カーネル2．6の実力を探る

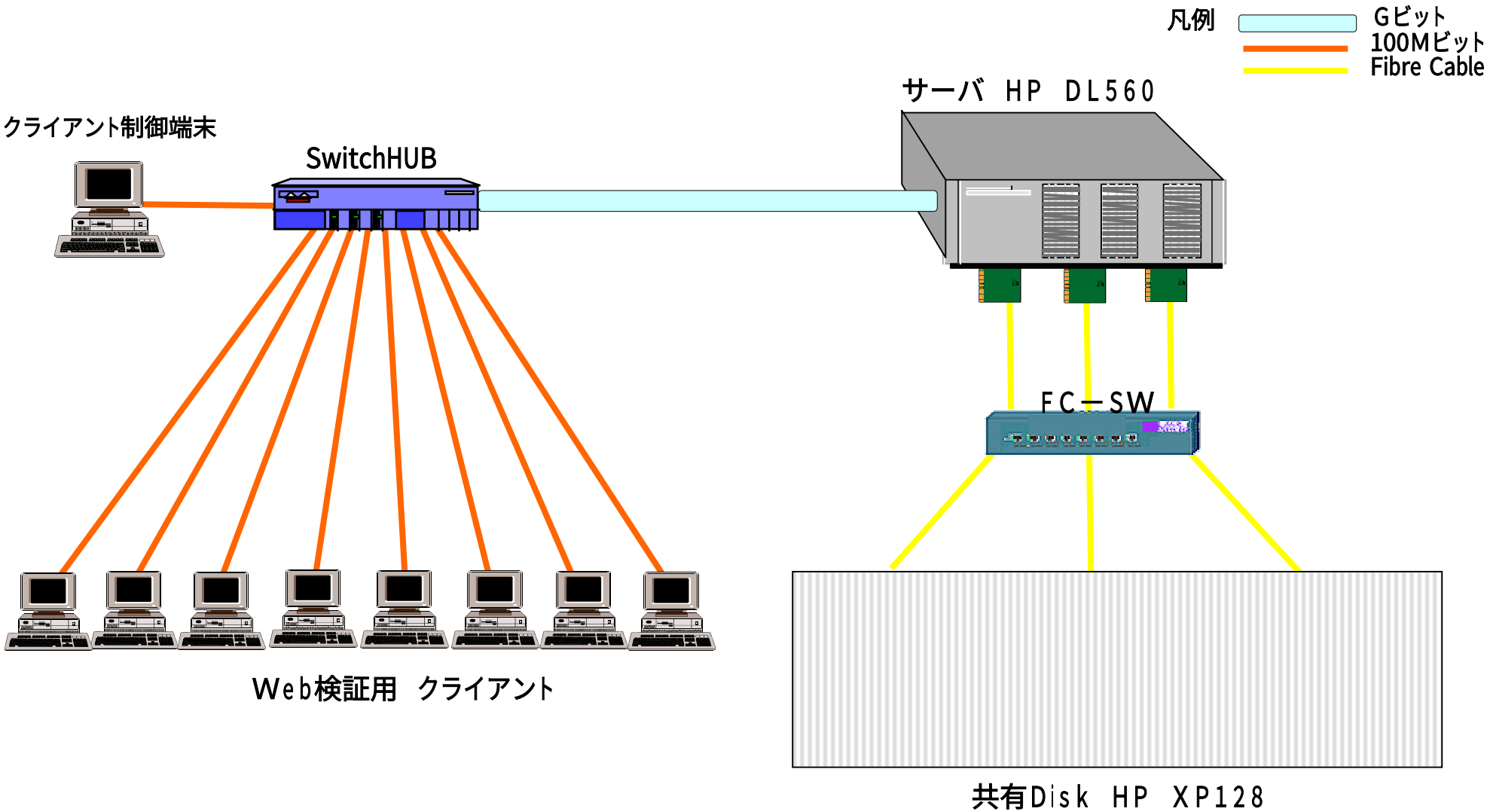
NTTコムウェア株式会社
Linuxセンタ 佐々木博正

測定協力者　大越、本多

目次

- ファイルI／Oの実力を探る
  ファイルI／Oベンチマークによる検証
- スレッドの実力を探る
  JavaVMベンチマークによる検証
- Webサーバ基盤の実力を探る
  (epollの実力を探る)
  Webサーバベンチマークによる検証

検証環境： システム構成



検証環境： ハードウェア

サーバ

製品名	HP ProLiant DL560
CPU	Xeon 2.0GHz × 4
メモリ	1GB （注1）
ネットワーク	NC7781 PCI-X Gigabit NIC × 2

（注1）実メモリを4GB搭載していたが、ファイルI/O検証でのキャッシュの影響を少なくするため、kernelをHighMem未使用（トータル1GB）にて検証

FC－SW

製品名	SilkWorm 3800 Enterprise Fabric Switch
ポート数	16
転送能力	2Gbit/s（全二重）

HBA

製品名	QLA2300F
I/F	64-bit 66MHz PCI
速度	最大2Gb Fiber Channel Adapter

共有DISK

製品名	HP XP128
内部データ転送方式	7.5GB/秒 クロスバー・スイッチ
キャッシュメモリ	最大32GB（本検証では12GB）
アレイ・コントロール・プロセッサ（ACP）	1～2 プロセッサ・ペア（本検証では2）
アレイへのバス	100MB/秒FC-AL×8(ACPペアあたり)
アレイの構成	1～31グループ、4ディスク/1グループ （本検証では 3グループを使用）
DISK容量	最大18TB(本検証では 8TB/1ディスク)
RAID構成	RAID 1
クライアント/ホスト・インタフェース・プロセッサ	1～3プロセッサ（本検証では2）

検証環境：ソフトウェア

Kernel 2.4

ディストリビューション	RedHat 7.3（FTP版)
Kernelバージョン	Ｋｅｒｎｅｌ 2.4.22（GCC 2.95でコンパイル)
Qlogicドライバ	qla2x00-v6.01.00
GCC	2.96
GLIBC	2.2.5
ｂｉｎｕｔｉｌｓ	2.11.93.0.2
ファイルシステム	EXT2

Kernel 2.6

ディストリビューション	RedHat 7.3（FTP版)
Kernelバージョン	Ｋｅｒｎｅｌ 2.6.0-test5（GCC 2.95でコンパイル)
Qlogicドライバ	qla2xxx-v8.00.00b5
GCC	2.96 / 3.3.1 (スレッド検証時)
GLIBC	2.2.5 / 2.3-2003.09.15版（スレッド検証時)
ｂｉｎｕｔｉｌｓ	2.11.93.0.2 / 2003.09.15版（スレッド検証時)
NPTL	0.58（スレッド検証時)
ファイルシステム	EXT2

