- Linuxカーネルの問題点
- タスク構造体のカラーリング
- 性能評価

問題分析のアプローチ (1)

- メモリアーキテクチャの観点からの定量的な評価

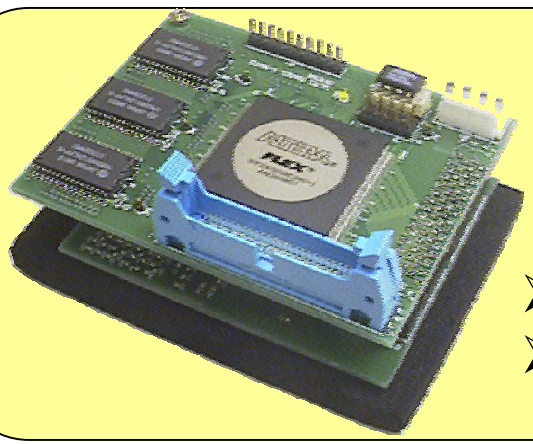


CODE

プロファイリング
ツール

➢プログラムのどこで時間が消費されているか？

DATA



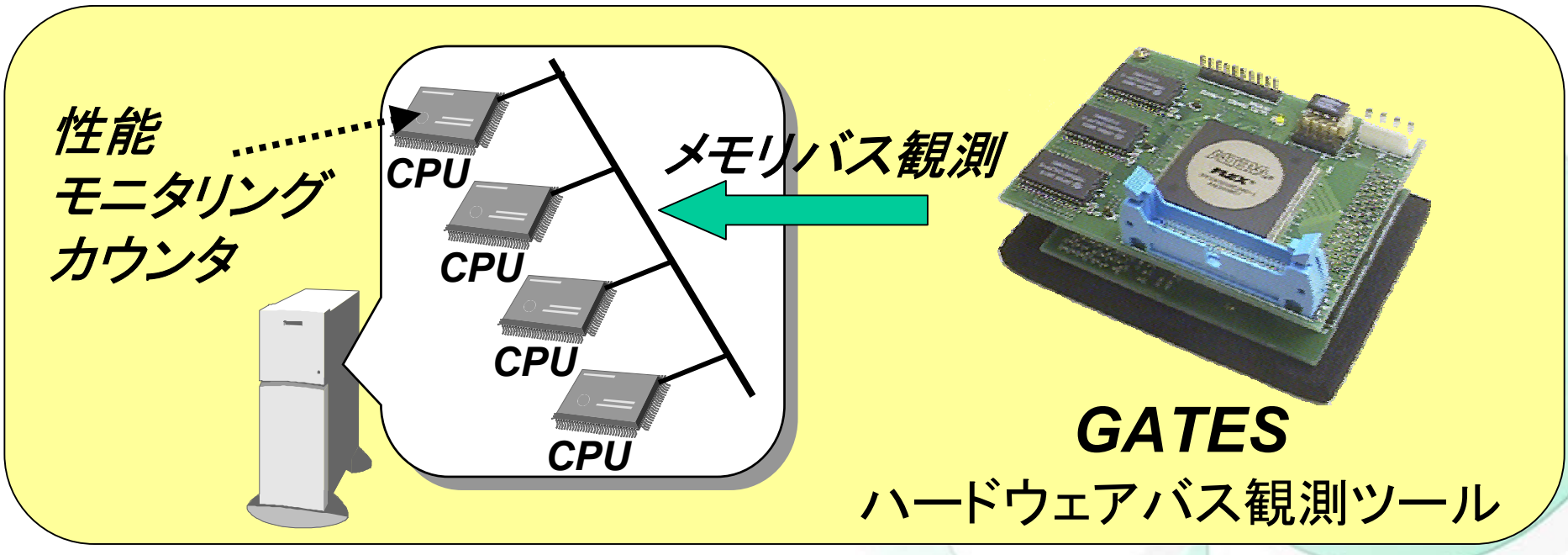
GATES

ハードウェアバス観測ツール

➢キャッシュミスが頻繁に発生するアドレスは？
➢False sharingされているデータは？

問題分析のアプローチ (2)

- メモリアーキテクチャの観点からの
  チューニングと定量的評価
  - カーネルプロファイリング
  - 共有メモリバスの観測
  - CPU内部で発生するイベントの計測