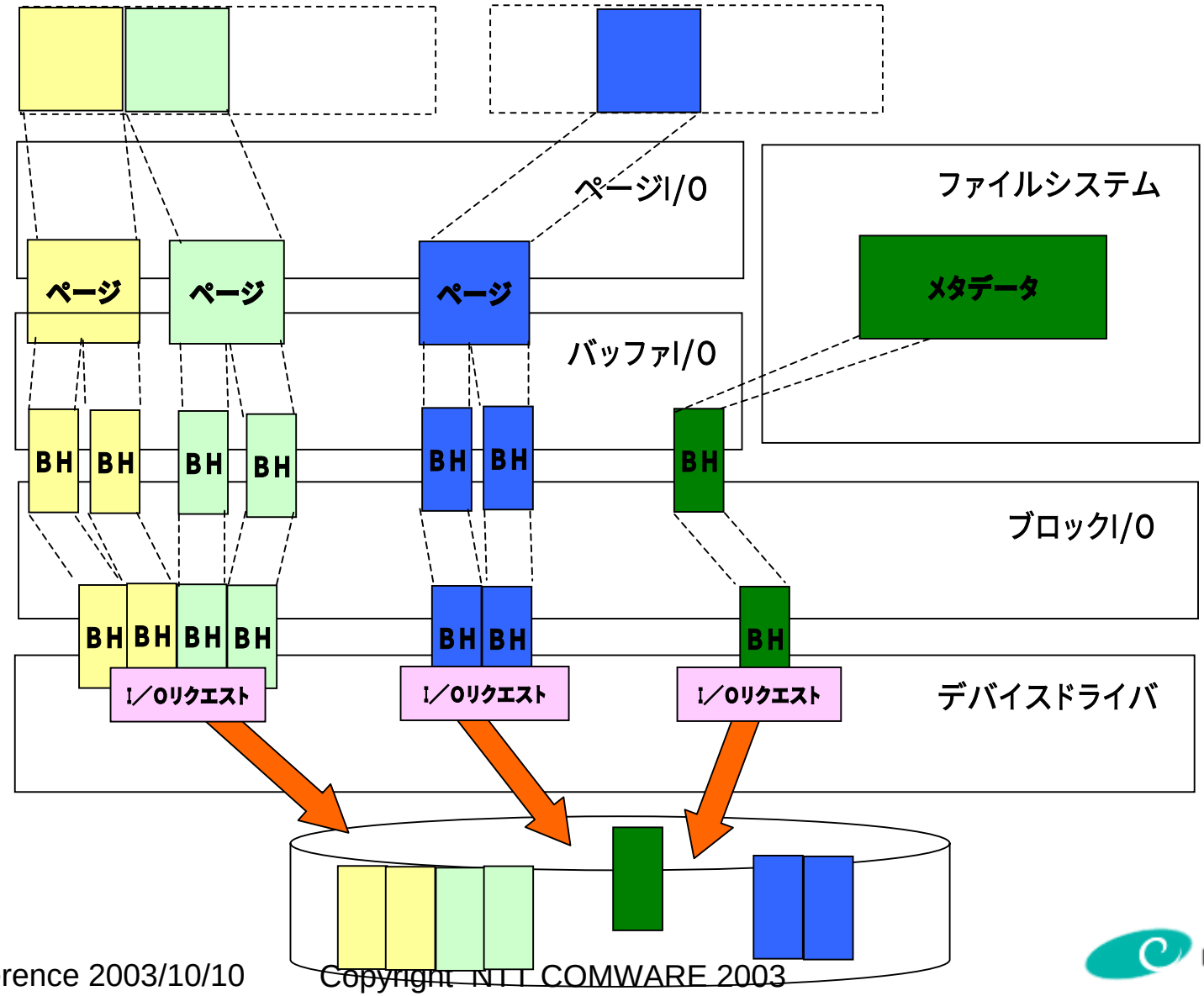
ファイルI／Oの実力を探る

― ファイルI／Oベンチマークによる検証 ―

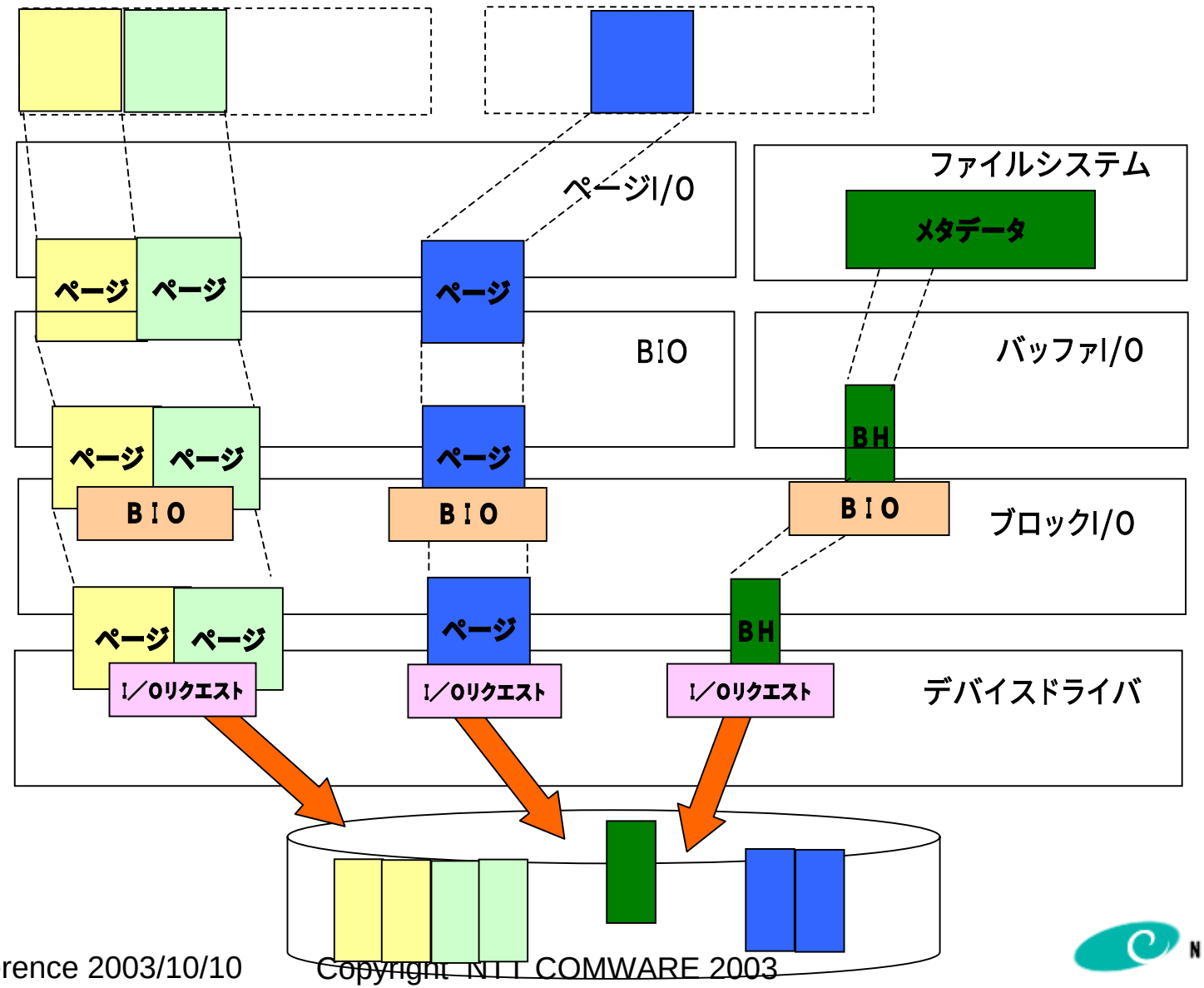
効率的になったファイルI／O

- ページ単位でクラスタ化可能なファイルI／O
  (マルチページI／O)
- ディスクヘッドの移動量を少なくするDirtyなキャッシュの管理方法

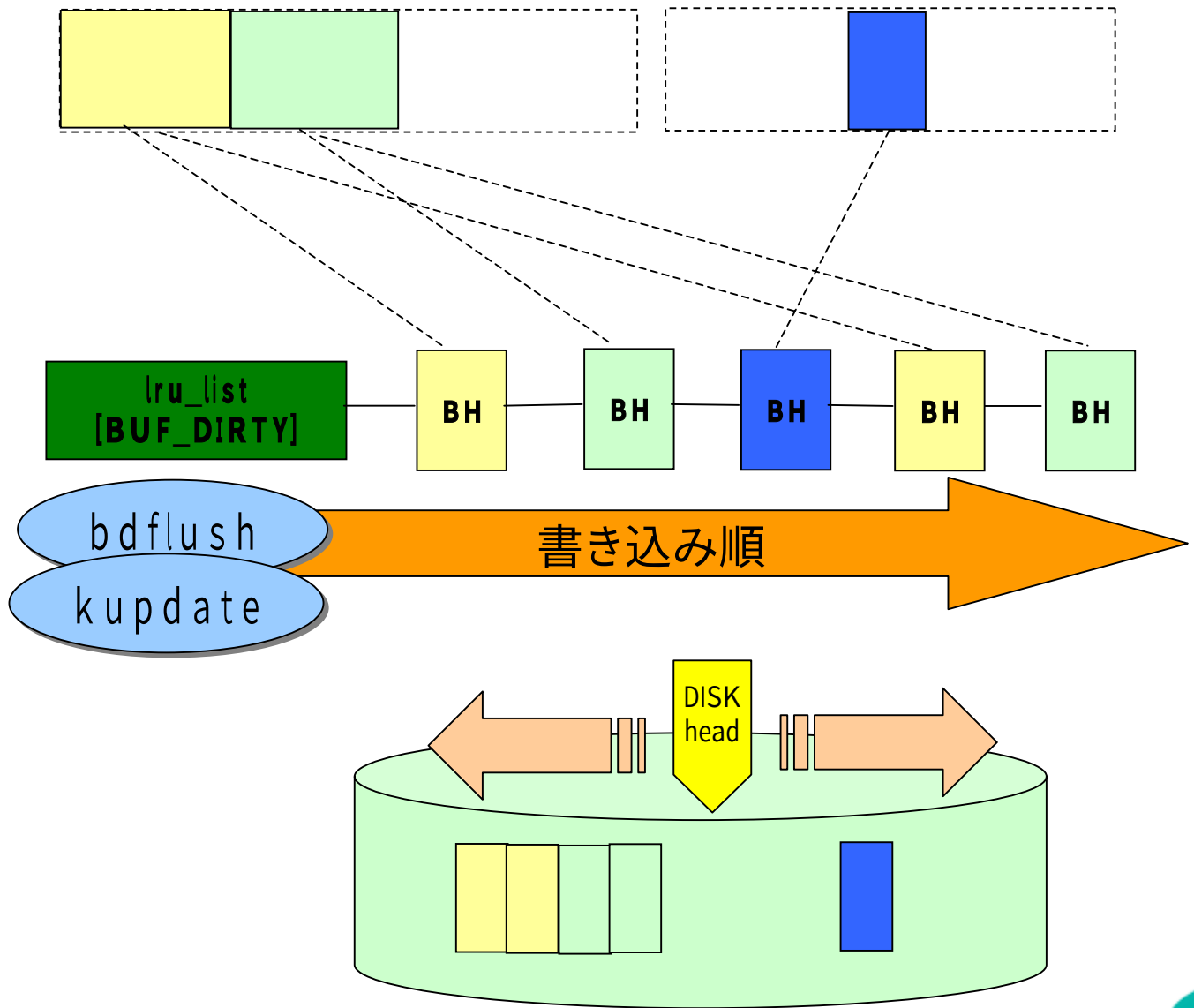
カーネル2．4のファイルI／O



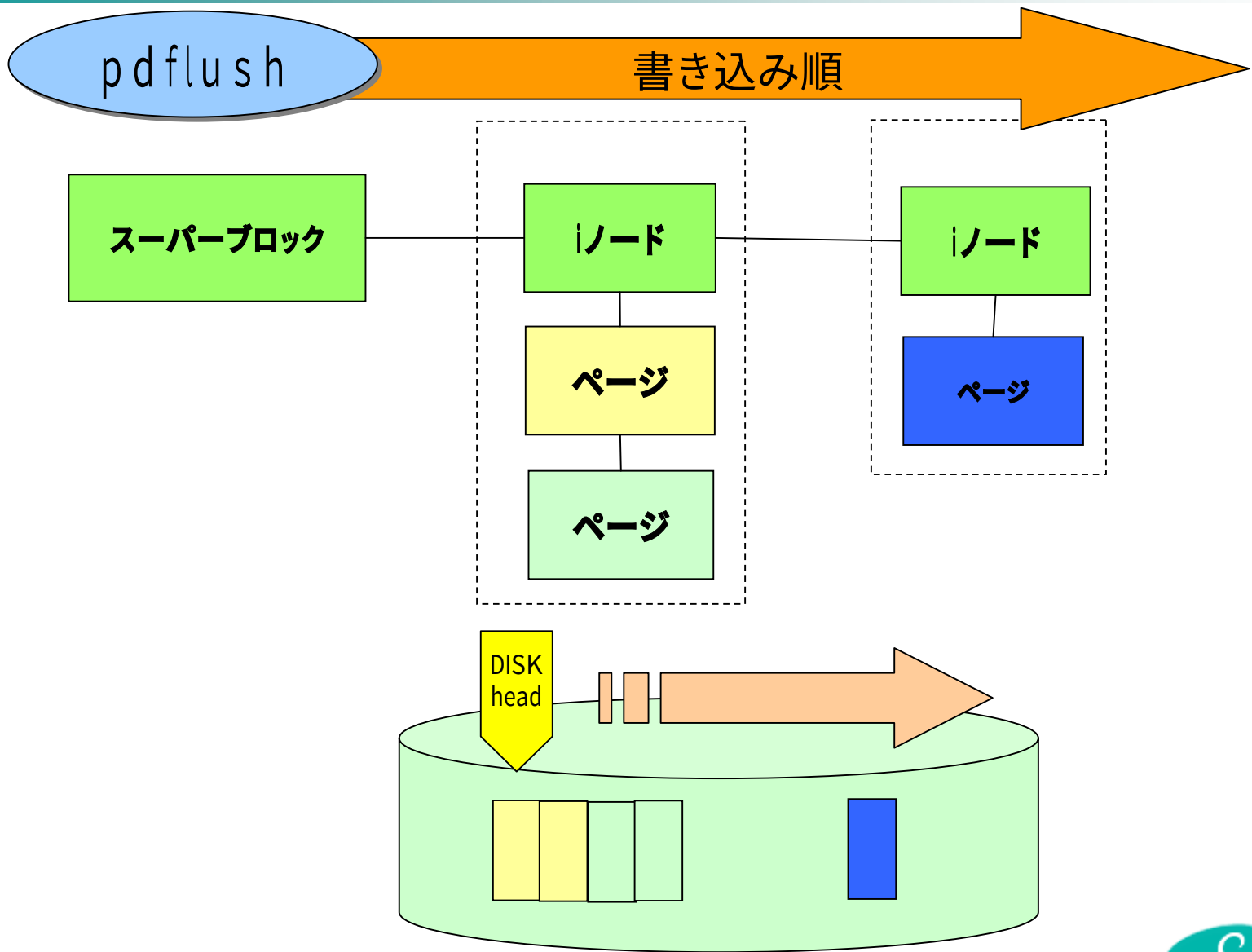
カーネル2．6のファイルI／O



カーネル2．4の遅延書き込み



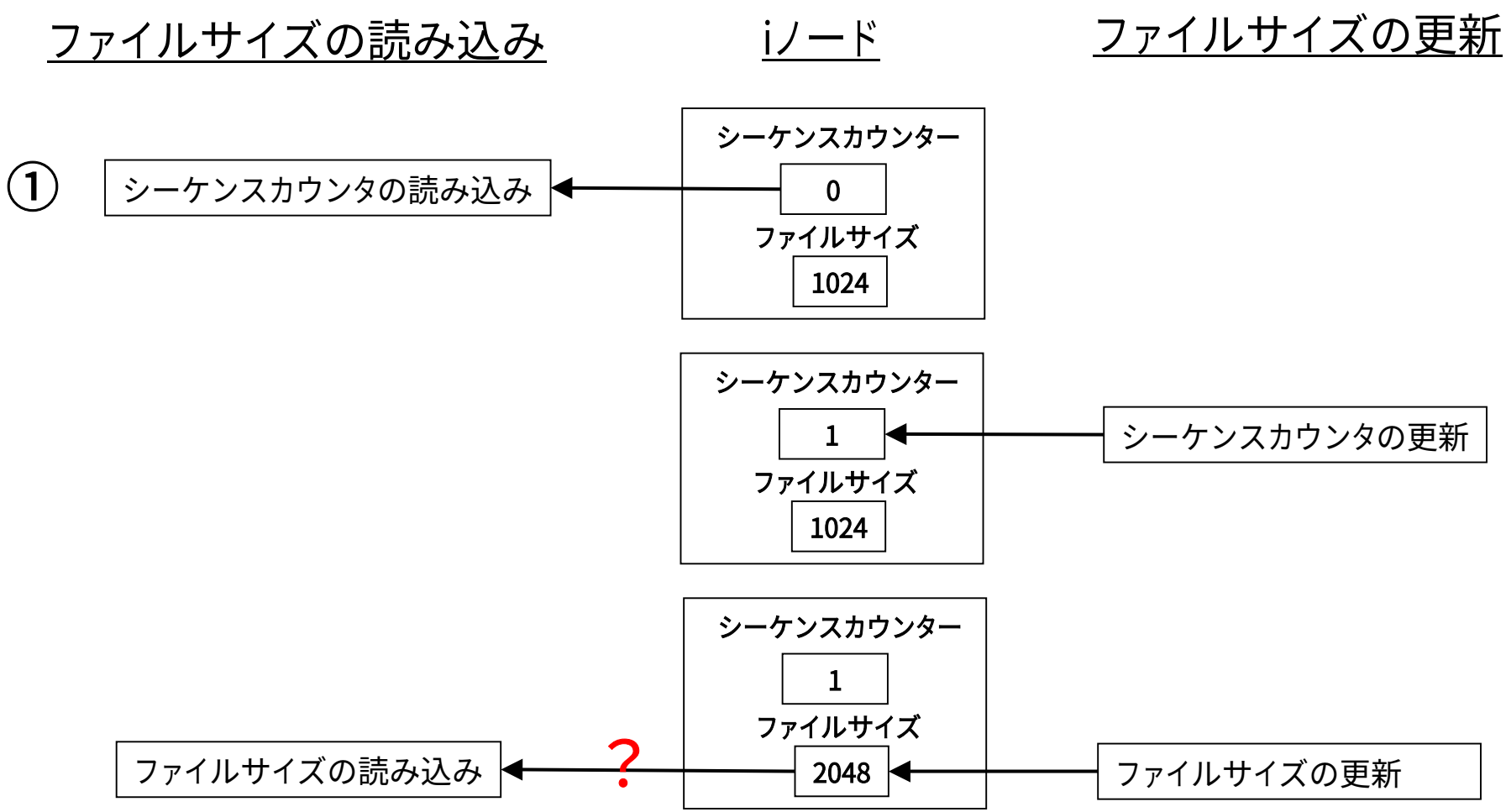
カーネル2．6の遅延書き込み



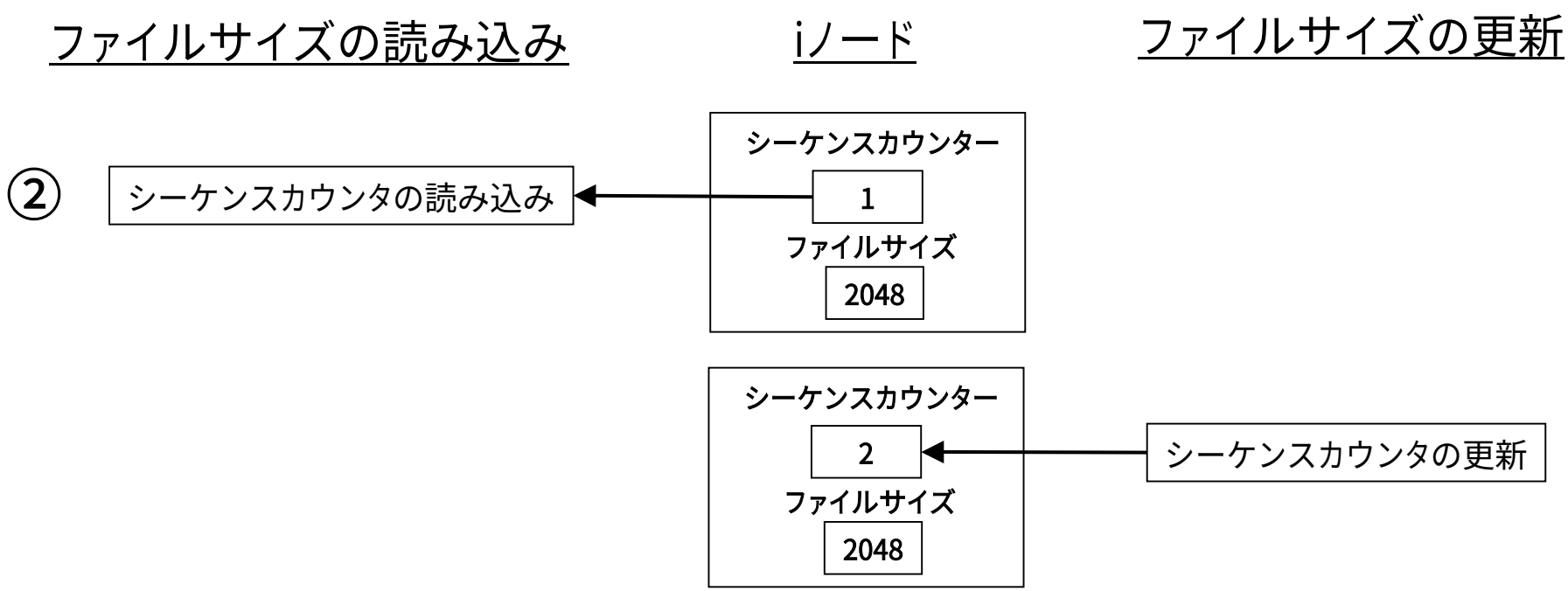
マルチプロセッサ構成における
並列実行度が向上したファイルI／O

- スピンロックに変わる新たな排他機構の導入により、並列実行度が向上
  - RCU（Read－Copy Update）
  - Sequence Counter
- ファイルシステム、ブロックI／Oレイヤにおけるロック粒度の改善により、並列実行度が向上
- デバイスドライバのI／O完了割り込み遅延処理が、BHハンドラから、CPU毎に並列実行可能なソフトウェア割り込みに変更

SequenceCounterを用いたファイルサイズの更新 (1)



SequenceCounterを用いたファイルサイズの更新 (2)

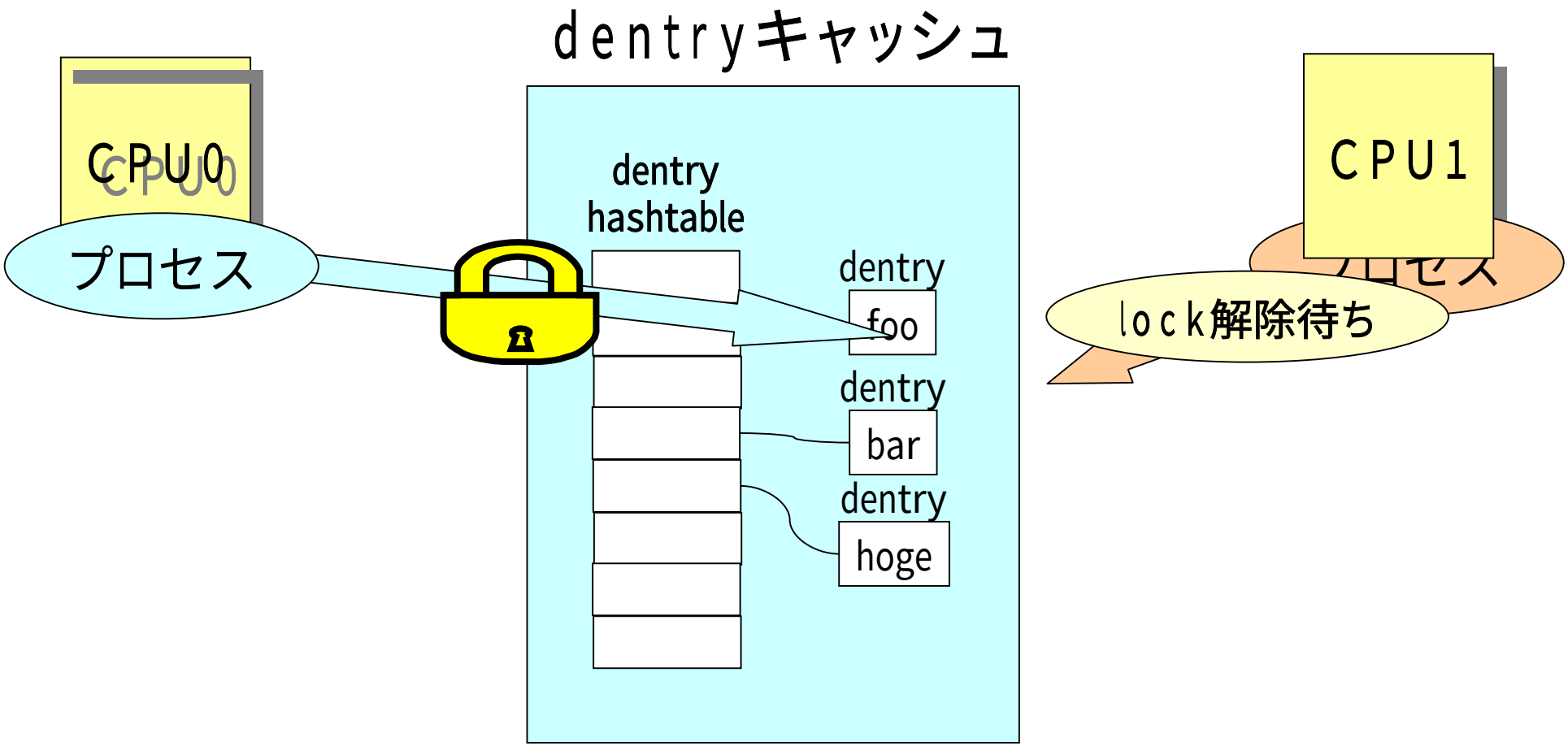


③

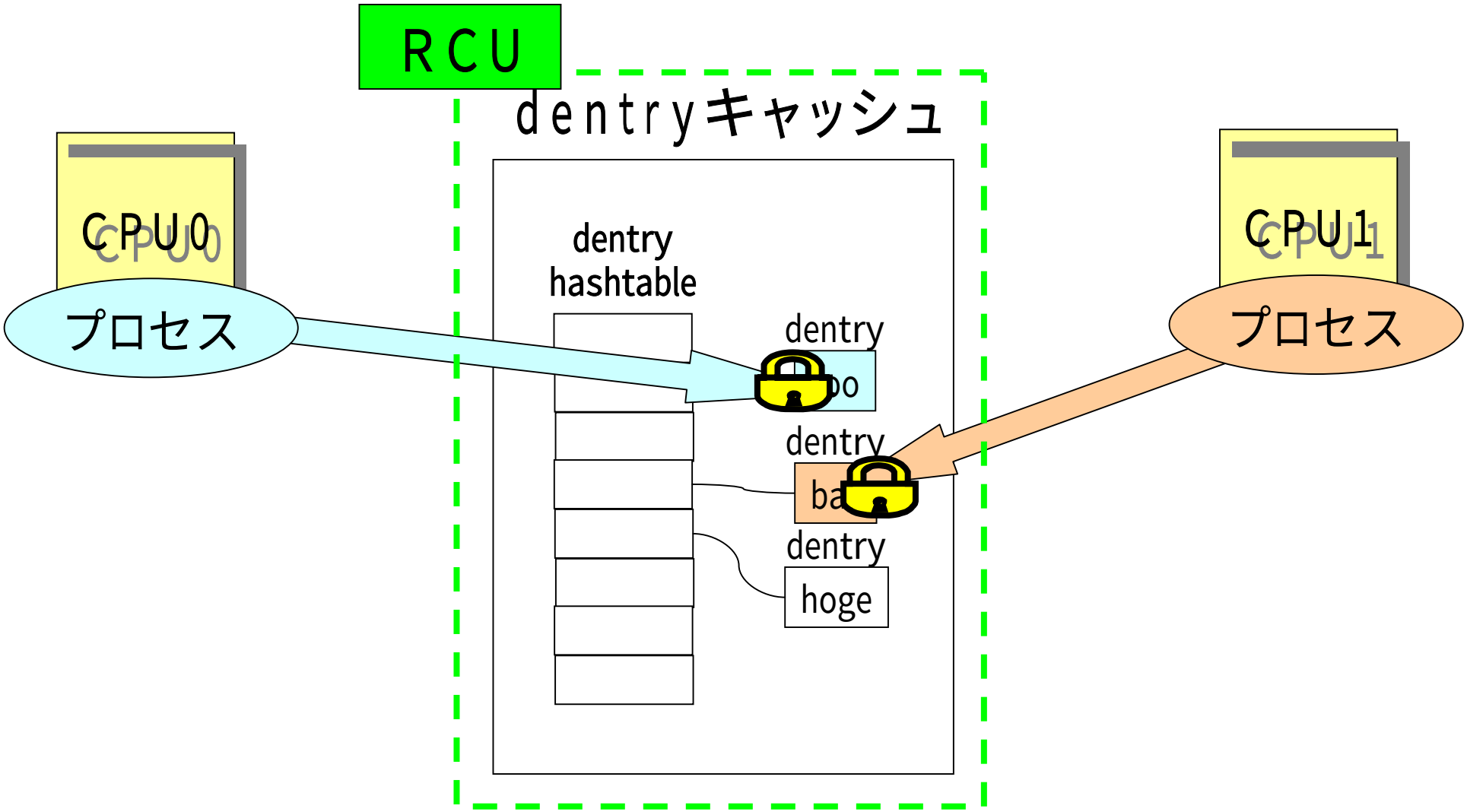
①で読み込んだシーケンス番号が奇数
→ 更新処理が既に開始中であったため、①からやり直し
①と②で読み込んだシーケンス番号が異なる
→ 読み込み処理中に更新されたため、①からやり直し