高負荷時における性能ボトルネック

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

カーネルプロファイリング

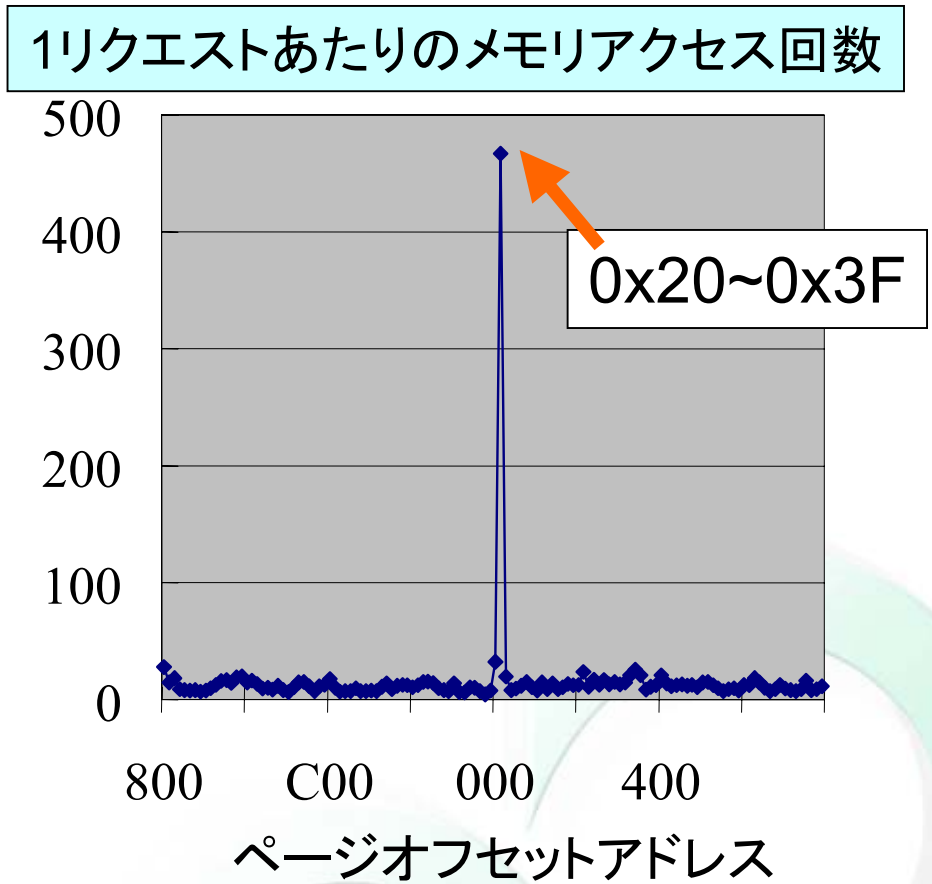
スケジューラ内部でOS実行時間の10%~20%を消費

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

イベント計測

キャッシュミス回数などを定量的に評価



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

問題解決へのアプローチ

- 高い負荷をLinuxシステムに与えたときの性能ボトルネックを解消
  - プロセススケジューラのチューニング

キャッシュカラーリングによるスケジューラ内部のキャッシュミス削減

- キャッシュカラーリングの評価
  - カラーリングの効果を定量的に分析

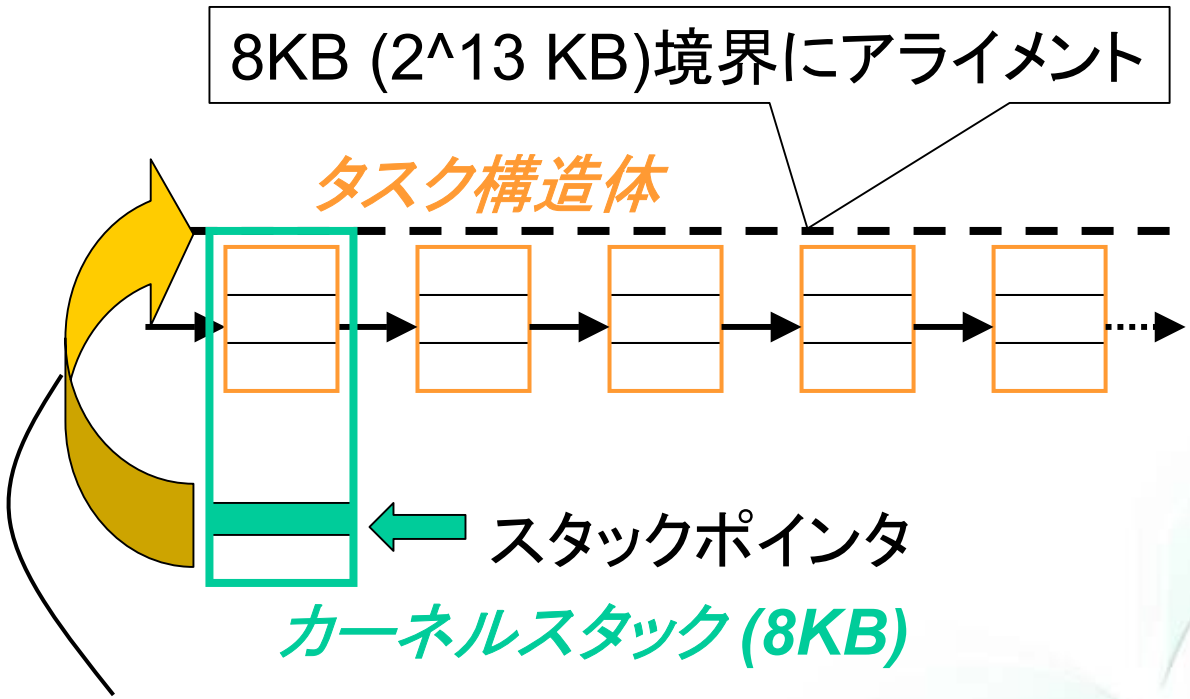
メモリバストラフィックやキャッシュミス率など．．．

内容

- 背景
- 性能ボトルネック
- **Linuxカーネルの問題点**
- タスク構造体のカラーリング
- 性能評価

スケジューラが使用するデータ構造

- **runqueueによりすべての実行可能プロセスを管理**
  - タスク構造体を要素とする双方向リスト構造
  - タスク構造体は各プロセスごとの情報を保持