ただし、書き込み同士で排他を行う場合は spinlockが必要

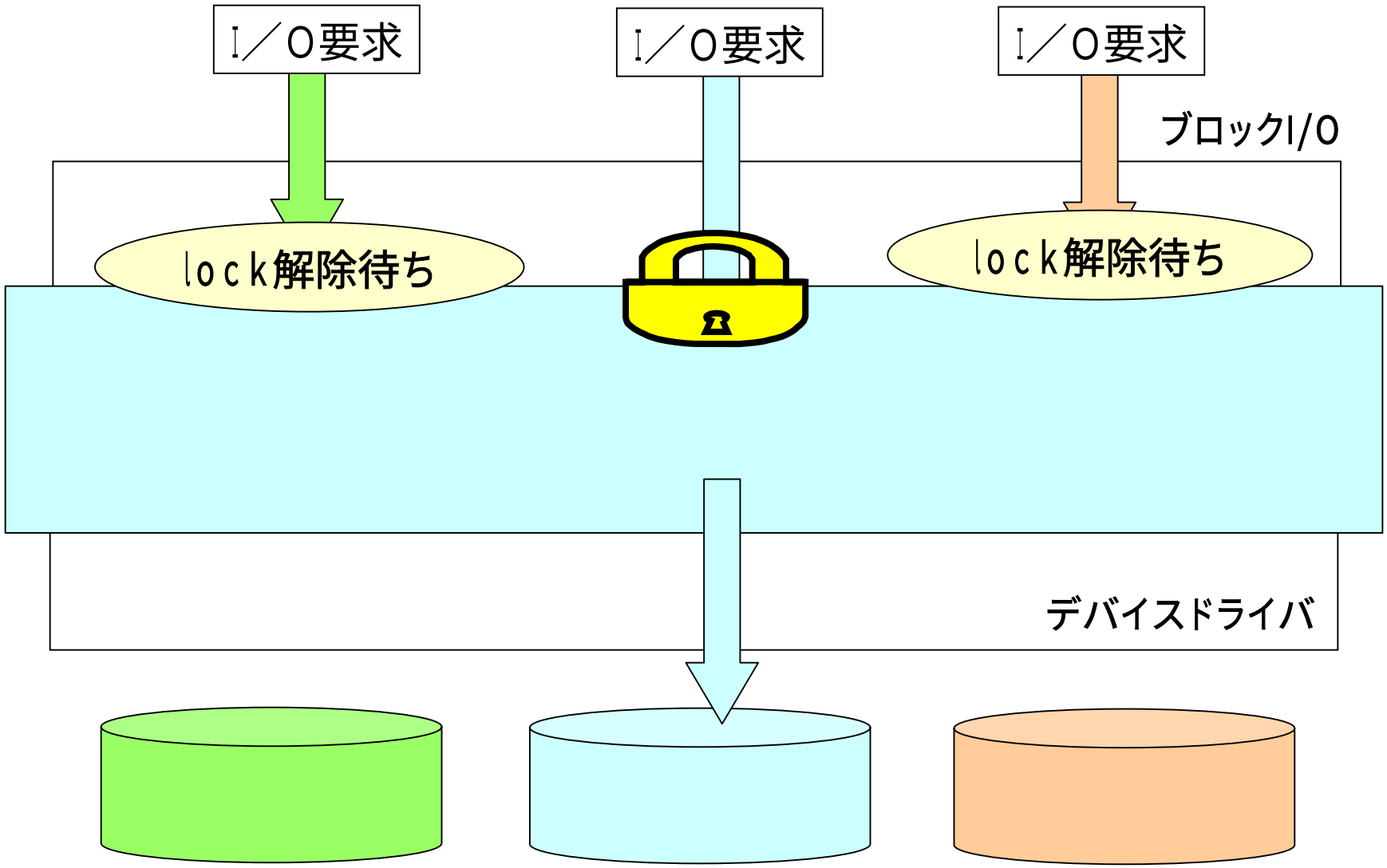
カーネル2．4のパス検索



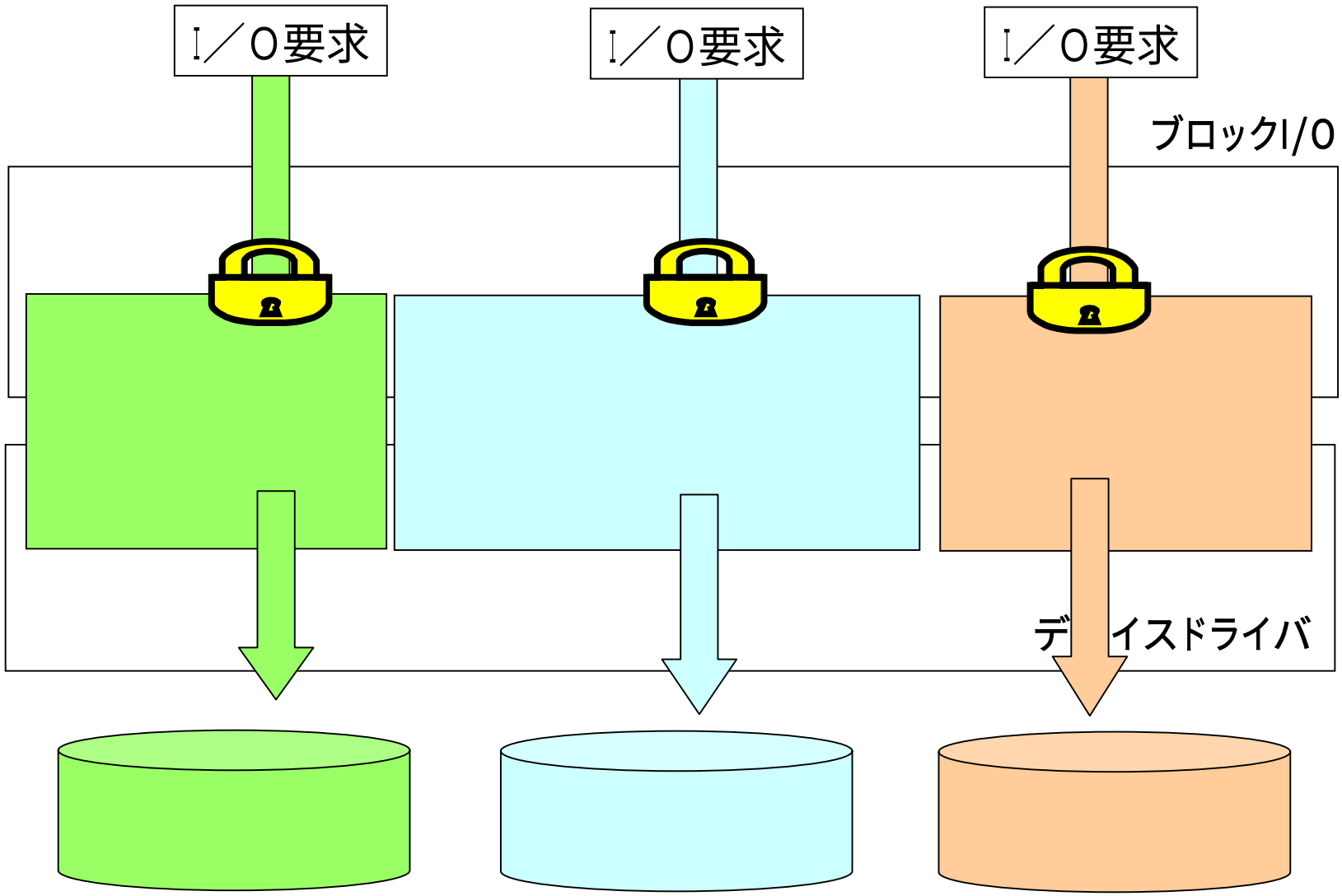
カーネル2．6のパス検索



カーネル2．4のブロックI／O要求処理



カーネル2．6のブロックI／O要求処理



ファイルI／Oのベンチマーク検証

- Bonnie＋＋による性能測定
- IOzoneによる性能測定

Bonnie＋＋とは

- データベースのような大規模なファイル操作のスループットを測定可能
- 比較的小さなファイルの作成･読込み･削除のスループットを測定可能
- バージョン 1．03

http://www.coker.com.au/bonnie++

Bonnie＋＋とは：テストの種類と内容 (1)

- 連続書き込み (Sequential Output)
  - Per Char　関数putc()を使用したキャラクタベースの書き込みテスト
  - Block　関数write()を使用したブロックベースの書き込みテスト
  - Rewrite　関数write()を使用した再書き込みテスト
- 連続読み込み (Sequential Input)
  - Per Char　関数getc()を使用したキャラクタベースの読み込みテスト
  - Block　関数read()を使用したブロックベースの読み込みテスト
- ランダムシーク (Random Seek)
  - Seek　drand48()を使用した8000回のlseek ()

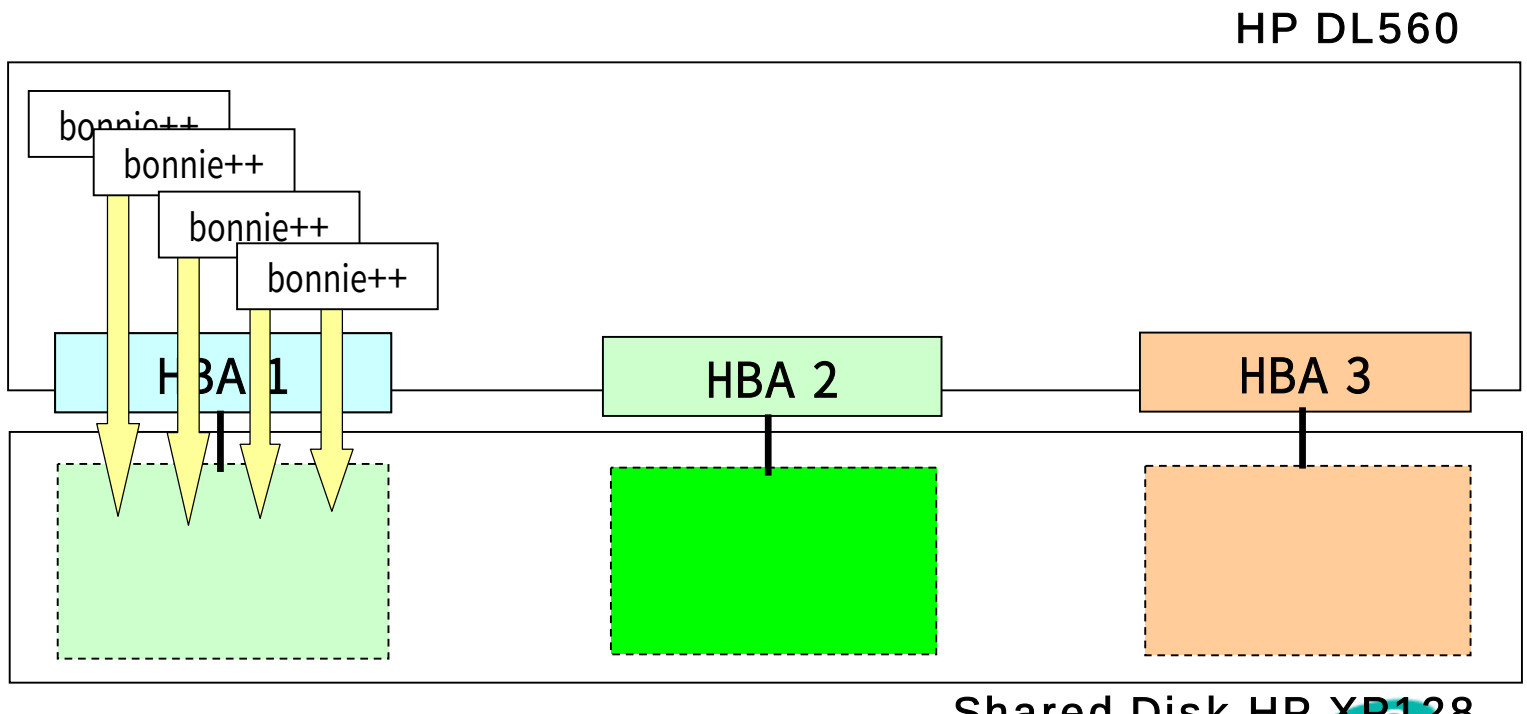
Bonnie＋＋とは：テストの種類と内容 (2)

- 連続操作 (Sequential Create)
  - Create　creat()を使用したファイルの作成テスト
    ファイル名は7桁数値、0-12までの英数字
  - Read　stat()を使用したファイル情報の確認テスト
  - Delete　unlink()を使用したファイルの削除テスト

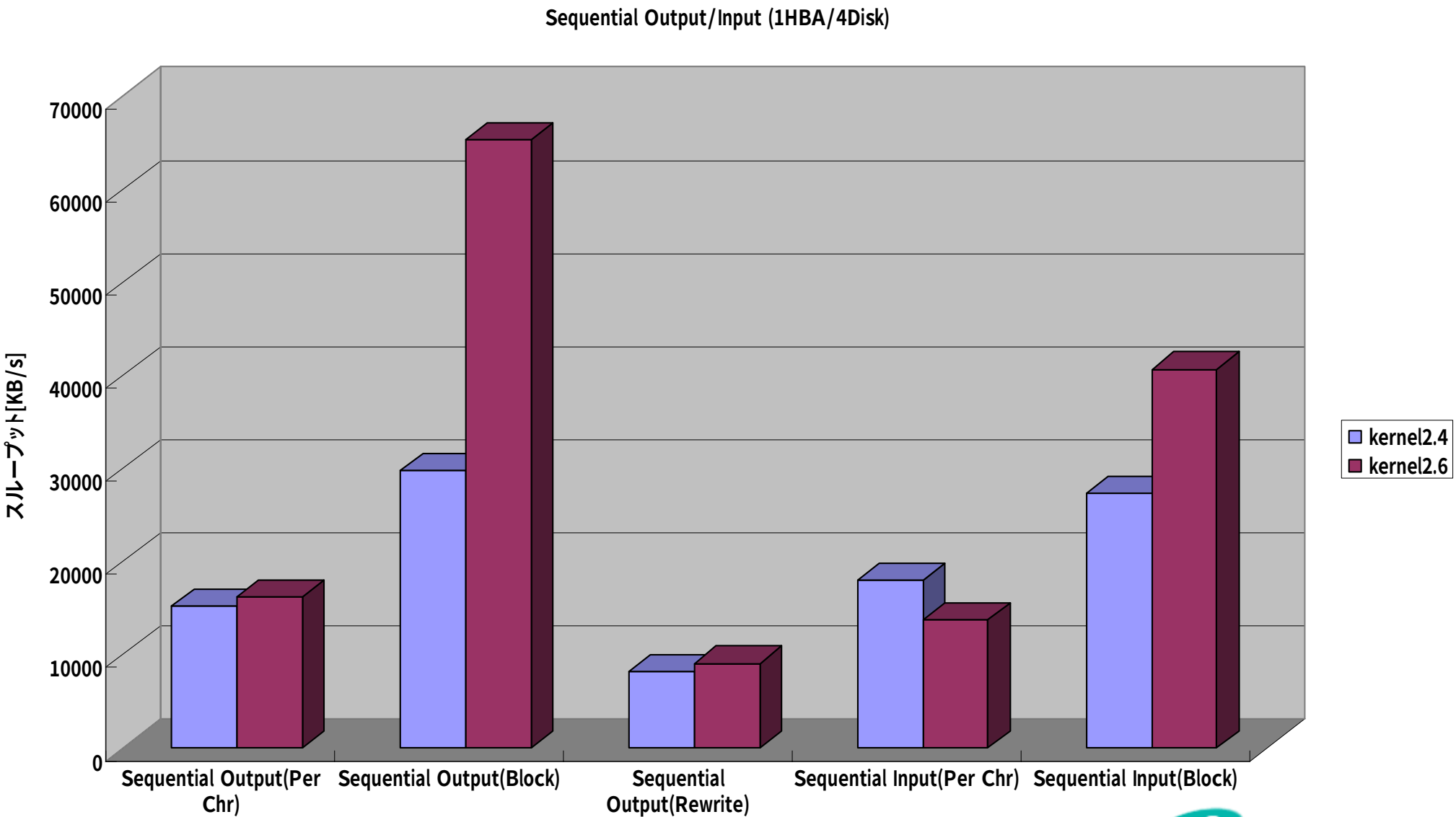
- ランダム操作 (Random Create)
  - Create　creat()を使用したファイルの作成テスト
    ファイル名は任意
  - Read　stat()を使用したファイル情報の確認テスト
  - Delete　unlink()を使用したファイルの削除テスト

テスト1

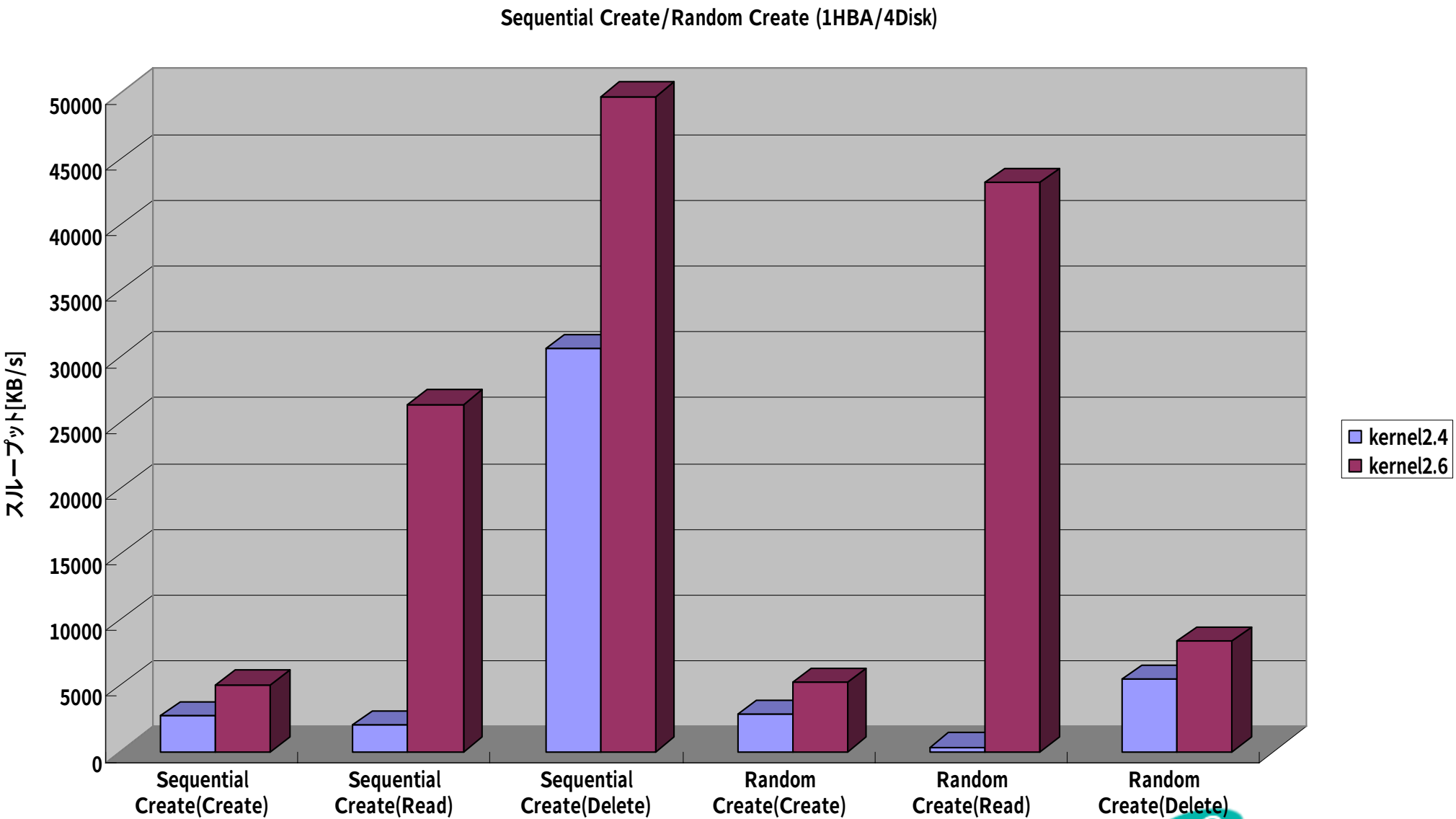
- 1HBA／4Diskのテスト
- 1プロセス（Bonnie++)／1Disk（全部で4プロセス起動）
- 連続書き込み／連続読み込み
  ファイルサイズ 6GB
- 連続操作／ランダム操作
  約10万のファイル（80～100バイト）を10ディレクトリに作成
- 測定回数 1回、測定結果は全プロセスの平均値