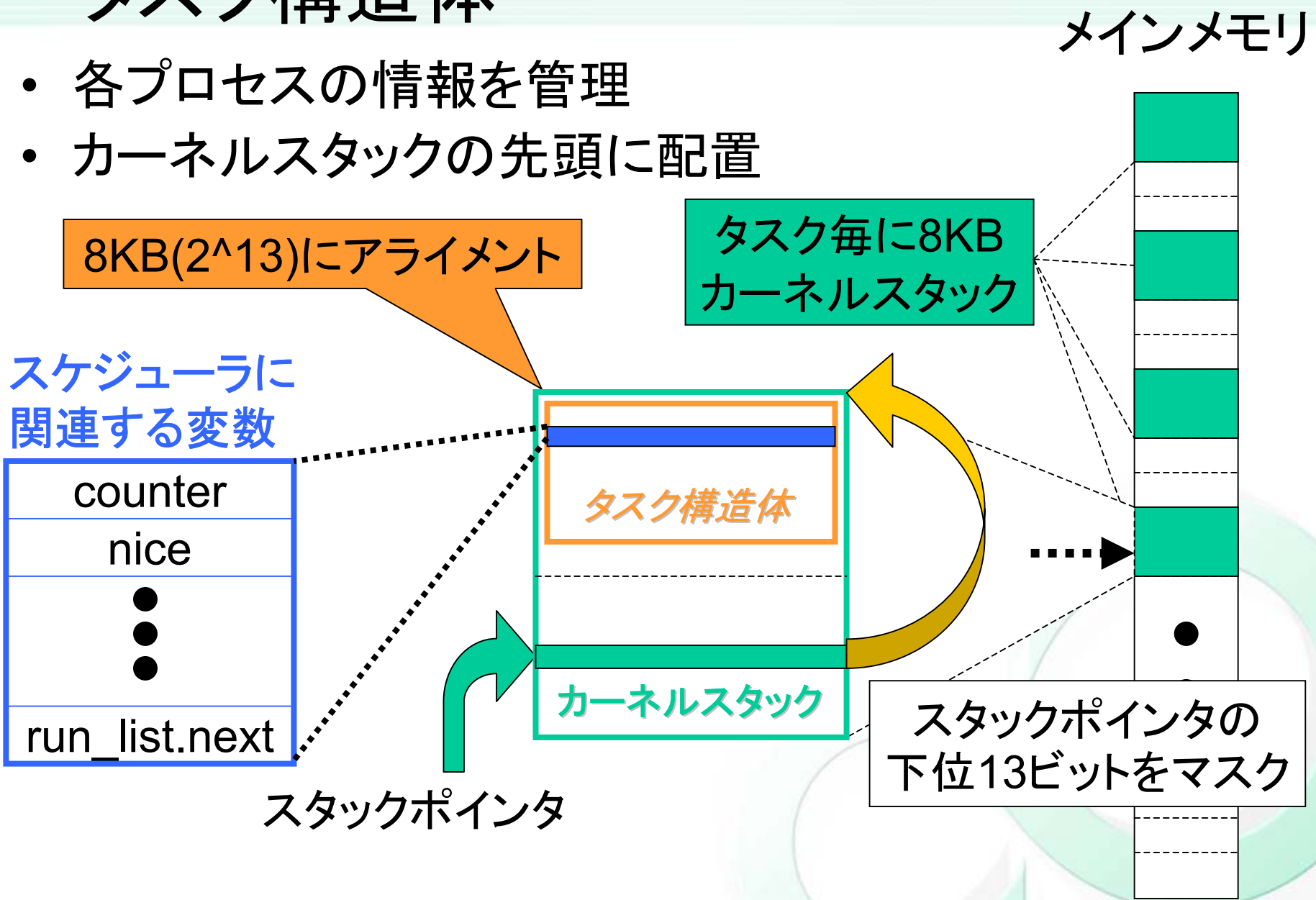


- ***CURRENTマクロによる高速アクセス***

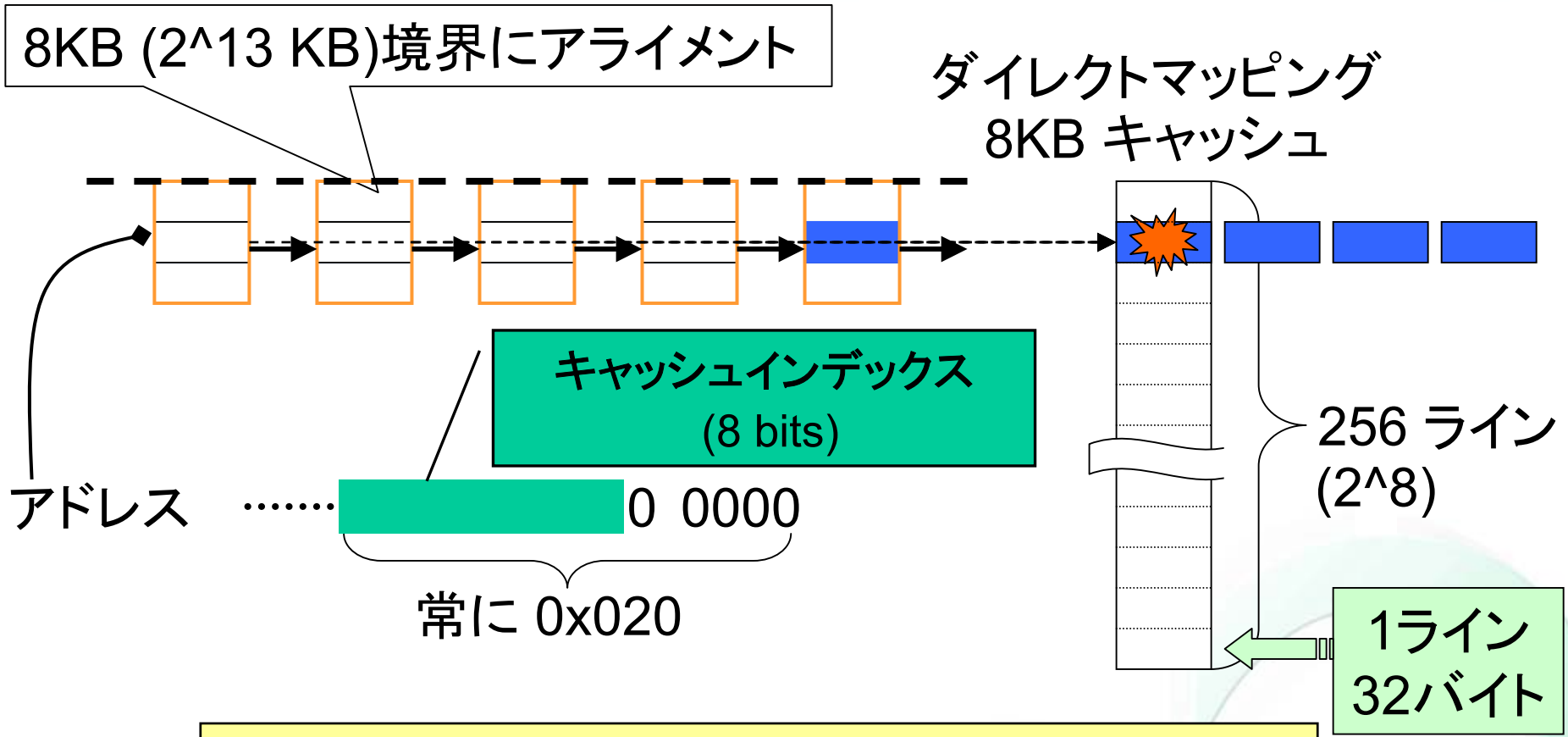
スタックポインタの下位13ビットをマスクするだけで
カーネルスタックの先頭アドレスを取得

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



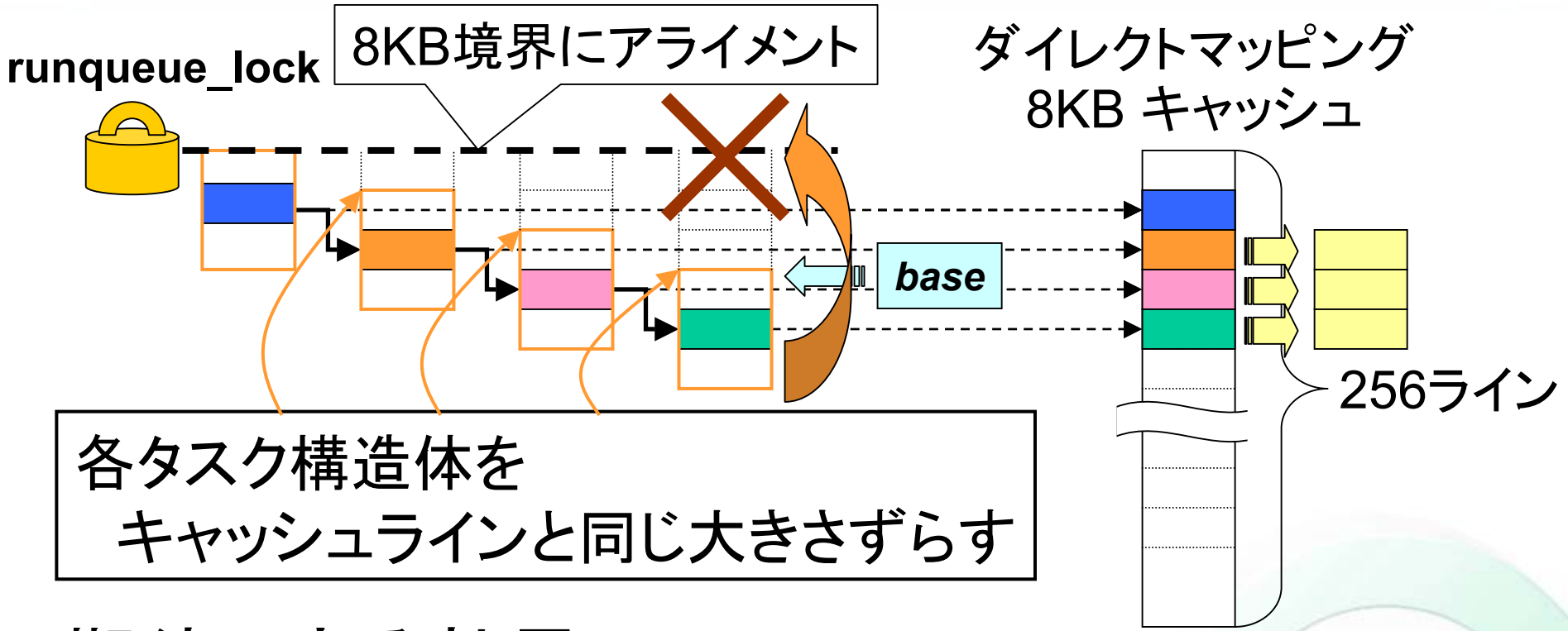
スケジューラ内部で発生するキャッシュライン競合



実測値：2次キャッシュミス率 > 90%
通常は，ミス率10～20%程度

ランキューの線形走査が多数のキャッシュミスを発生

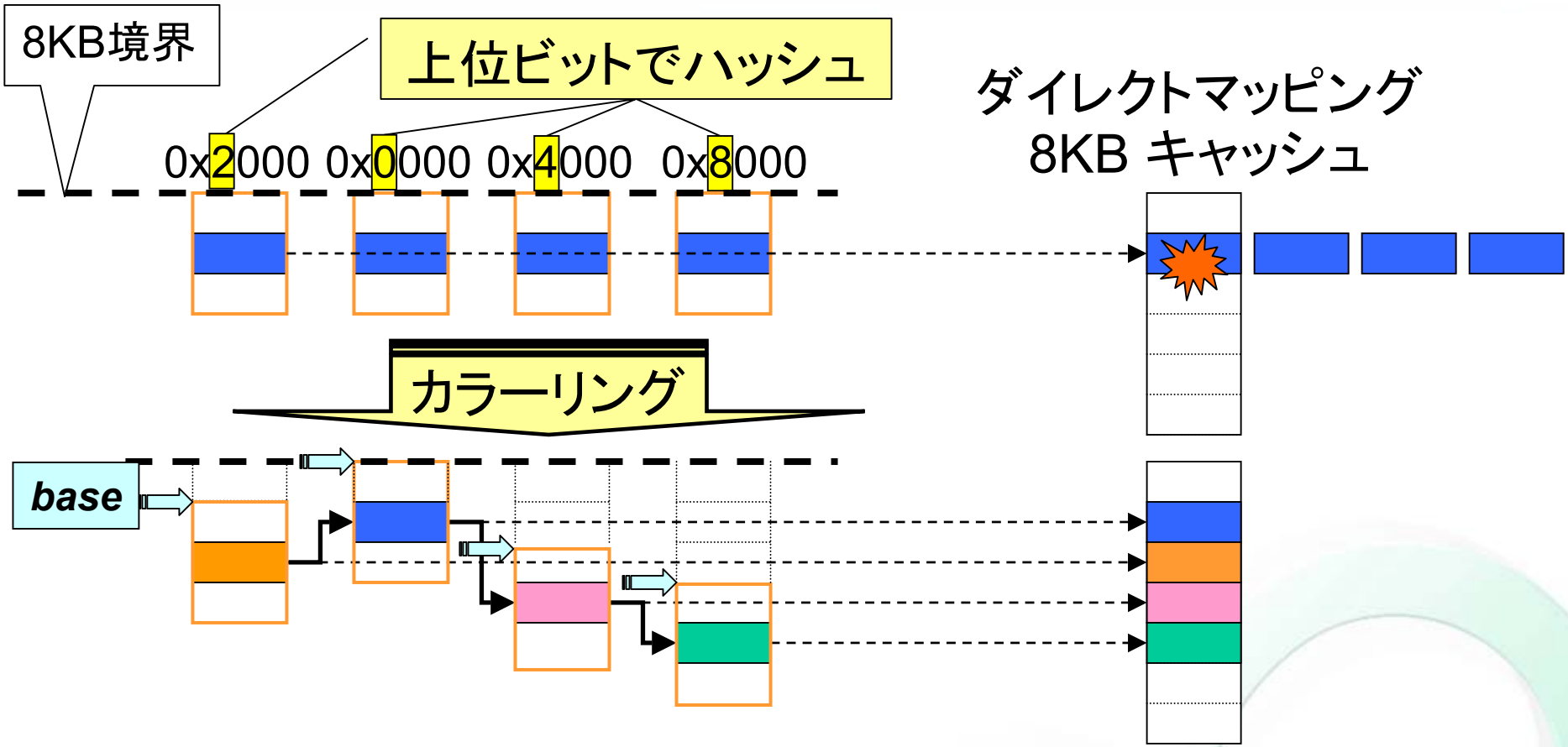
一般的な解決法 －キャッシュカラーリング－



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
- × カラーリングによって他の有用なデータが追い出される可能性がある.

キャッシュカラーリングの実装



修正したCURRENTマクロがハッシュされた先頭アドレスを返す

- 高速アクセス
- ~~タスク構造体は8KB境界への配置が必要~~

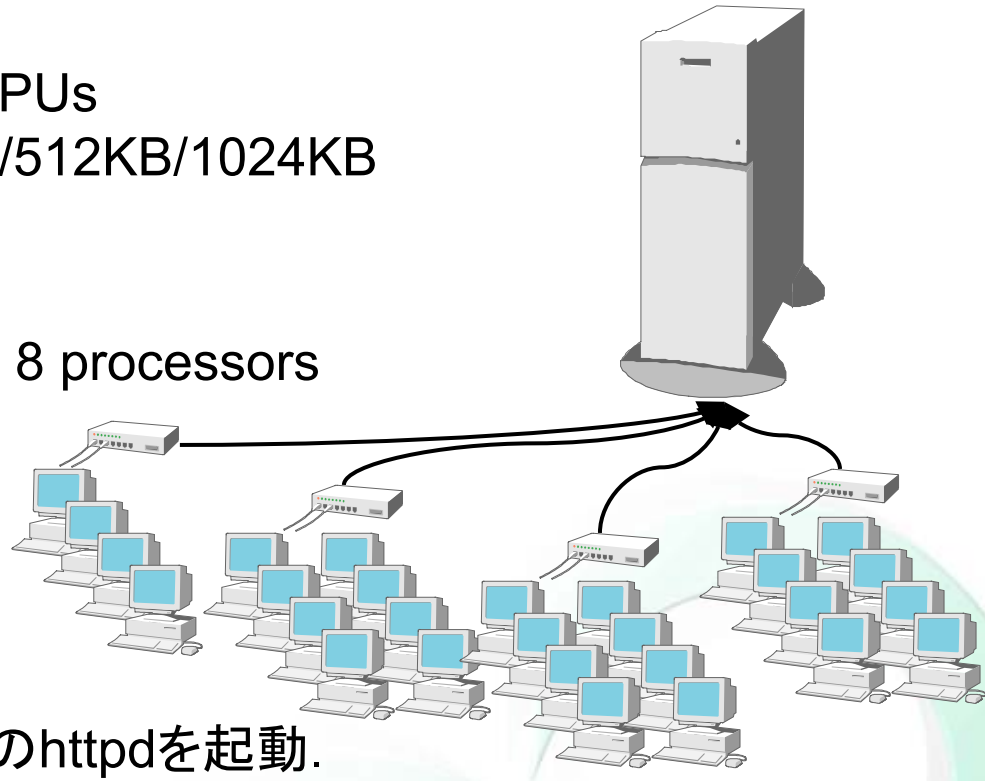
内容

- 背景
- 性能ボトルネック
- Linuxカーネルの問題点
- タスク構造体のカラーリング
- 性能評価

評価環境

システム構成

- メモリシステム評価
  - Pentium Pro 200MHz x 4 CPUs
  - 2次キャッシュサイズ: 256KB/512KB/1024KB
  - カラーリング数: 32
- スケーラビリティ評価
  - Pentium III Xeon 550MHz x 8 processors
  - 2次キャッシュサイズ: 1 MB
  - カラーリング数: 32