測定結果：テスト1：連続書き込み／読み込み

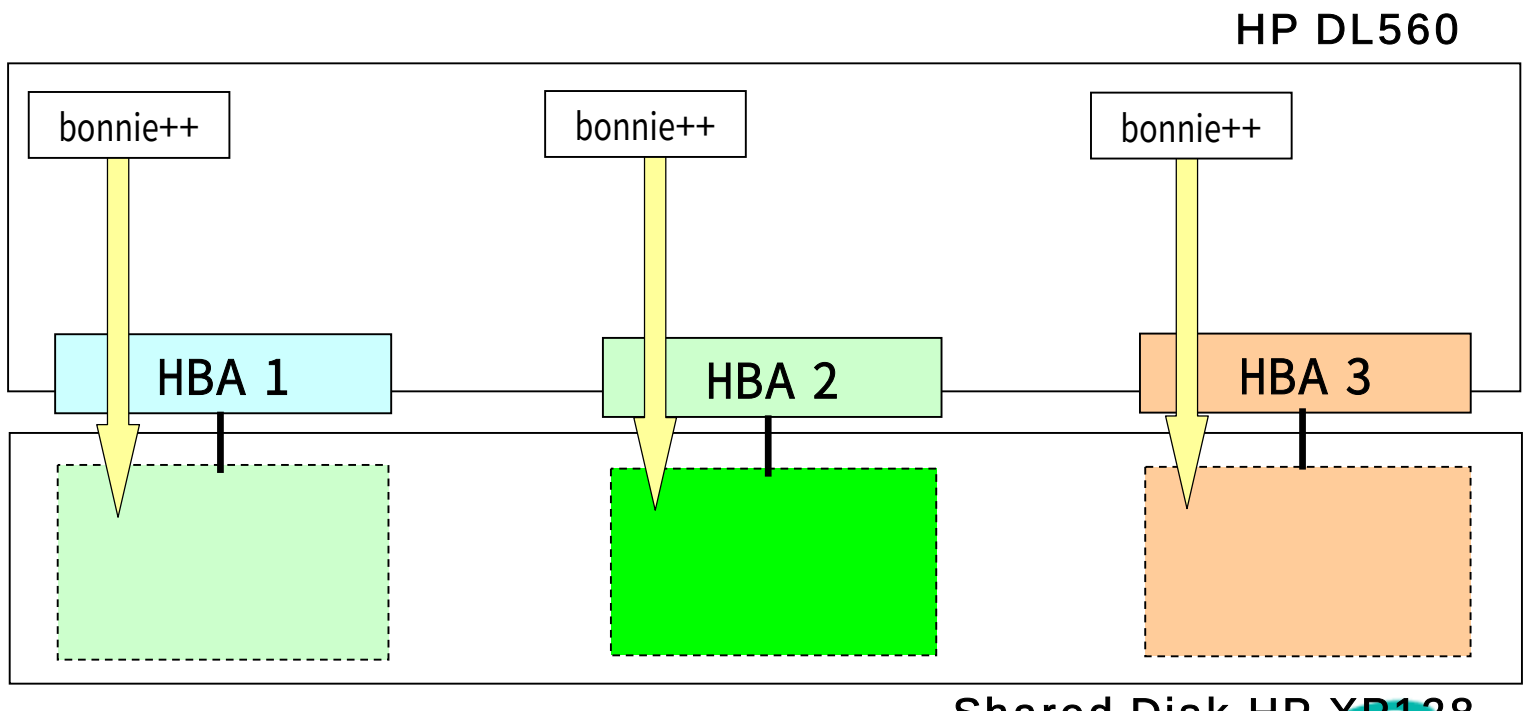


測定結果：テスト1：連続操作／ランダム操作

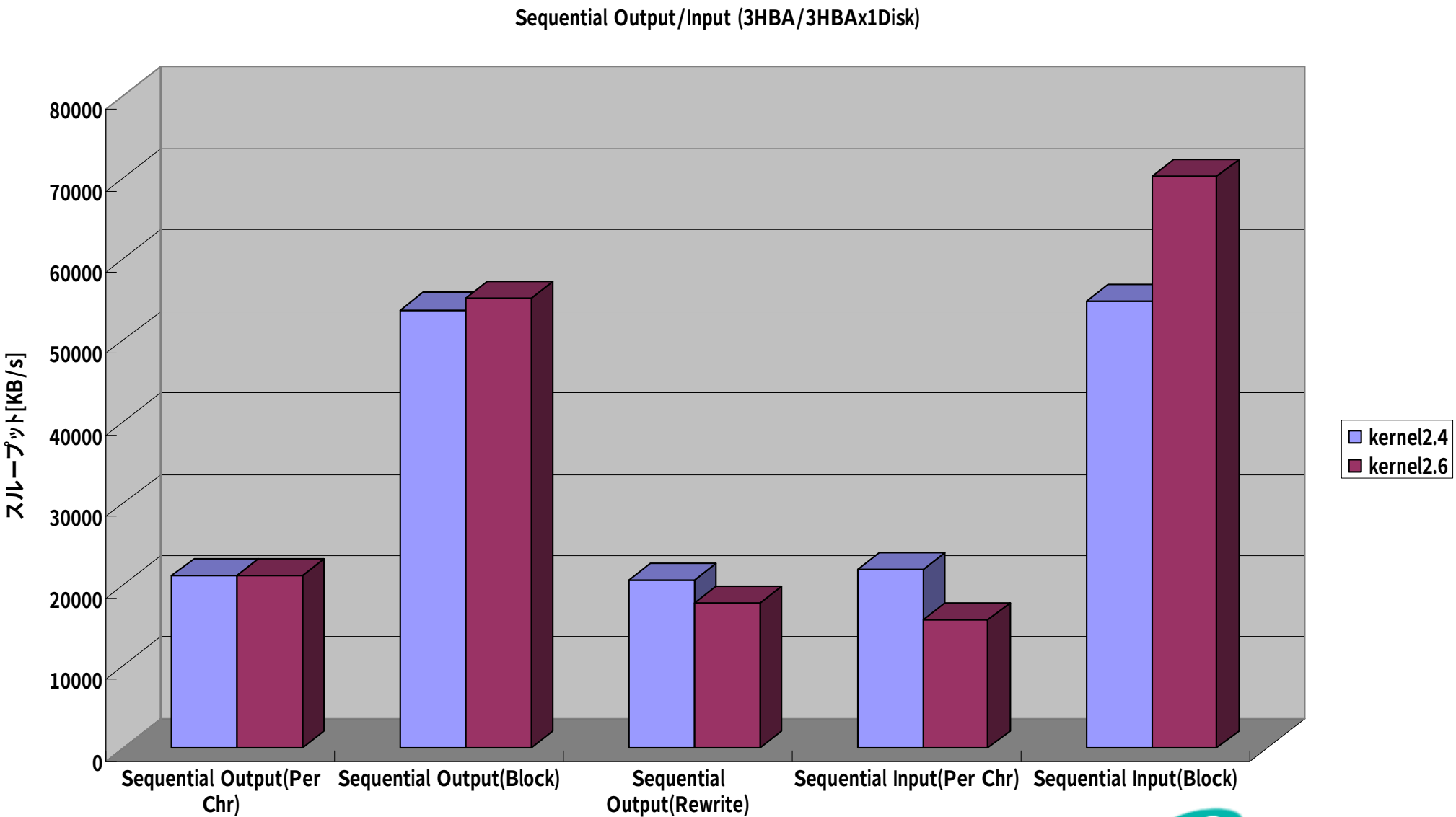


テスト2

- 3HBA／3Disk（3HBA×1Disk）のテスト
- 1プロセス（Bonnie++）／1Disk（全部で3プロセス起動）
- 連続書き込み／連続読み込み
  ファイルサイズ 6GB
- 連続操作／ランダム操作
  約10万のファイル（80～100バイト）を10ディレクトリに作成
- 測定回数 1回、測定結果は全プロセスの平均値

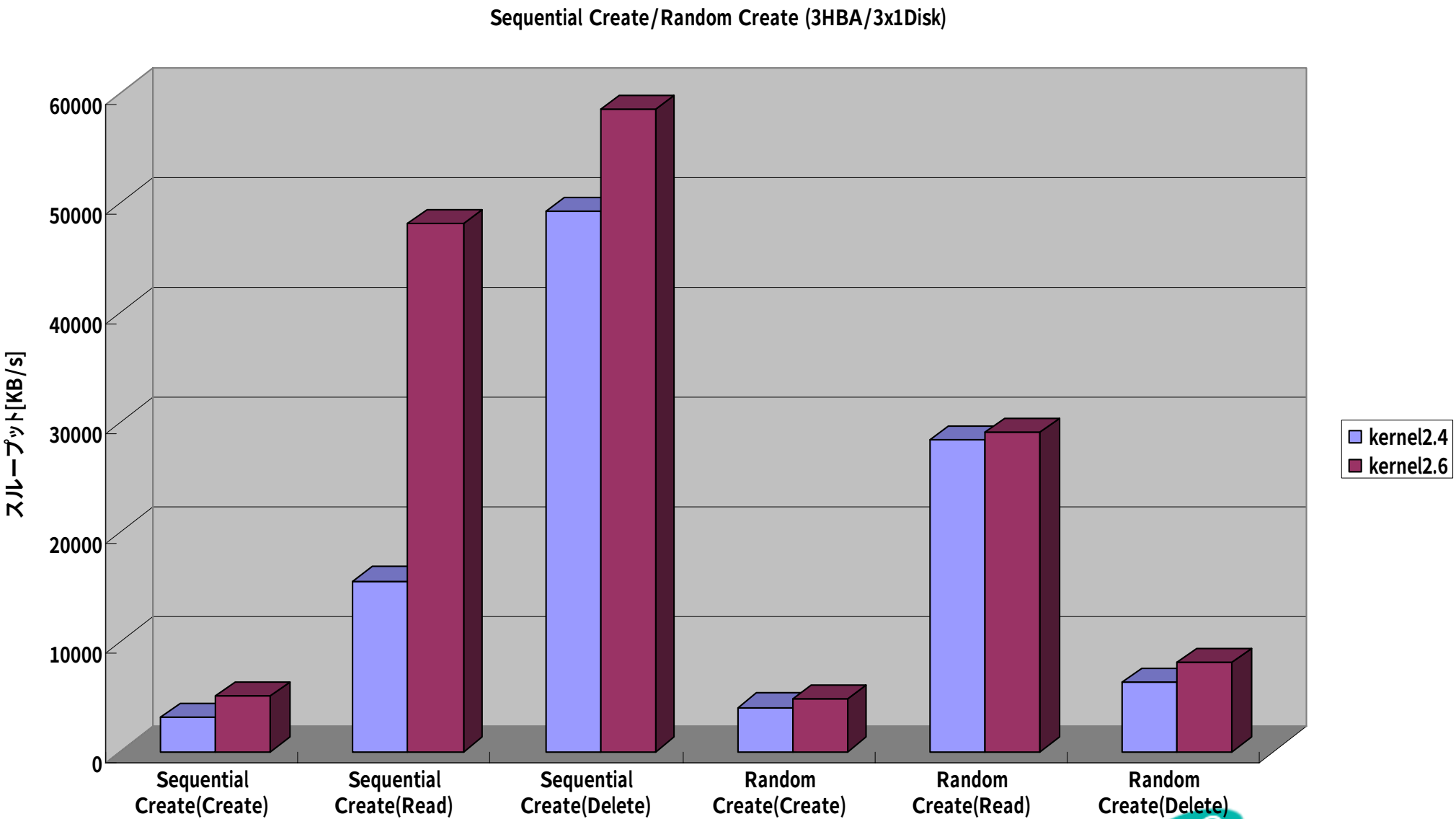


測定結果：テスト2：連続書き込み／読み込み

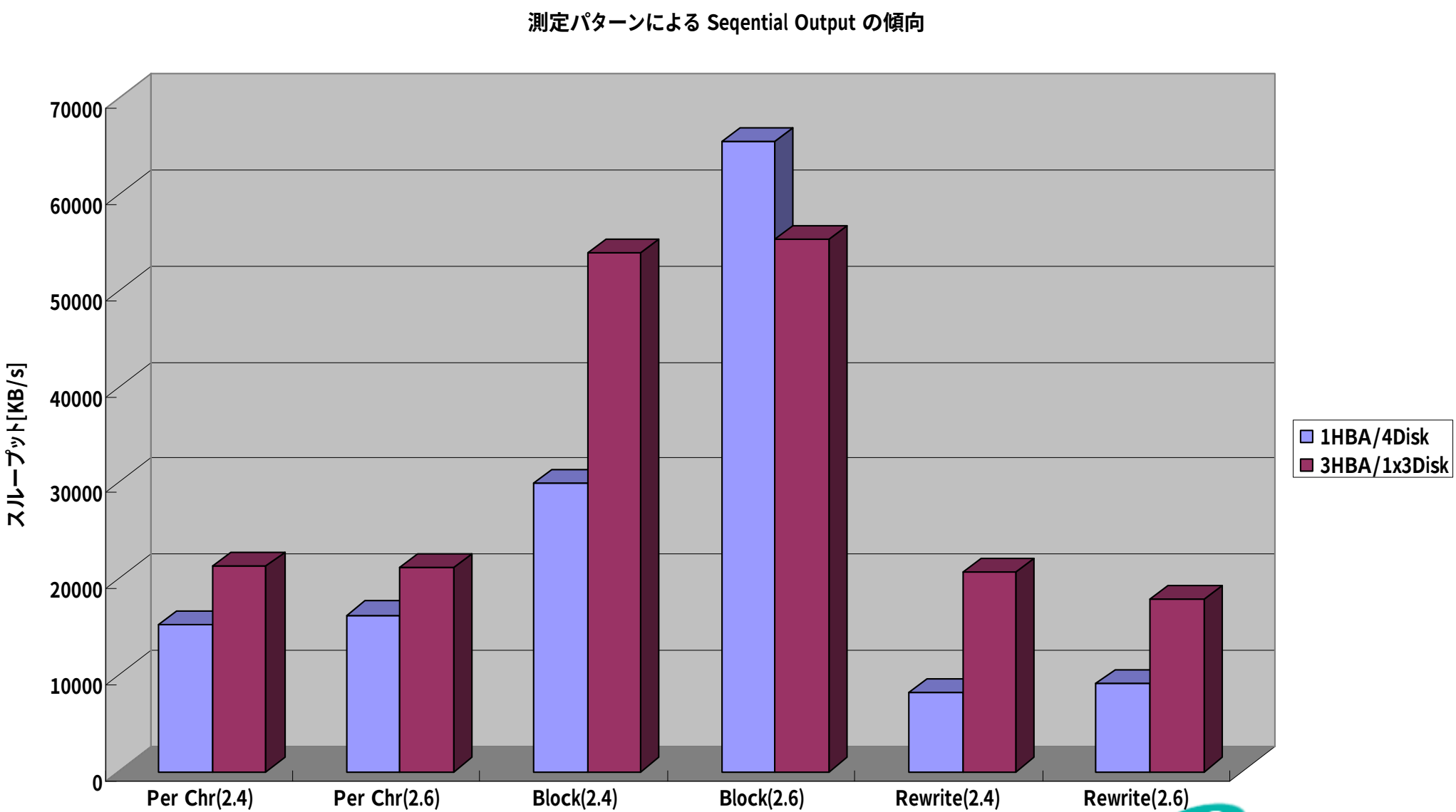


NTT COMWARE

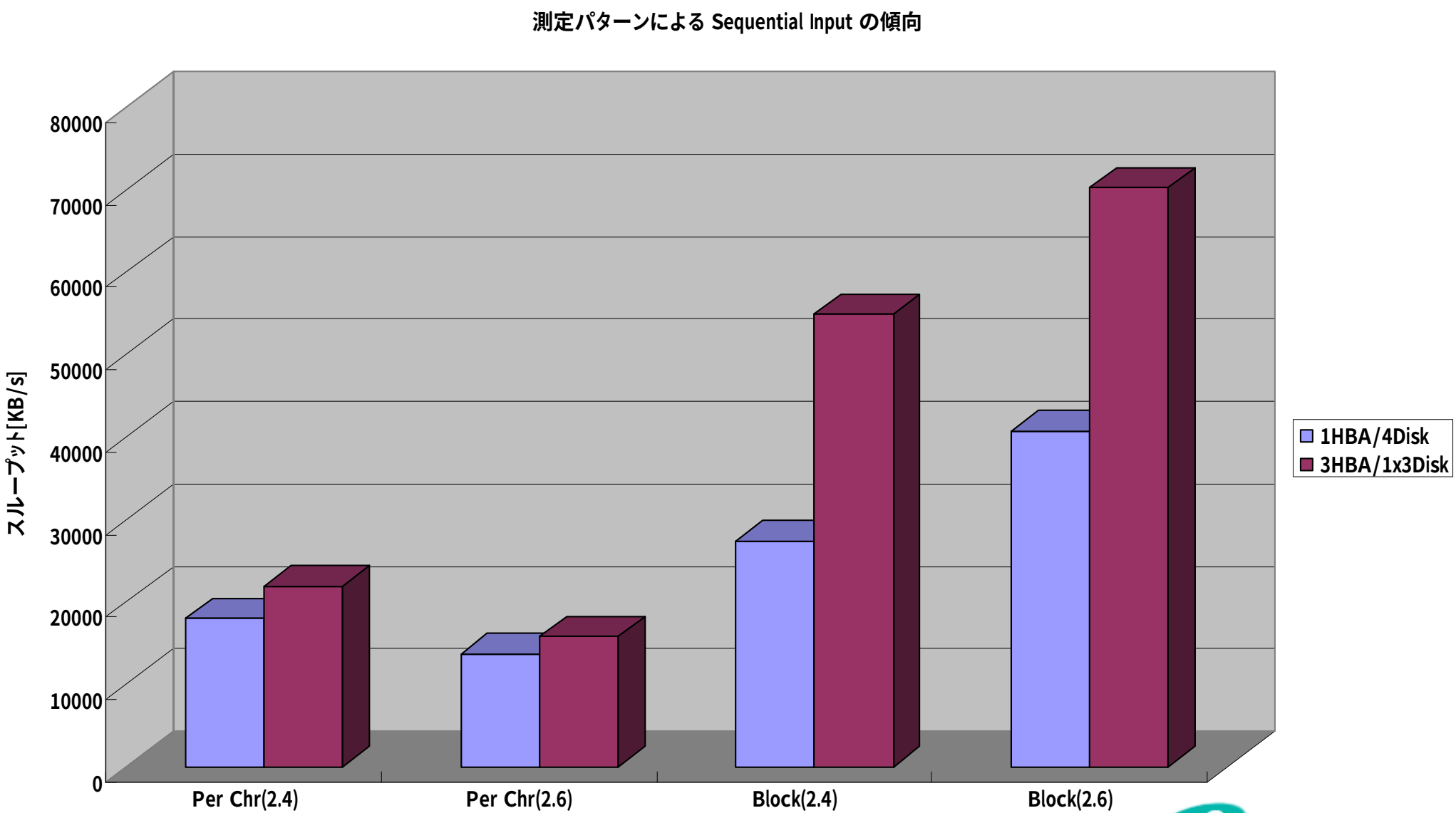
測定結果：テスト2：連続操作／ランダム操作



分析：測定パターンによる連続書き込みの傾向 (1)



分析：測定パターンによる連続読み込みの傾向 (2)



IOzoneとは

- 各種ファイル操作（read、write、fread、fwriteなど）のパフォーマンスを測定可能
- 操作対象ファイルのサイズとI/Oレコードサイズを変更しながら、ファイルシステムのパフォーマンスを計測
- バージョン 3．147

http://www.iozone.org/

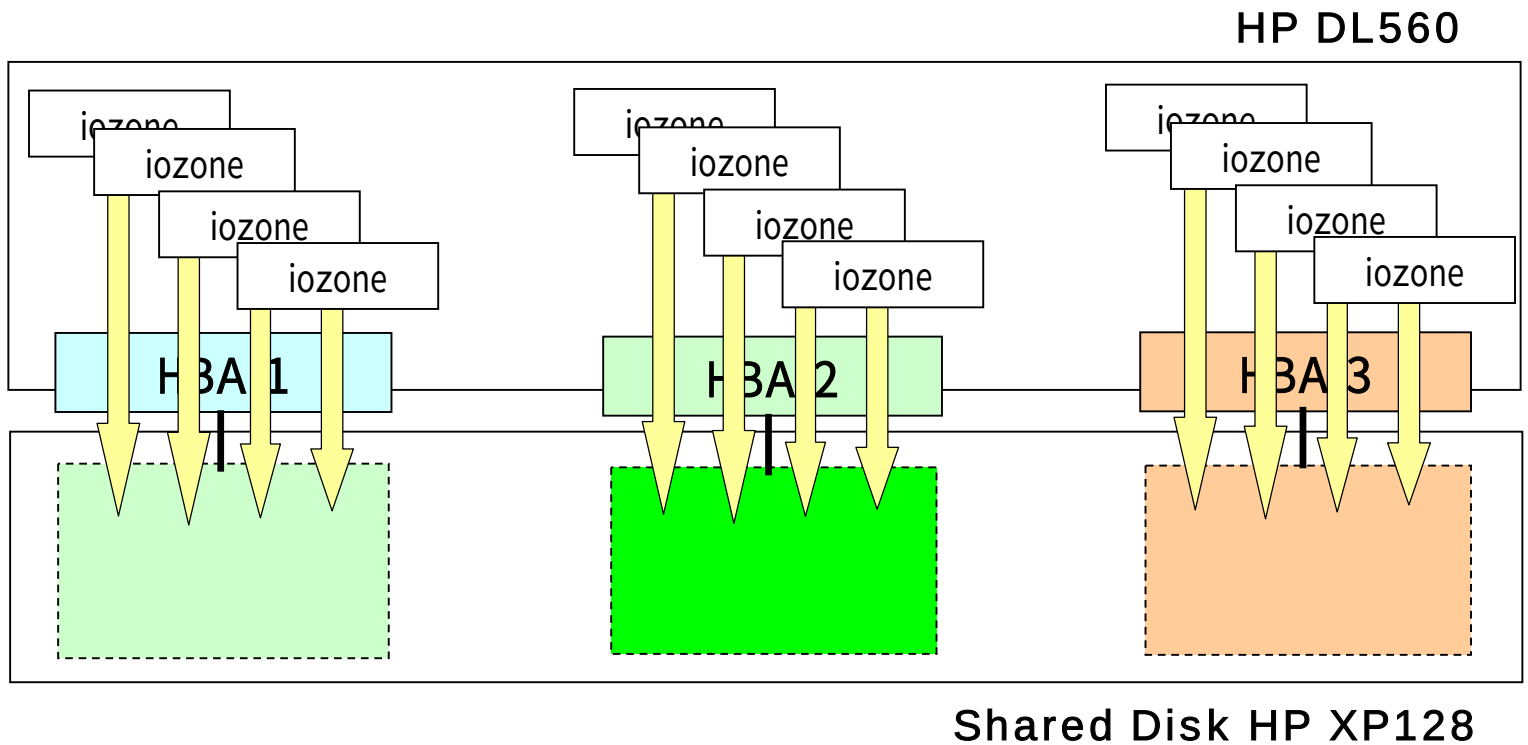
IOzoneとは： テストの種類と内容

- 新規ファイル書き込み
  write()を使用した書き込み
- 既存ファイルの読み込み
  read()を使用した読み込みテスト
- ランダム読み込み
  ファイル内のランダムな位置に対する読み込みテスト
- ランダム書き込み
  ファイル内のランダムな位置に対する書き込みテスト

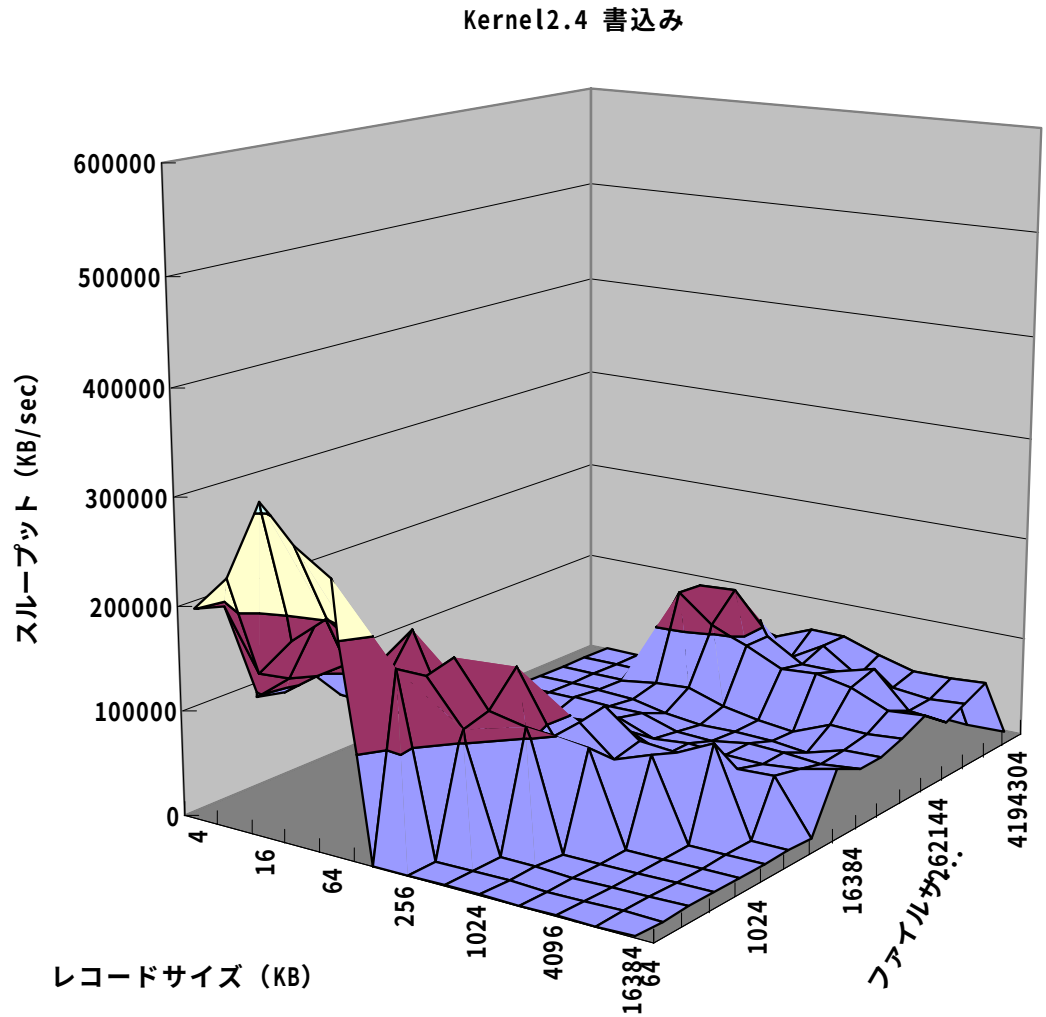
その他、fread()／fwrite()、mmap()など、多彩な測定が可能

テスト内容

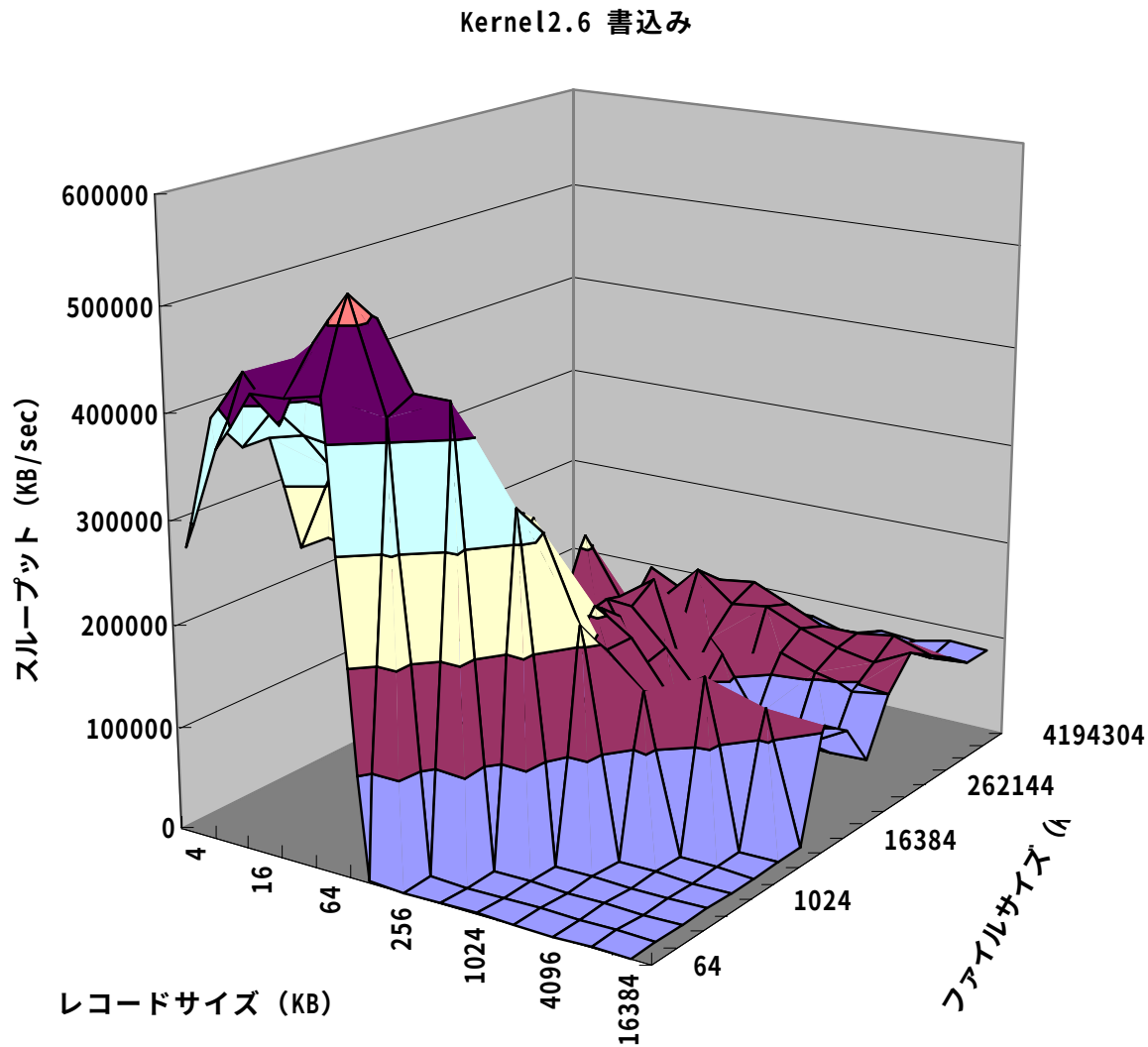
- 3HBA/12Disk（3HBA×4Disk）のテスト
- 1プロセス（Iozone)／1Disk（全部で12プロセス起動）
- 最大ファイルサイズ 4GB
- 測定回数 1回、測定結果は全プロセスの平均値



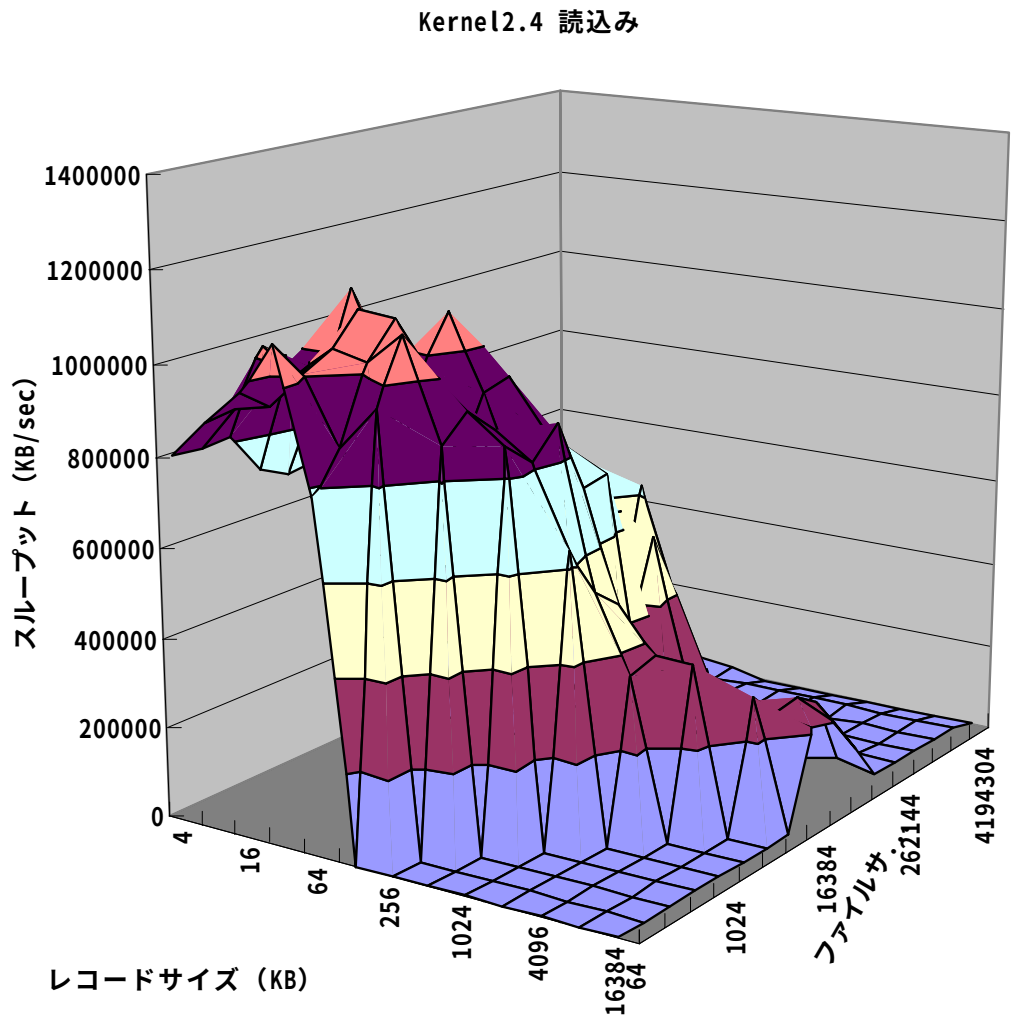
測定結果： 連続書込み：kernel 2.4



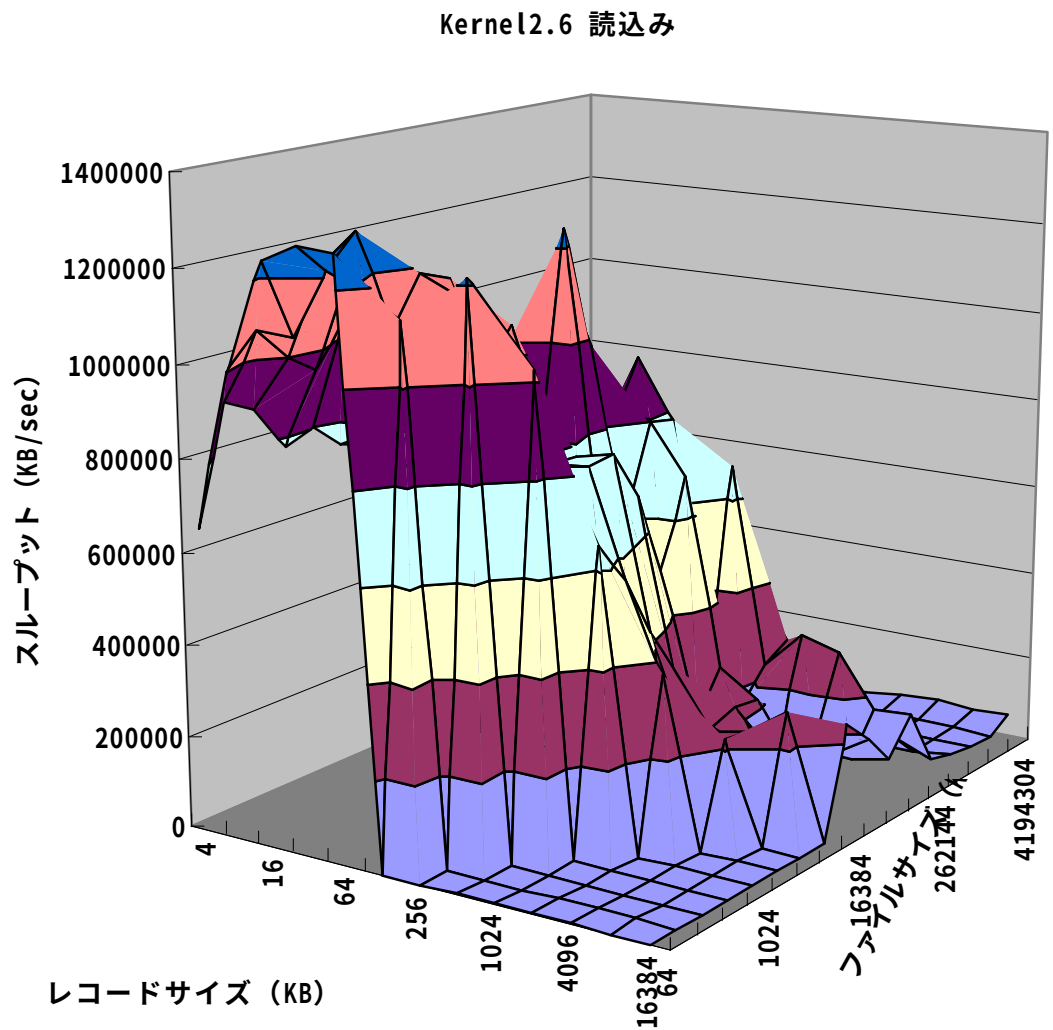
測定結果： 連続書込み：kernel 2.6



測定結果： 連続読込み：kernel 2.4



測定結果： 連続読込み：kernel 2.6



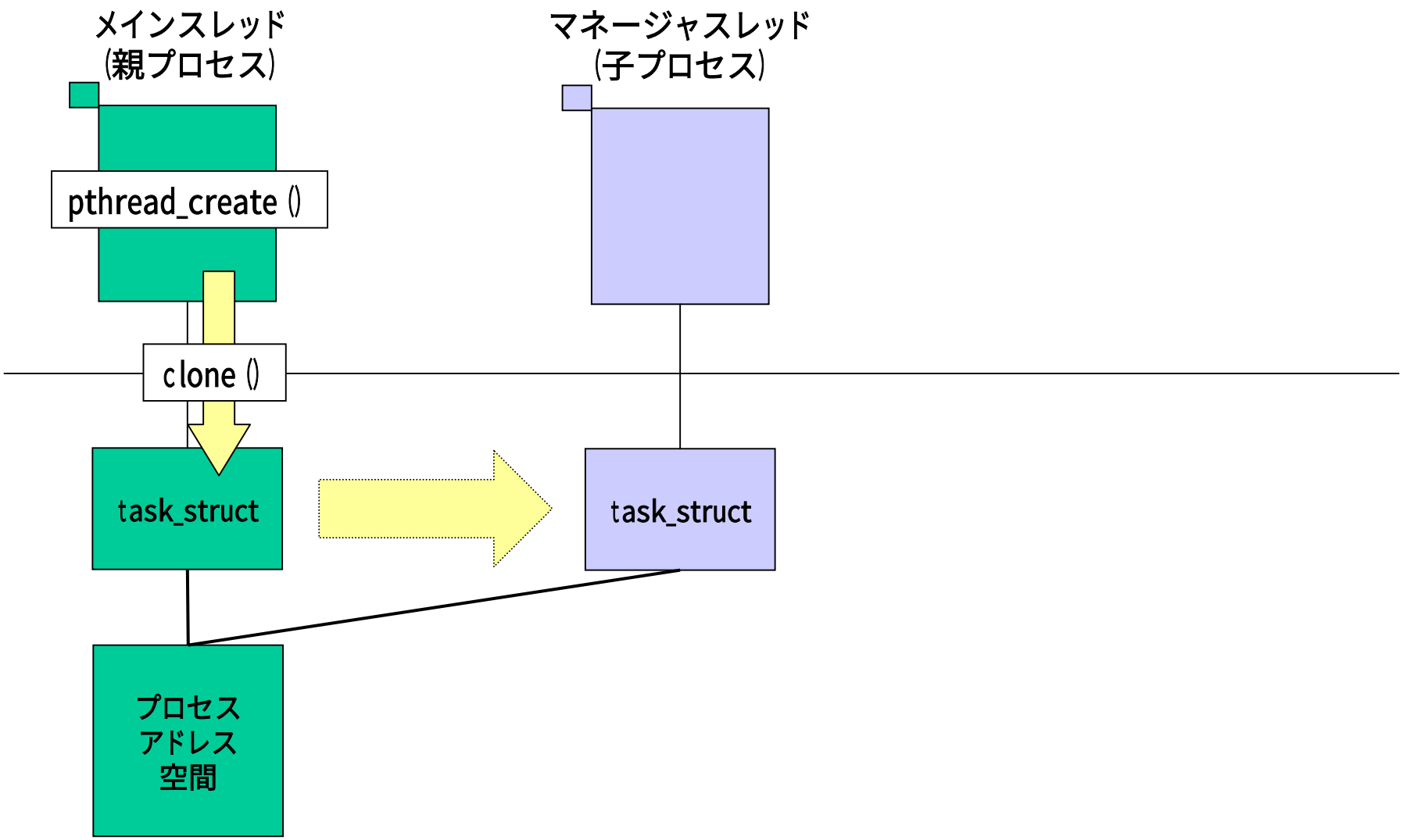
スレッドの実力を探る

― ＪａｖａＶＭベンチマークによる検証 ―

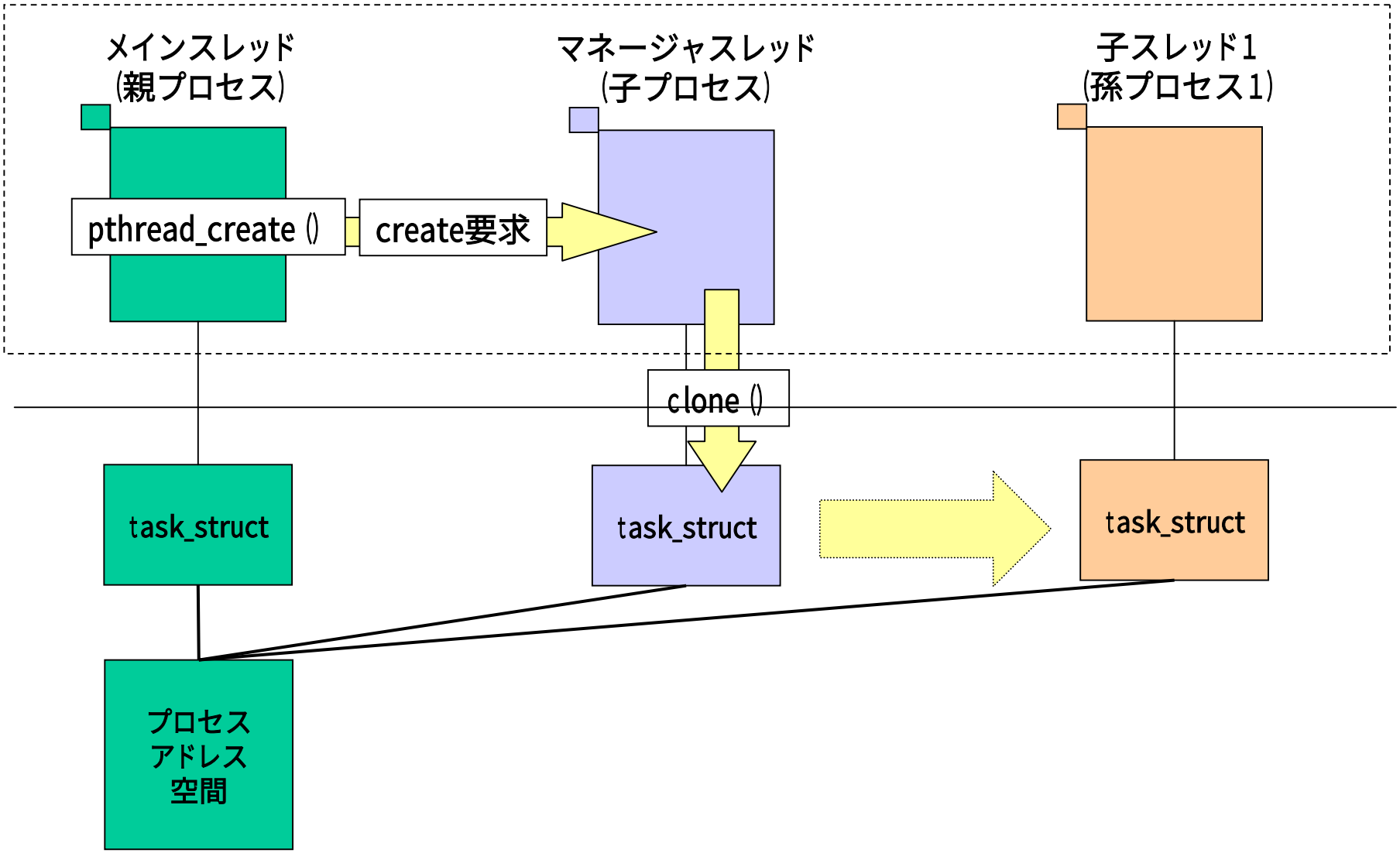
スレッド機能の性能向上

- スレッドを意識したカーネルの拡張
  (スレッドとプロセスを明確に区別)
- futex (Fast Userspace Mutex)
- NPTL (Native Posix Thread Library)