アプリケーション

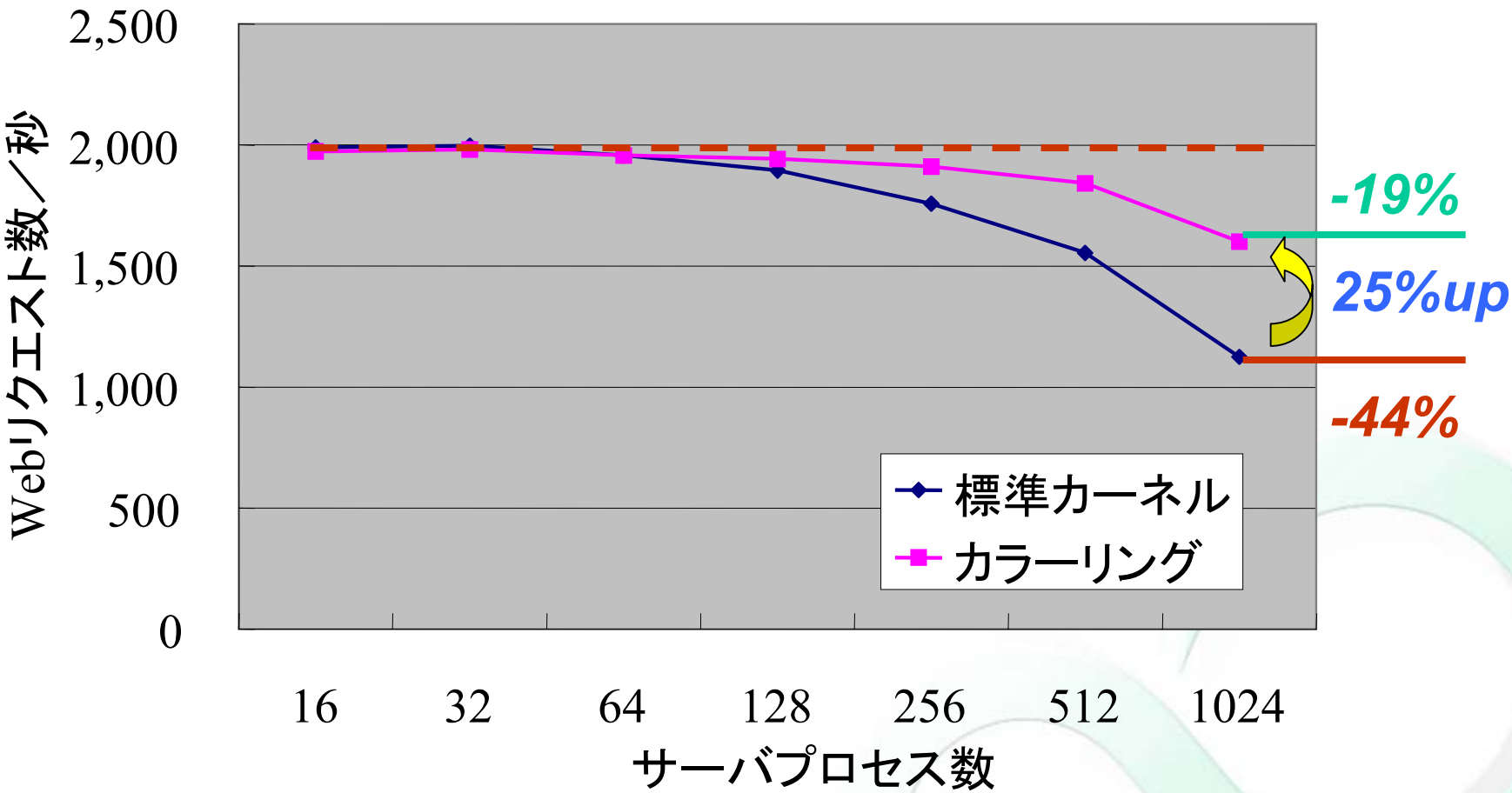
- WebBench 3.0

  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).