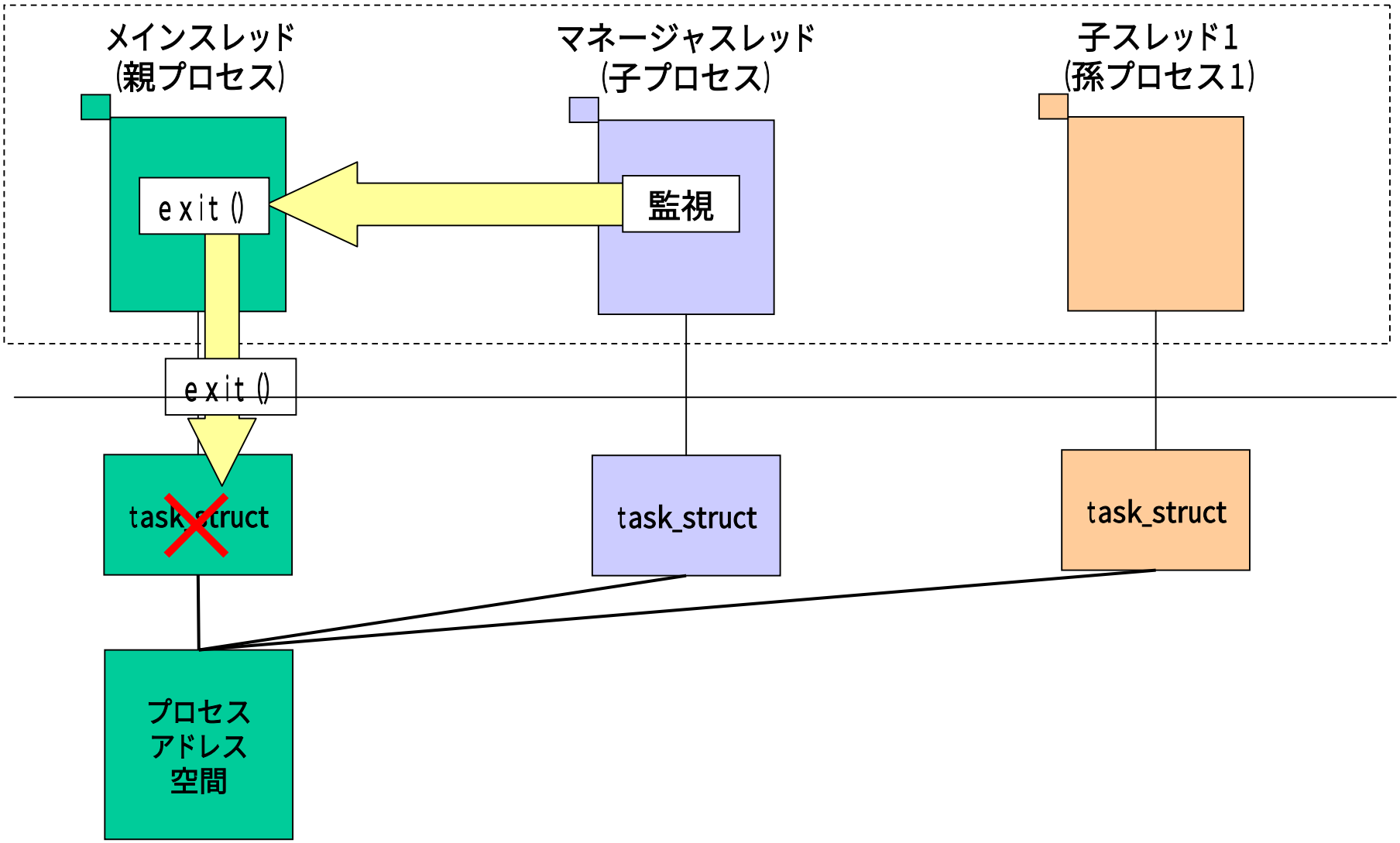
カーネル2．4のスレッド生成（1）



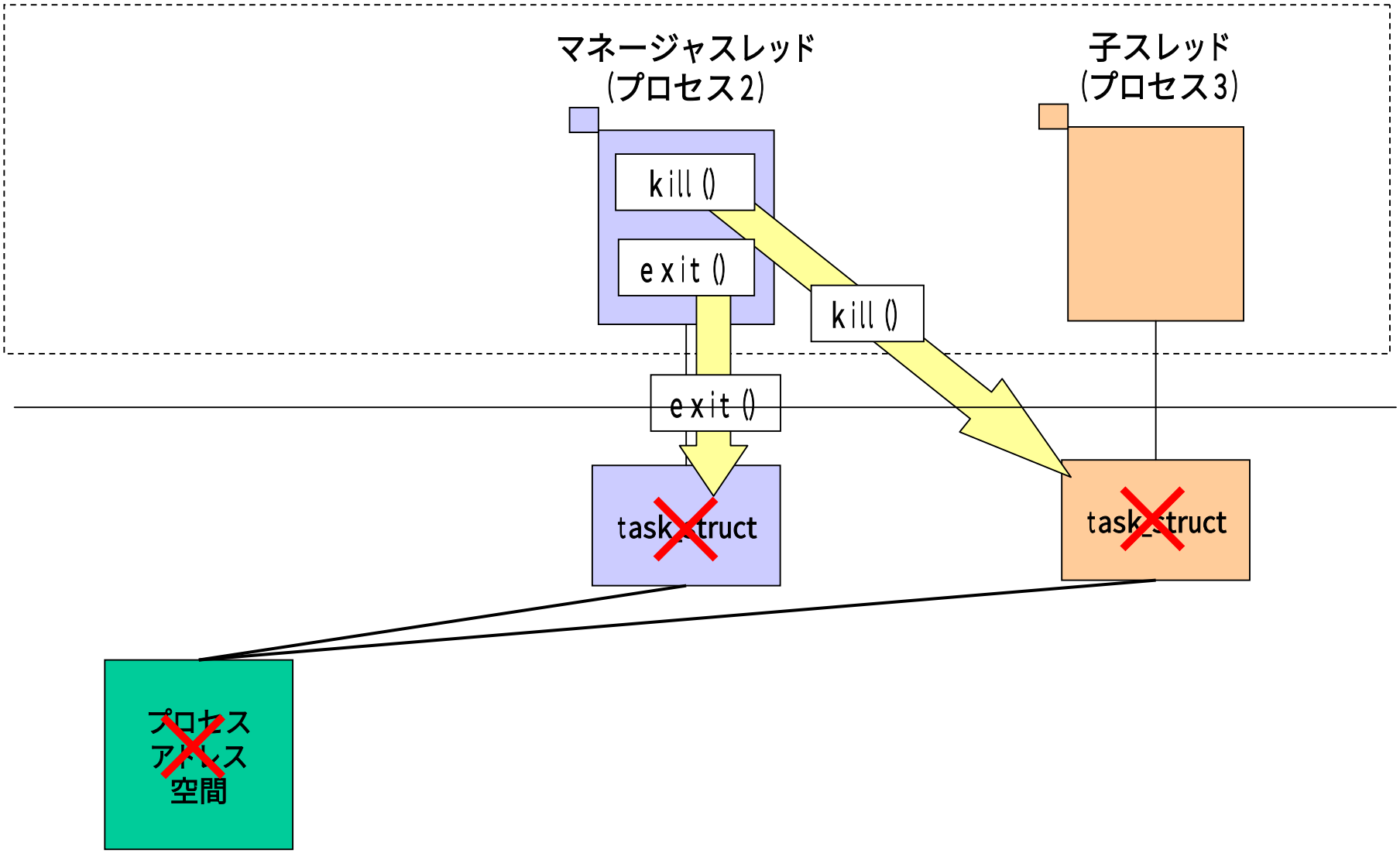
カーネル2．4のスレッド生成（2）



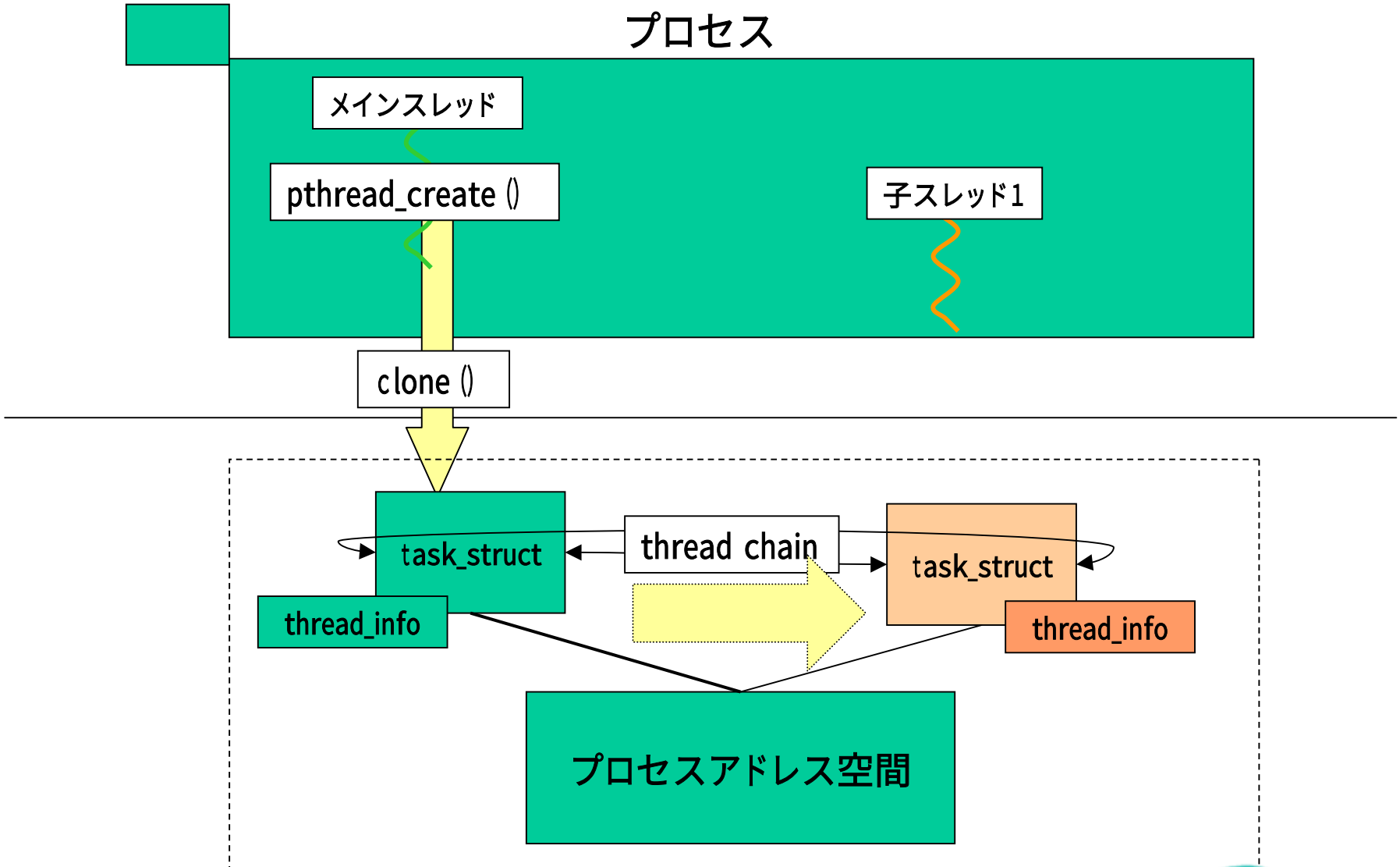
カーネル2．4のメインスレッド終了（1）



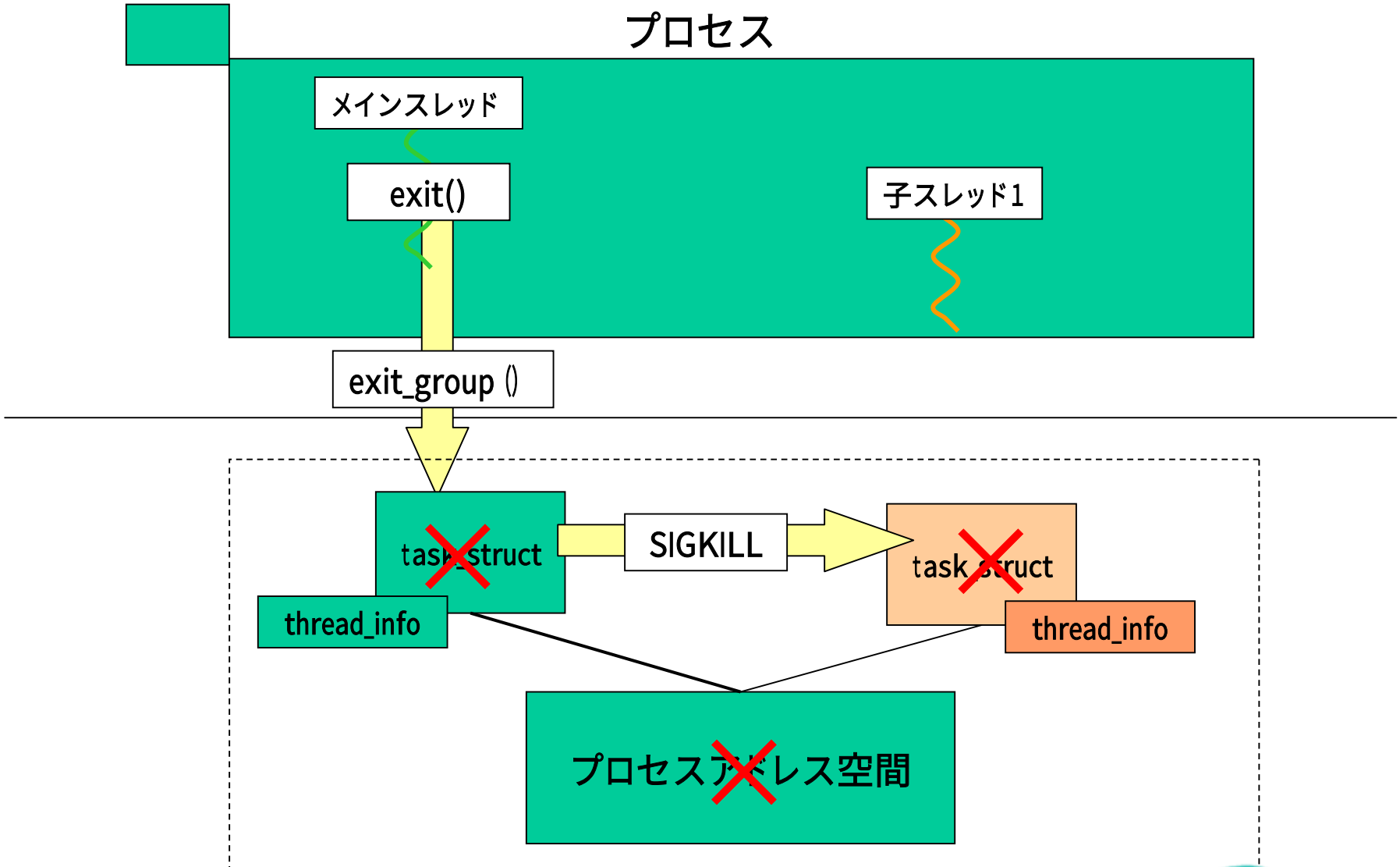
カーネル2．4のメインスレッド終了（2）



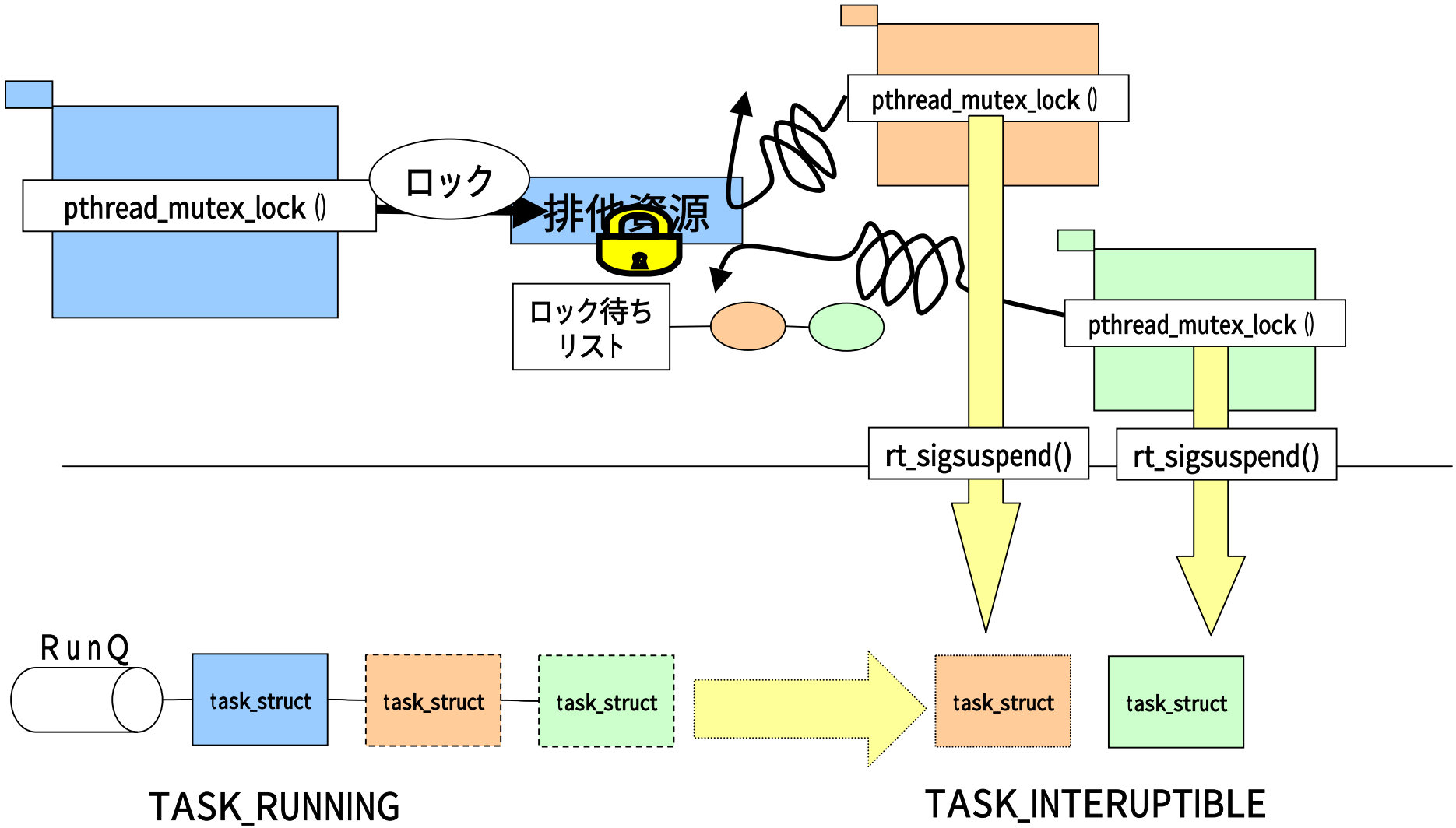
カーネル2．6のスレッド生成



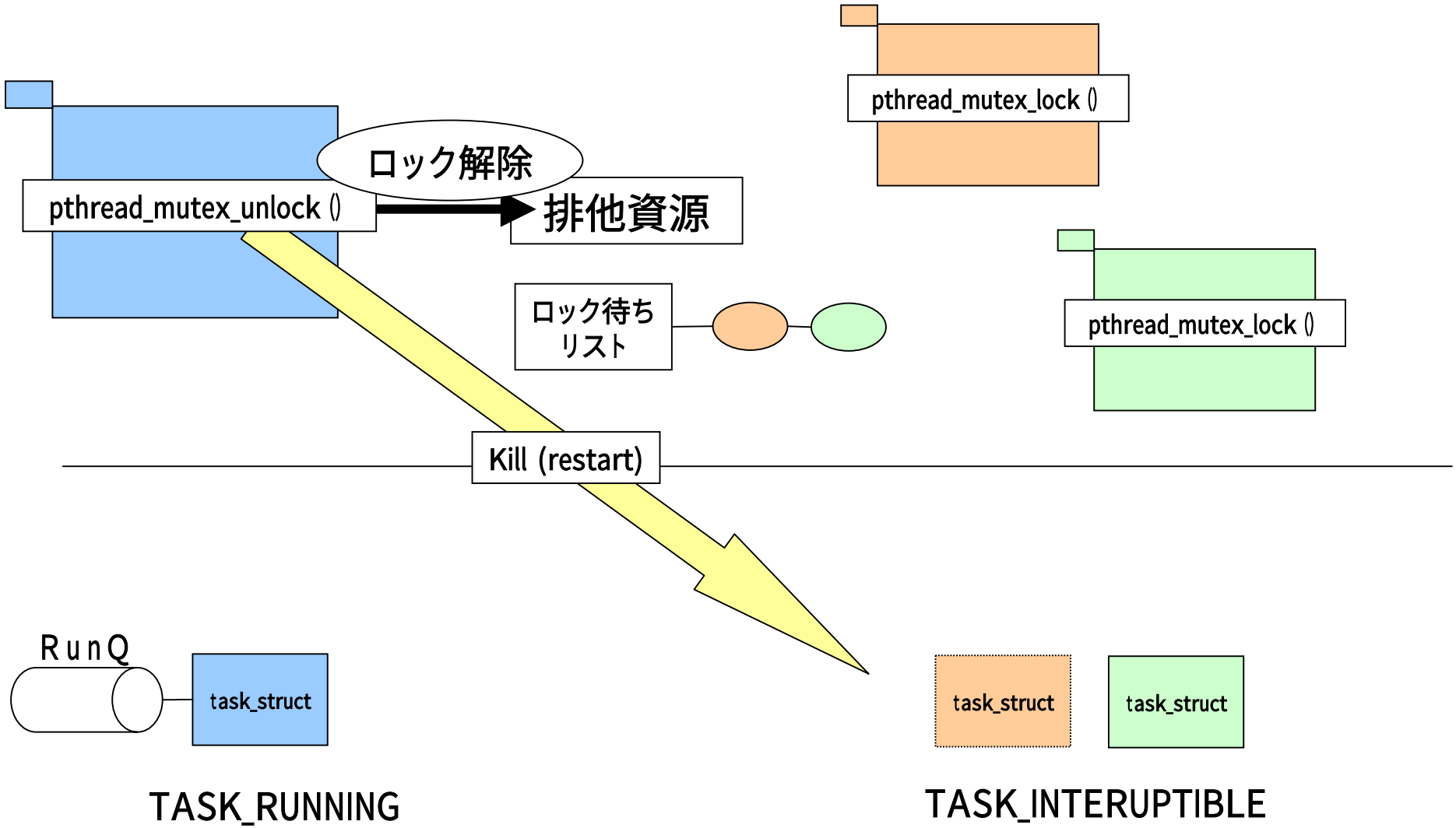
カーネル2．6のメインスレッド終了



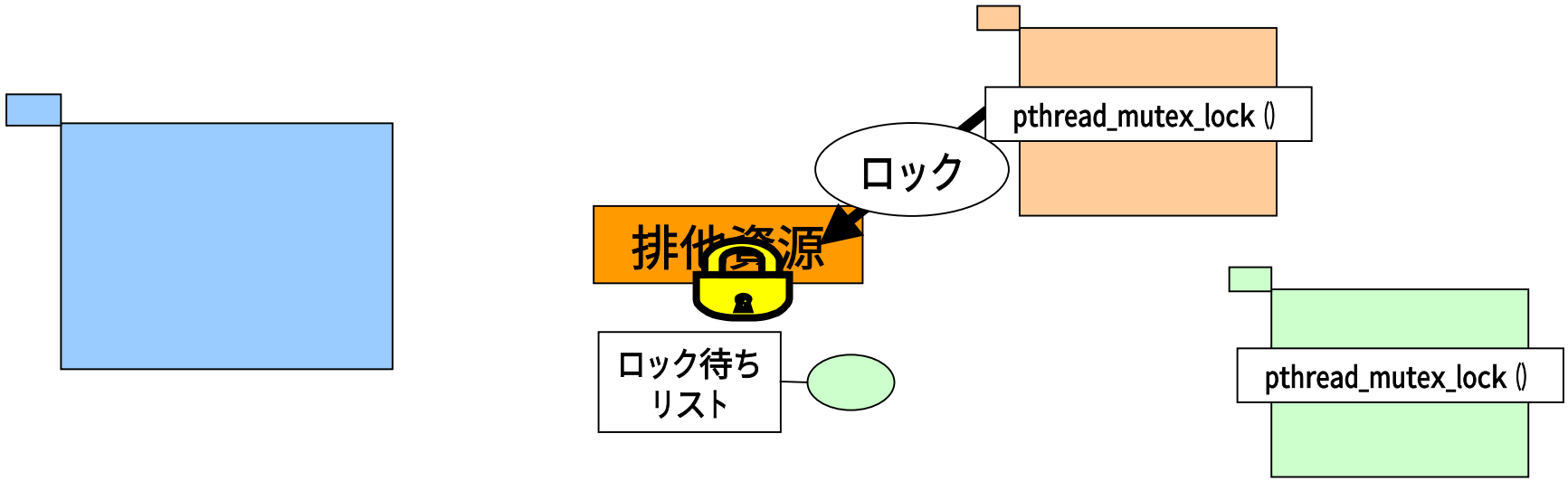
カーネル2．4のロック、アンロック（1）



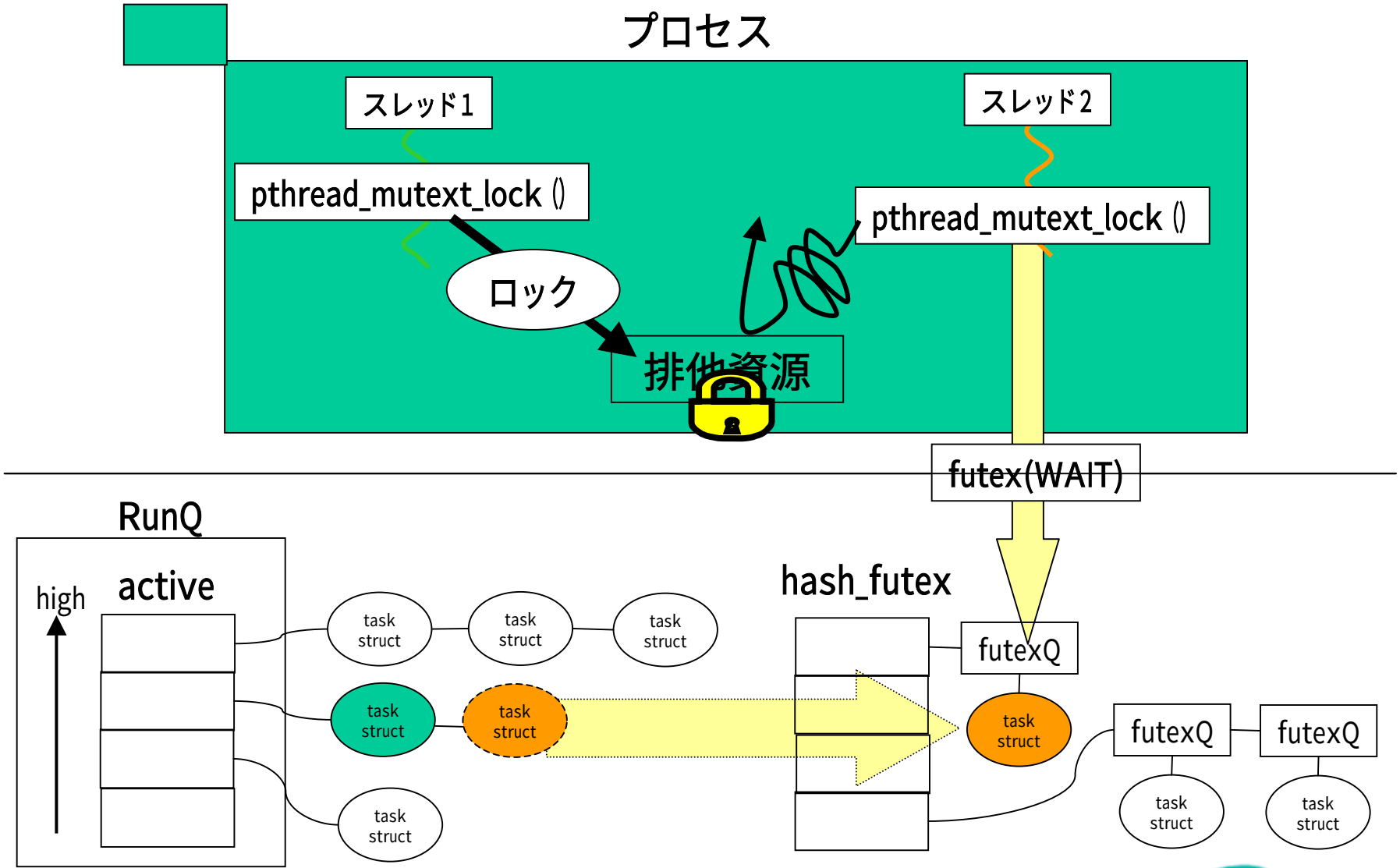
カーネル2．4のロック、アンロック（2）



カーネル2．4のロック、アンロック（3）