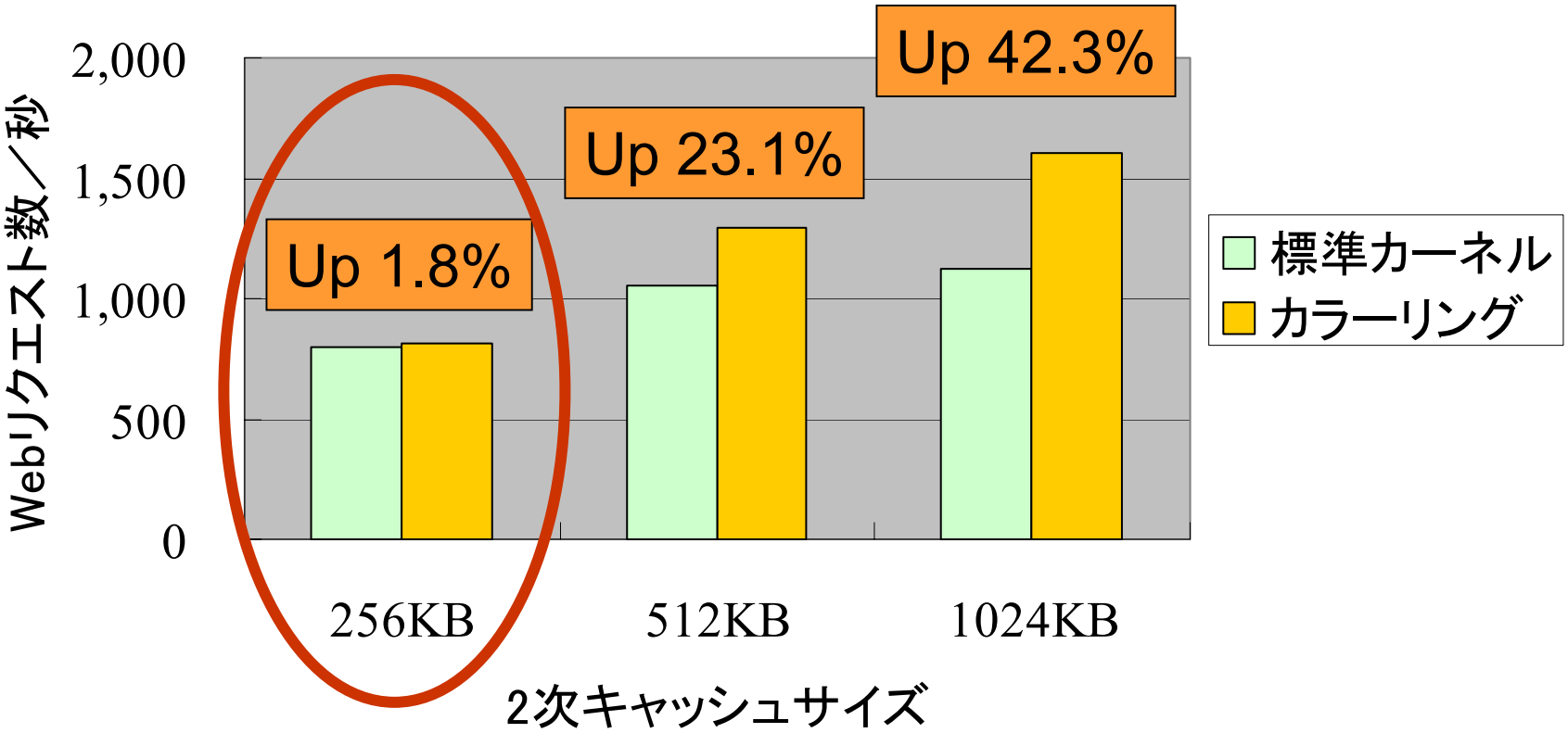
WebBenchにおける性能向上

Pentium Pro 200MHz x 4 CPUs/2次キャッシュ 1024KB/カーネル 2.4.4



高負荷な状況下で，25%の性能向上

キャッシュサイズを変化させたときのカラーリングの効果

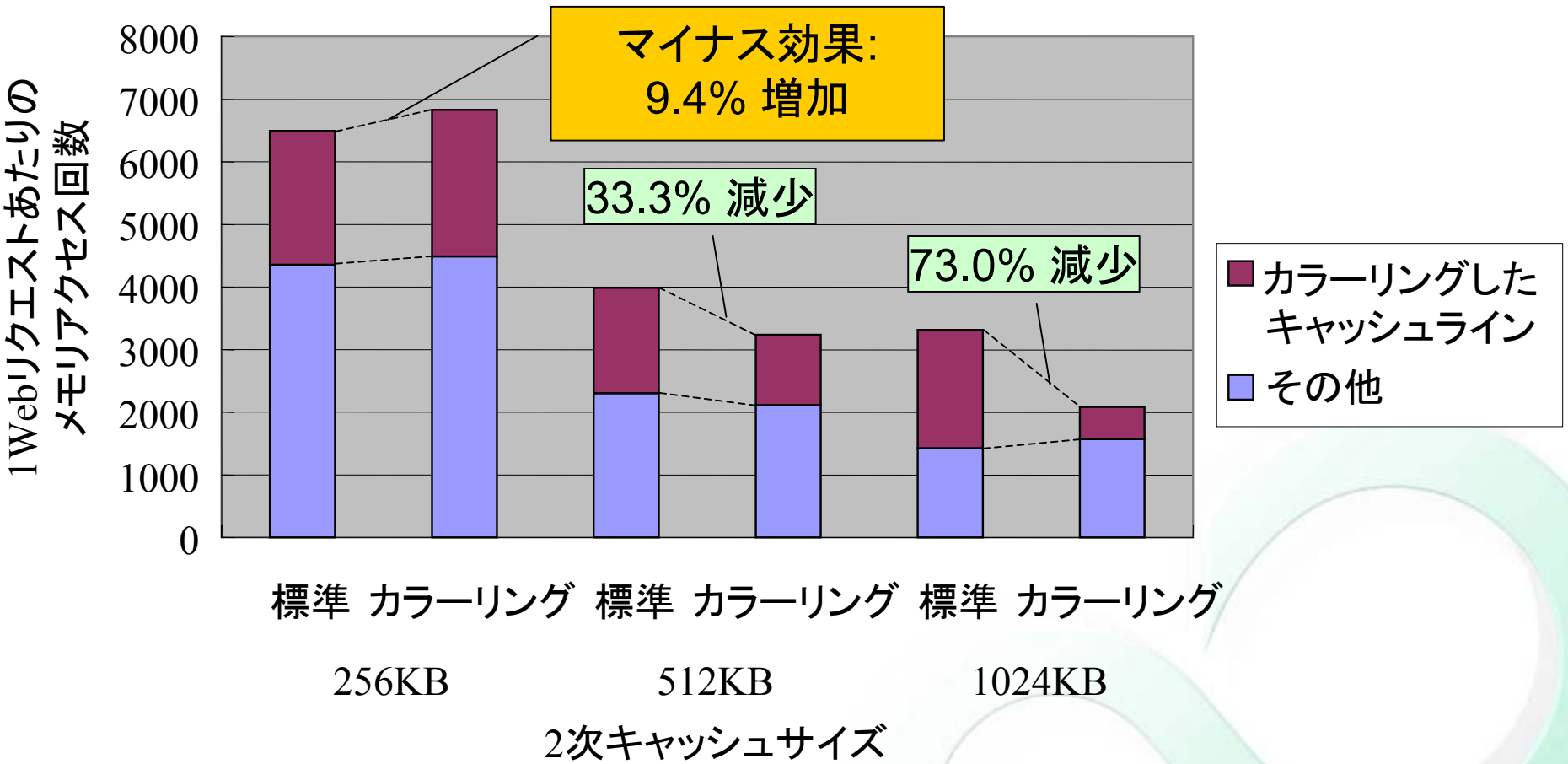


キャッシュサイズを変化させた場合の性能向上
(4-way Pentium Pro 200 MHz上で1024個のhttpdが走行している場合)