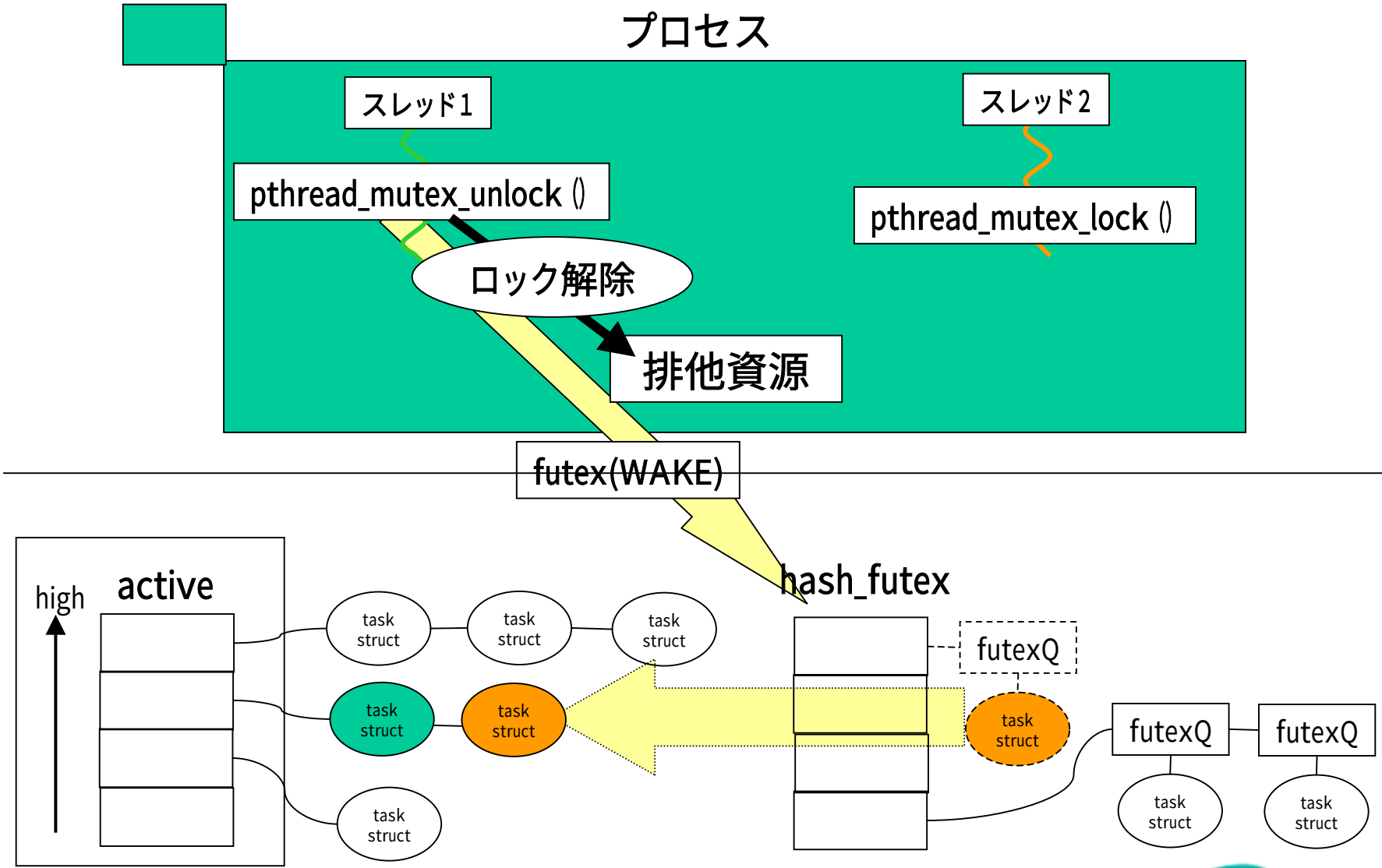


カーネル2．6のロック、アンロック（1）

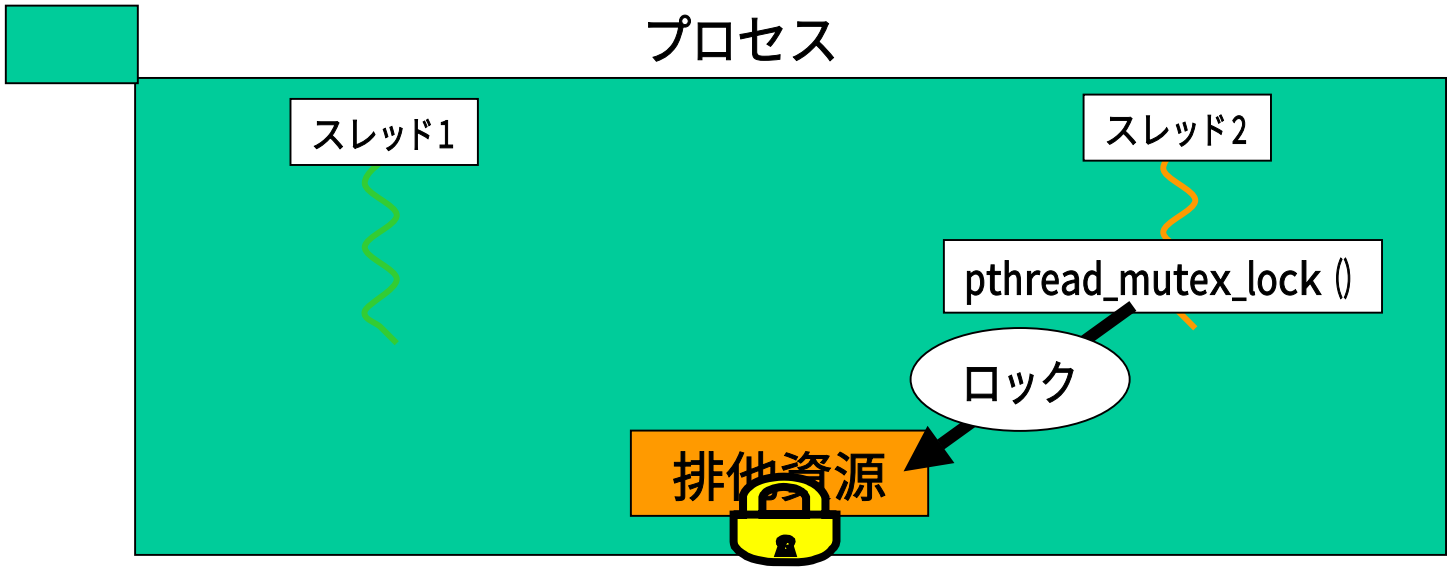


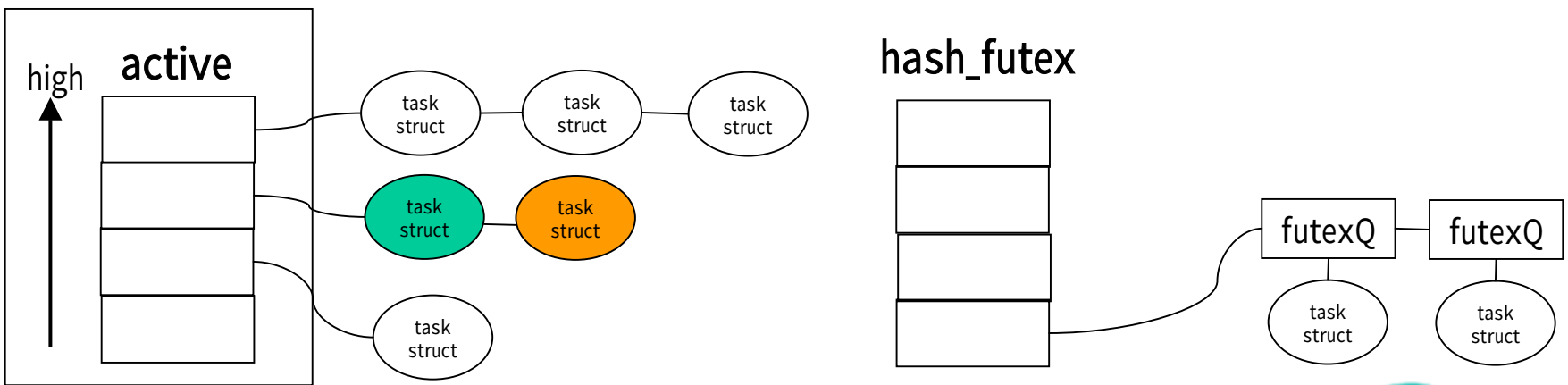
NTT COMWARE

カーネル2．6のロック、アンロック（2）



カーネル2．6のロック、アンロック（3）



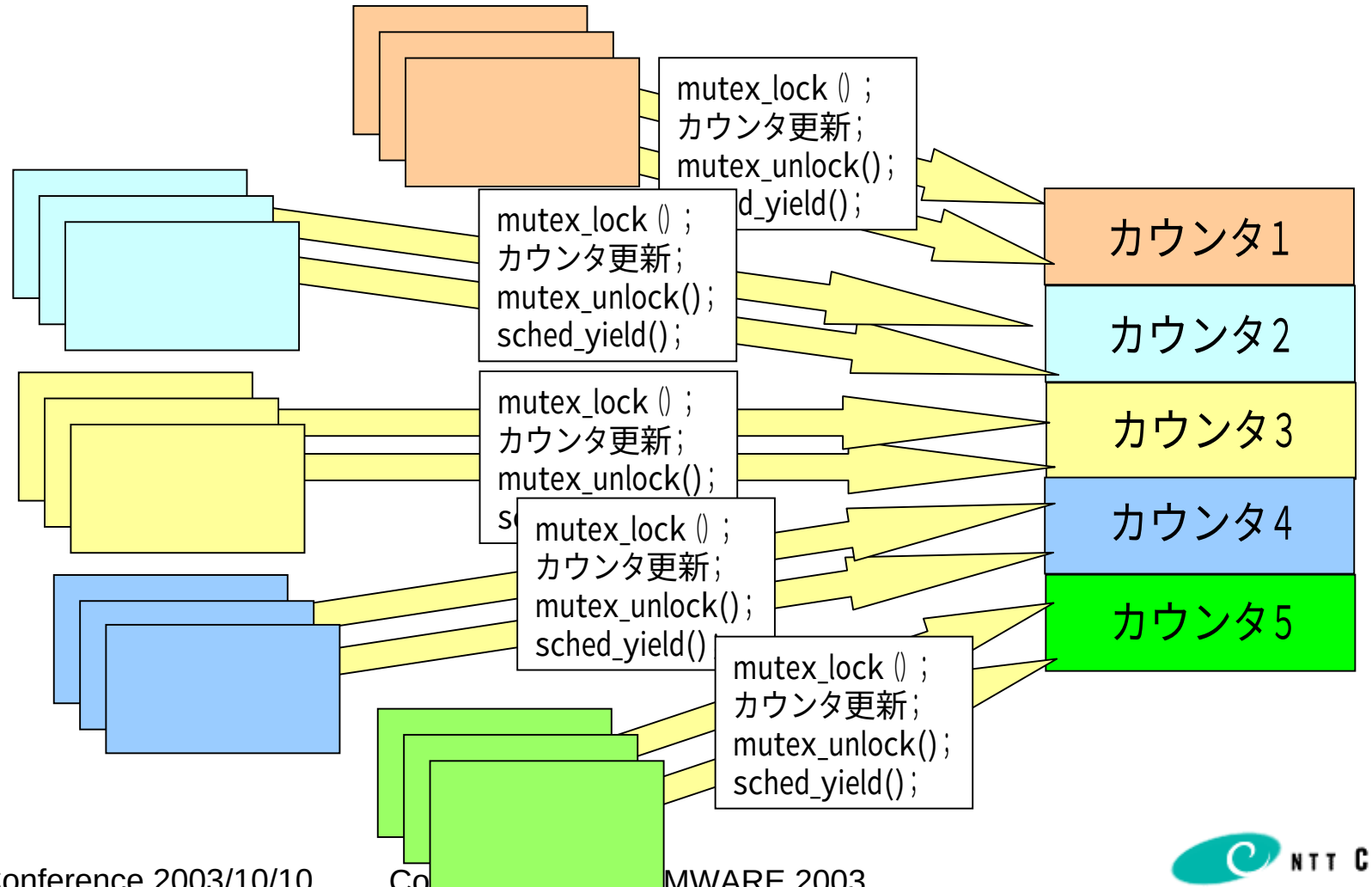


スレッドのベンチマーク検証

- テストプログラムによる性能測定
- VolanoMarkによる性能測定

テストプログラムの概要

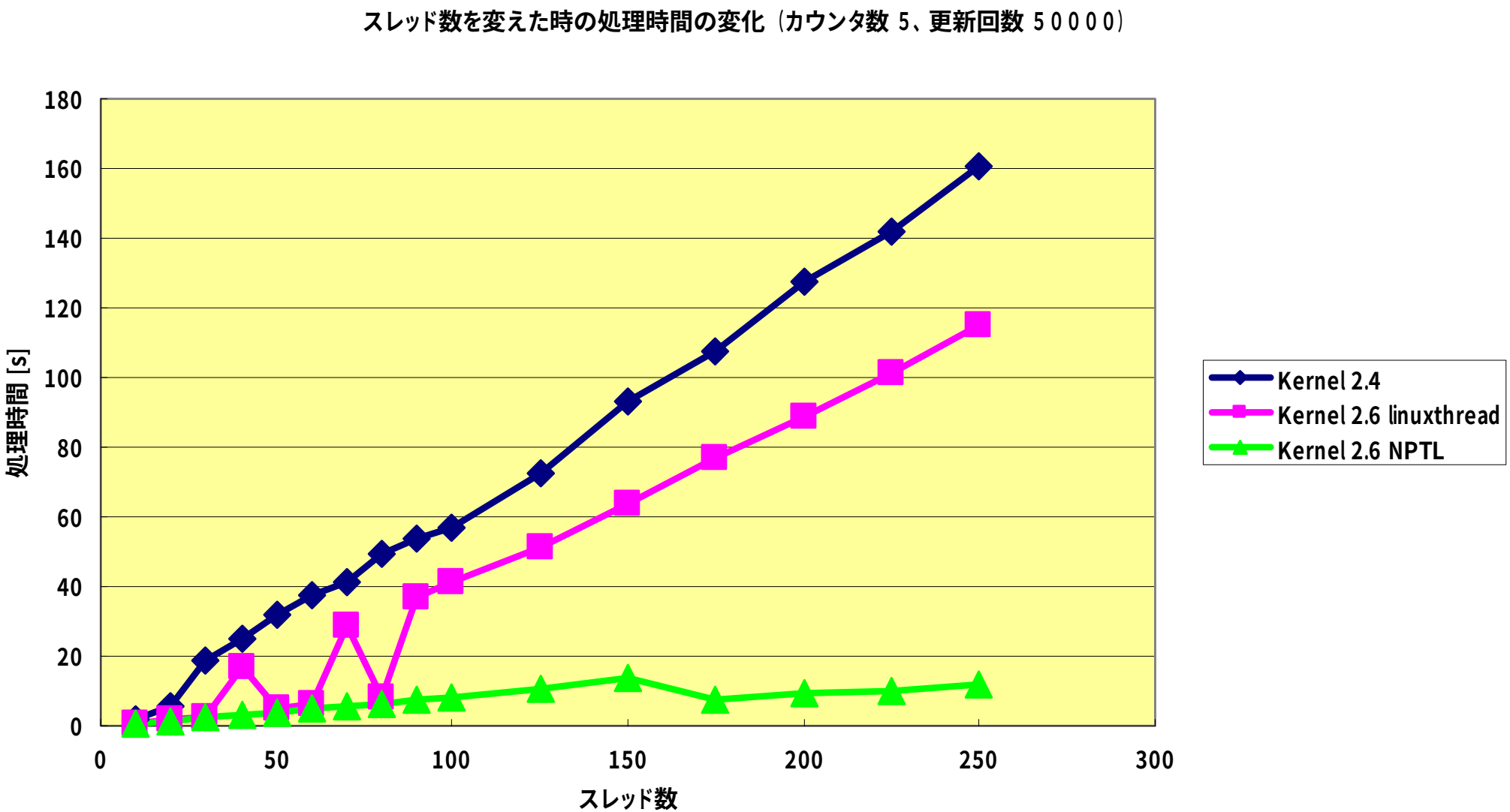
マルチスレッドでカウンタを更新し、その終了時間を測定



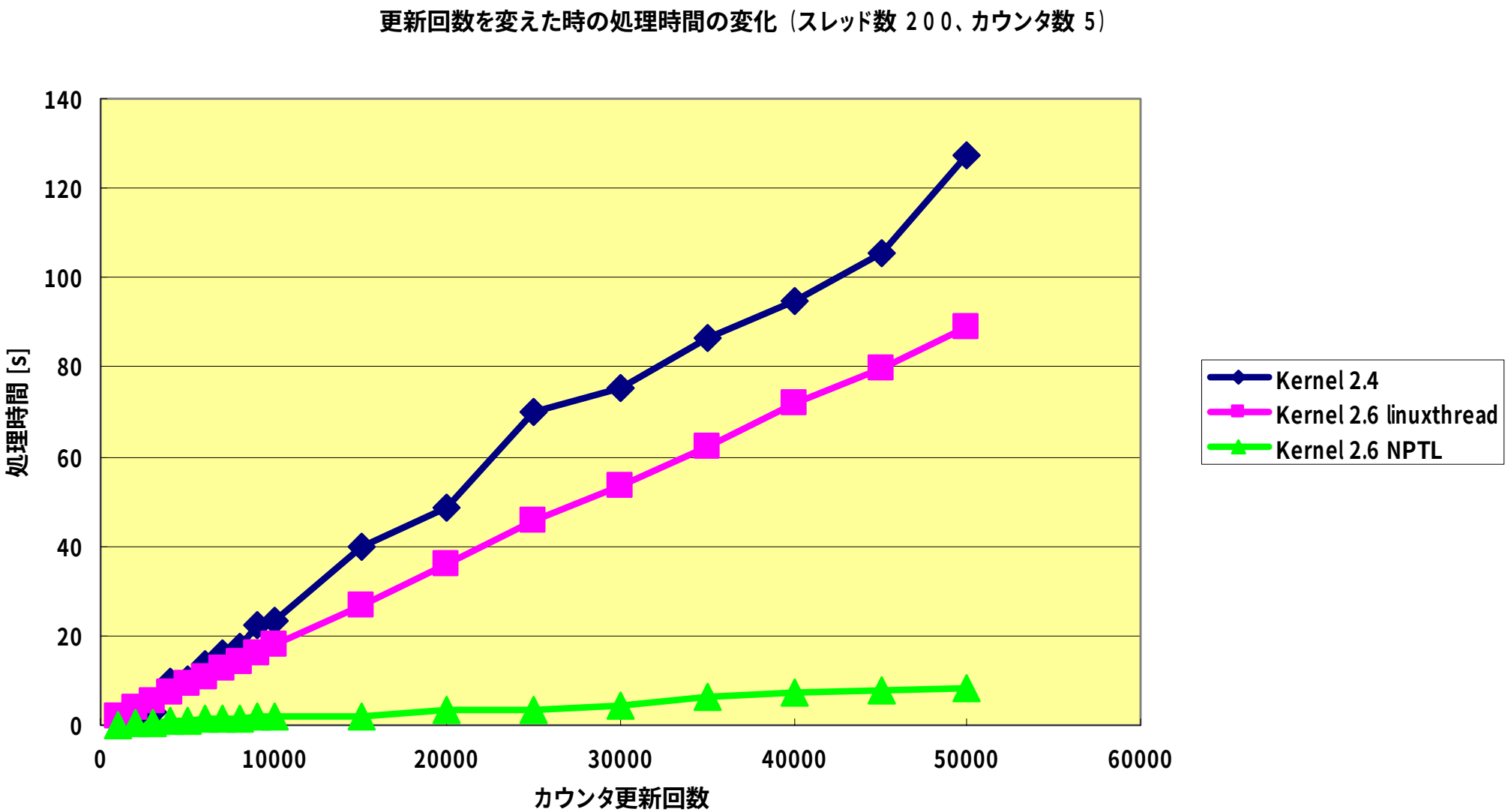
テスト内容

- スレッド数を変えた時の処理時間を測定
  - カウンタ数　5（固定）
  - 更新回数　50000回（固定）
  - スレッド数　10 ～ 250
- 更新回数を変えた時の処理時間を測定