- 最大 42.3% の性能向上
- キャッシュサイズが大きいほどカラーリングの効果大

GATESによるメモリバストランザクションの分析

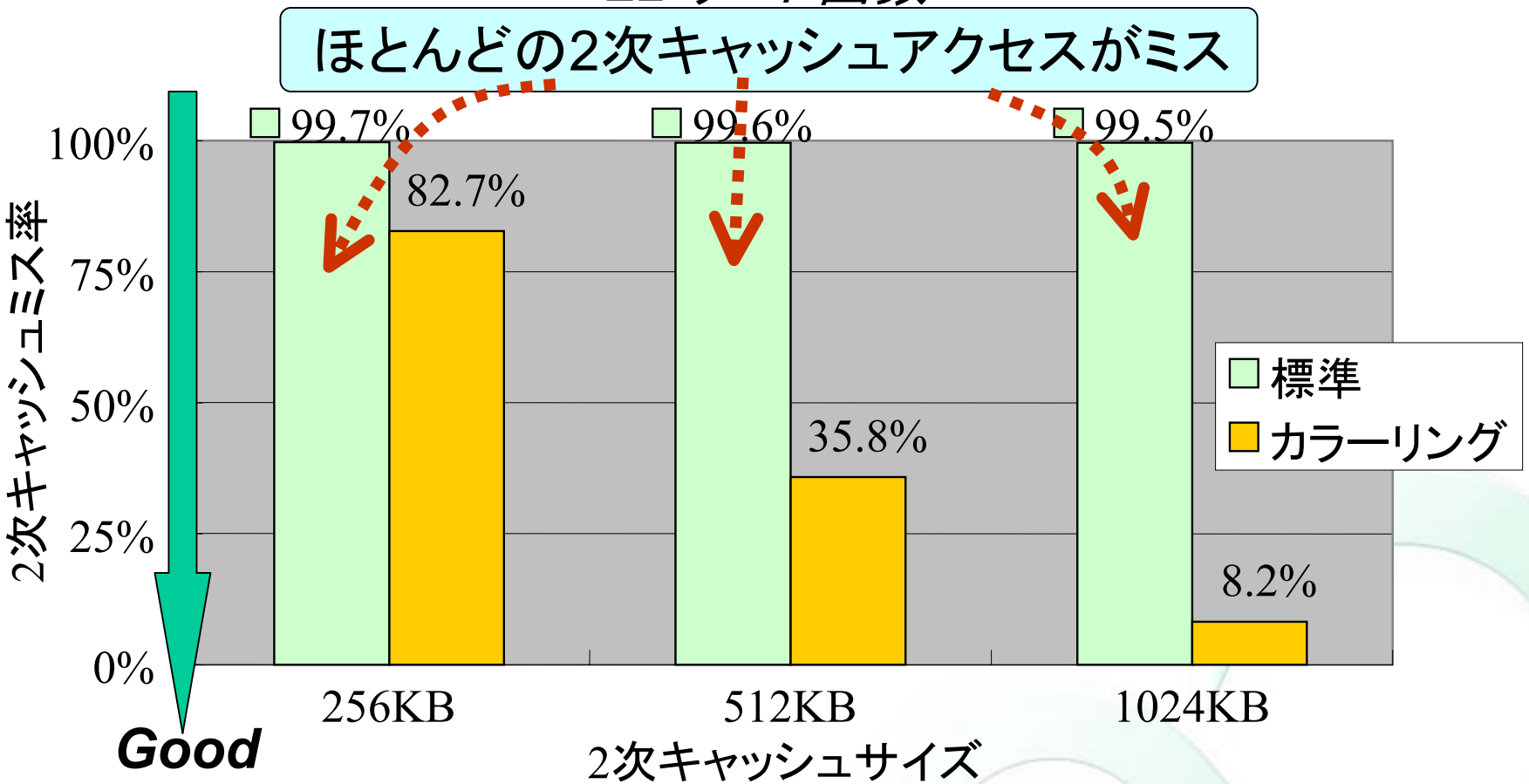
カラーリングされたキャッシュラインに関するメモリアクセス回数



キャッシュサイズが小さい場合にはカラーリングは逆効果

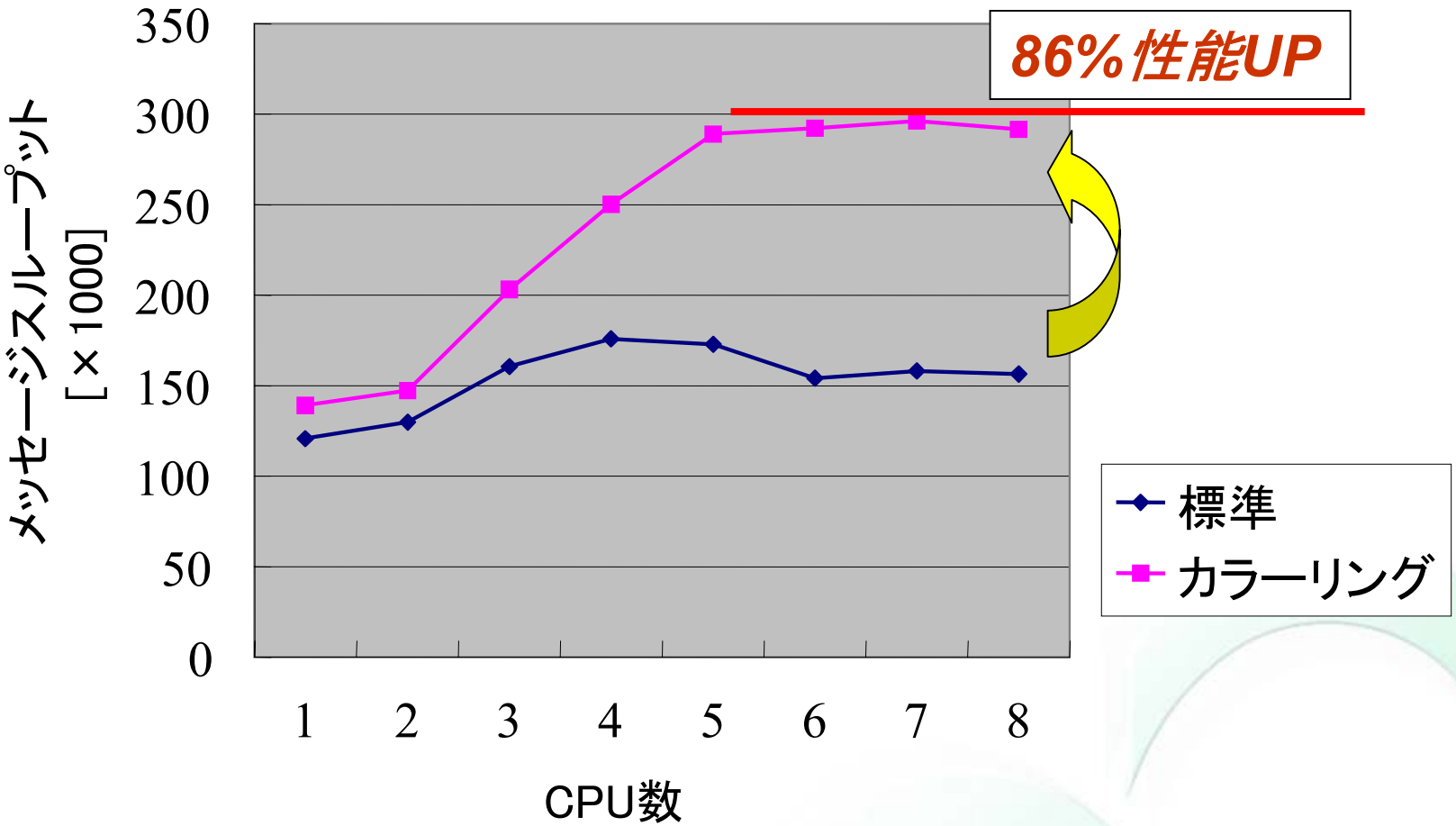
runqueue走査中の2次キャッシュミス率の減少

$$2\text{次キャッシュミス率} = \frac{L2\ \text{ミス回数}}{L2\ \text{リード回数}}$$



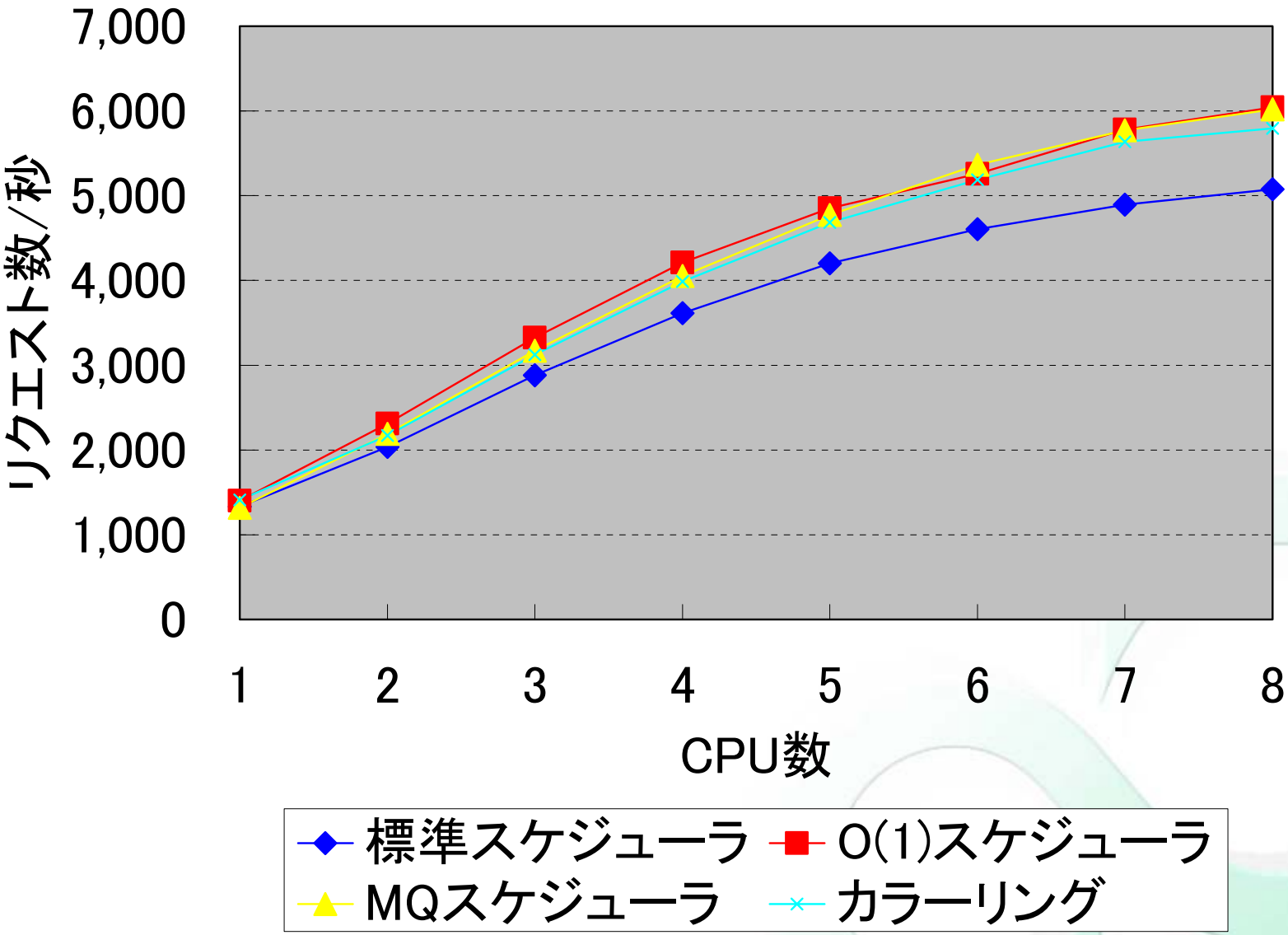
カラーリングにより，runqueue走査時の
2次キャッシュミスが大幅に減少（最大91.3%）

Chatベンチマークにおけるスケーラビリティの向上

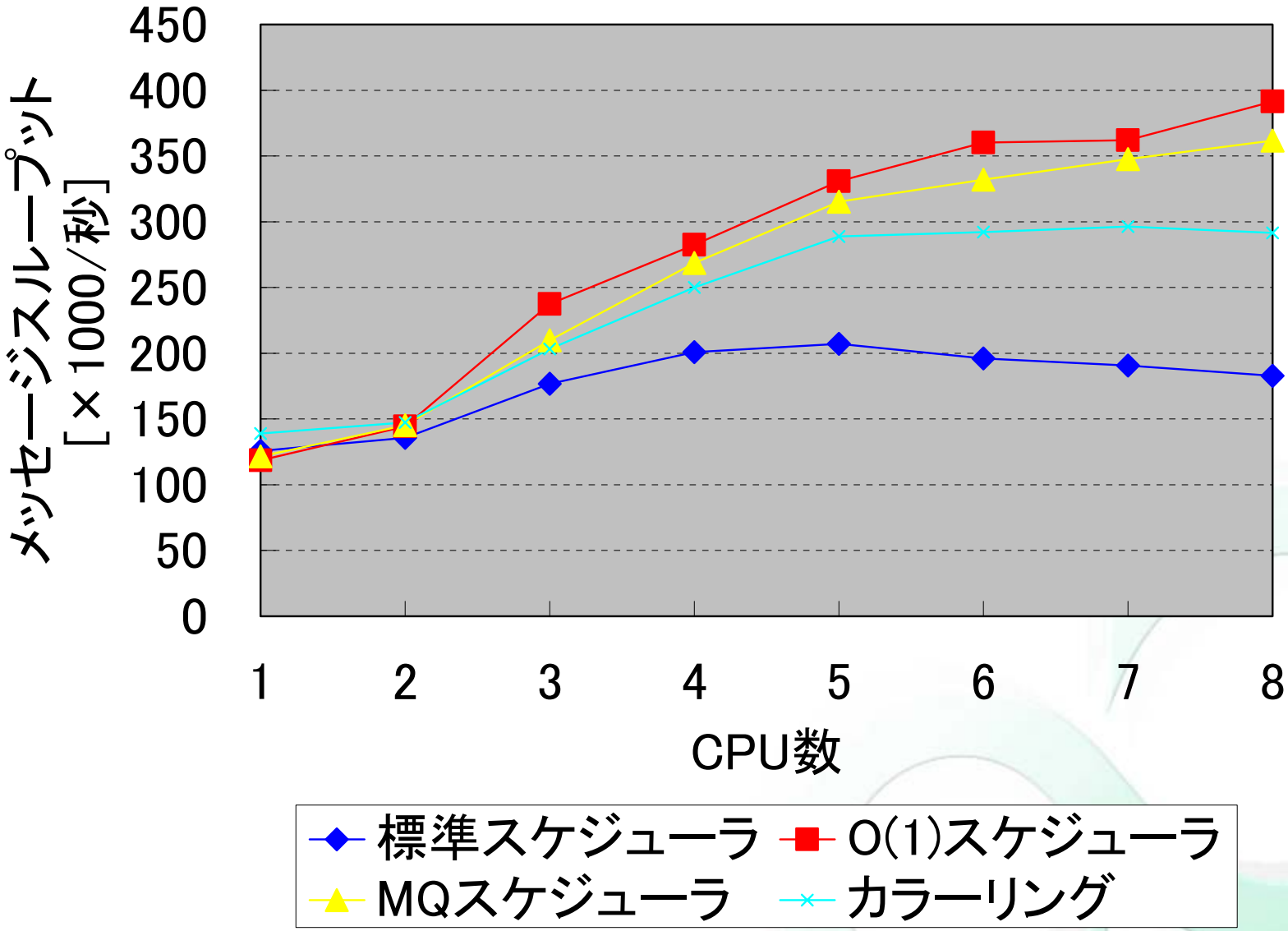


- 8CPUの場合，最大で86.3%の性能向上
- ロック競合率が減少　[86% ⇒ 70%]

他スケジューラとの比較（WebBench）



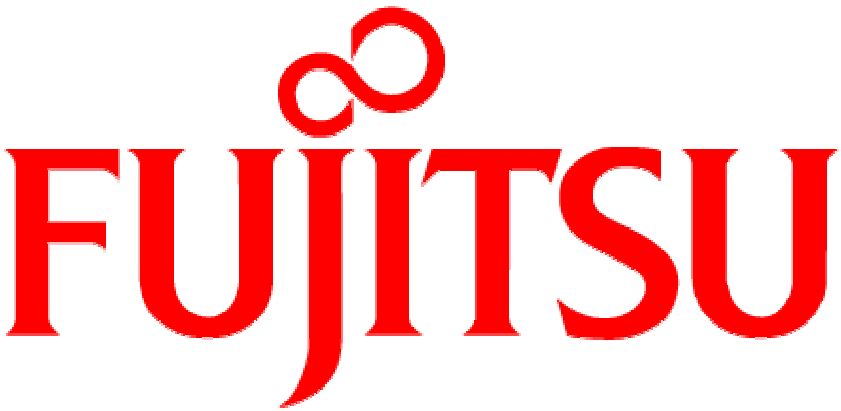
他スケジューラとの比較（Chat）



まとめ

- キャッシュミスの削減によりスケジューラの性能を改善
  - Linuxシステムの高負荷時における性能の改善
  - スケーラビリティの向上
- カラーリングの効果
  - キャッシュサイズが大きい場合に，より効果を発揮
  - キャッシュサイズが小さいと逆効果な場合もある
- 今後の予定
  - 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

富士通研究所 「Linuxカーネルに関する情報」
http://www.labs.fujitsu.com/techinfo/linux



THE POSSIBILITIES ARE INFINITE