  - カウンタ数　5（固定）
  - スレッド数　200（固定）
  - 更新回数　1000 ～ 50000
- 測定回数、測定結果の算出方法
  - 測定回数　1回
  - 測定結果の算出方法　全スレッド処理時間の平均値

測定結果：スレッド数を変えた時の処理時間



測定結果：更新回数を変えた時の処理時間



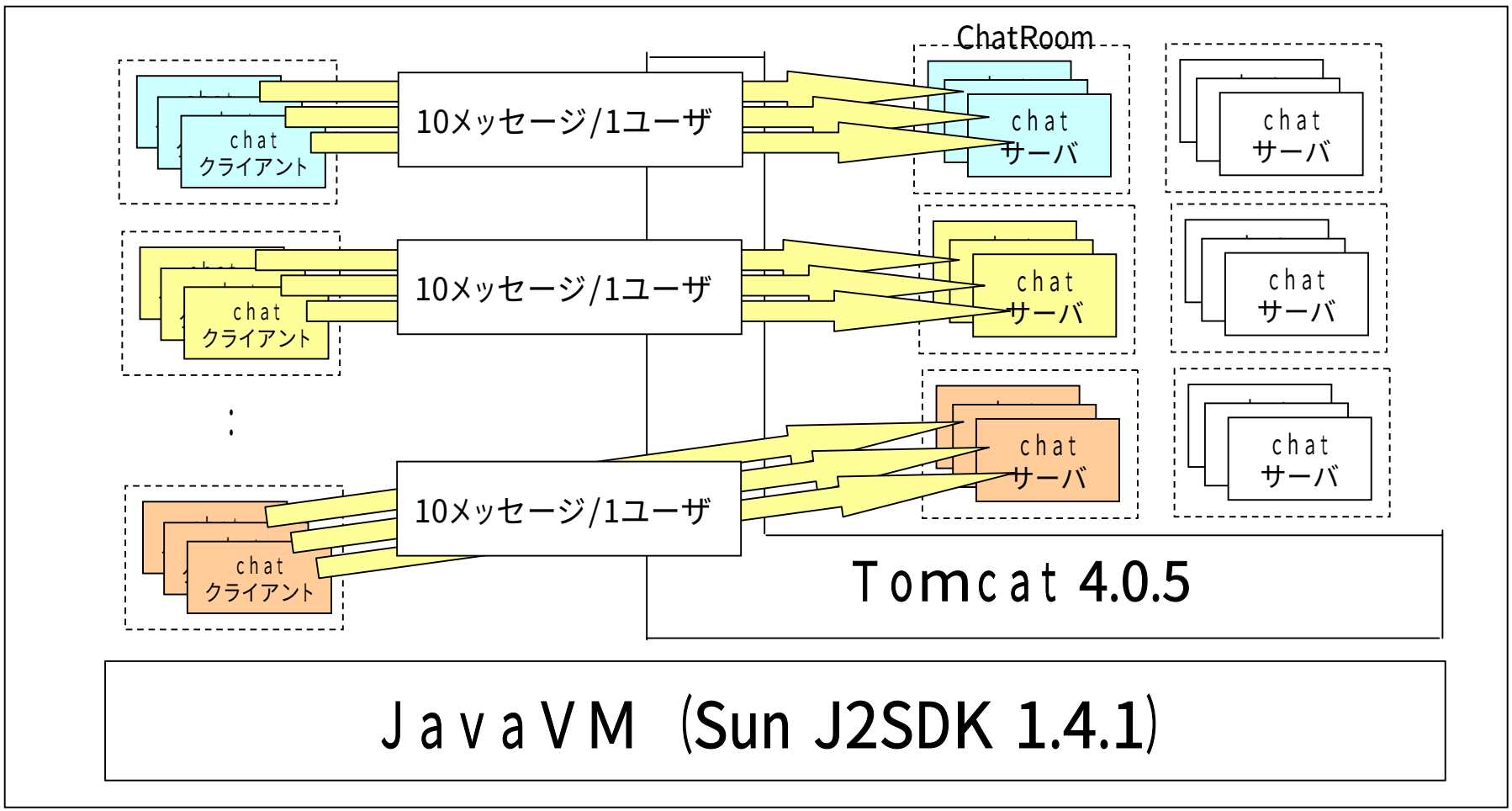
VolanoMarkとは

- Volano LLC社の VolanoChat製品のパフォーマンス測定用に開発されたソフトウェア
- Volanoチャットサーバ、およびチャットユーザをシミュレートするクライアントから構成される
- Javaで作成されており、現在はJavaVMのパフォーマンス測定として利用される
- バージョン 2.5.0.9

http://www.volano.com/benchmarks.html

テスト概要

20ユーザ（40スレッド）／チャットルーム



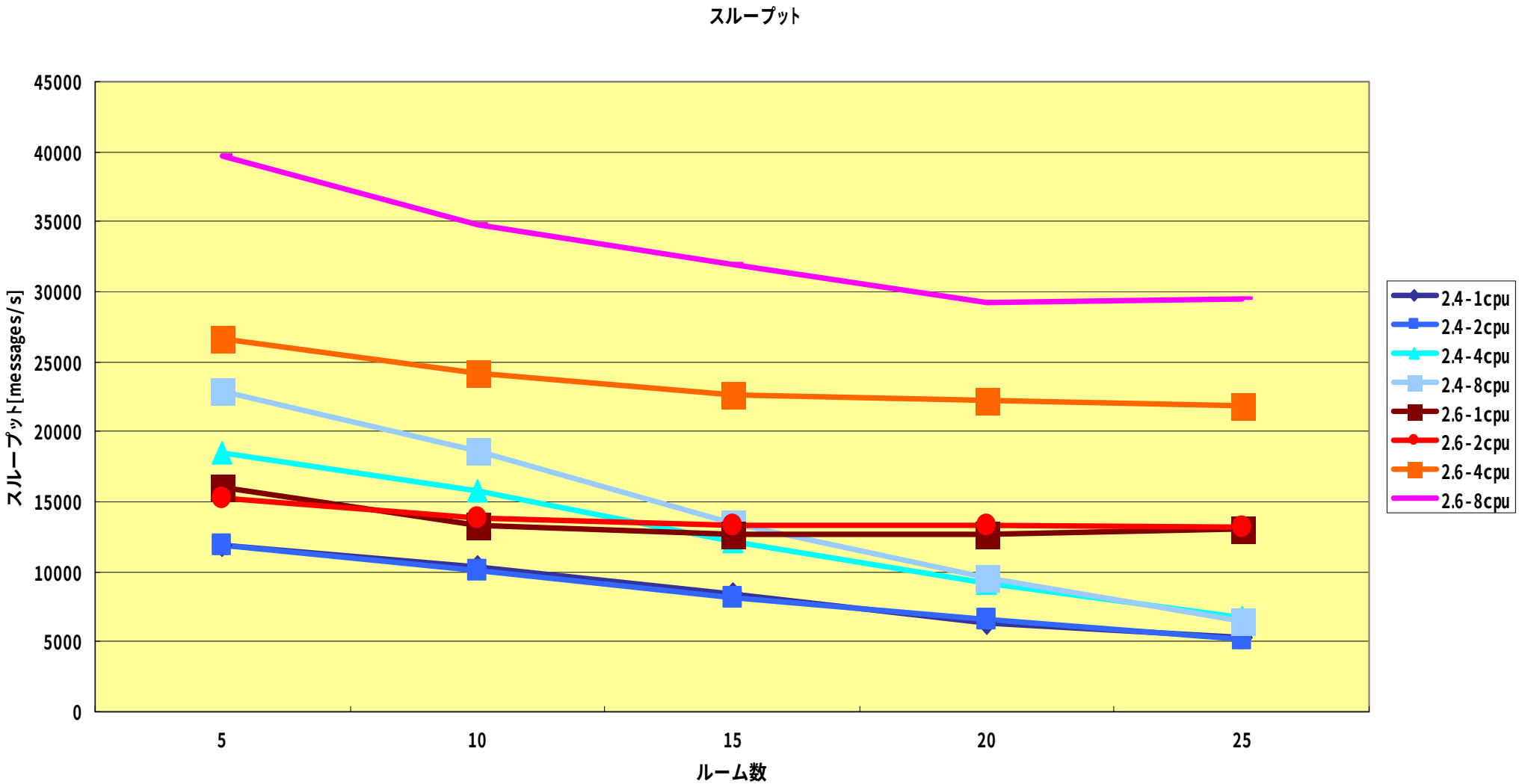
HP ProLiant DL560 G2

NTT COMWARE

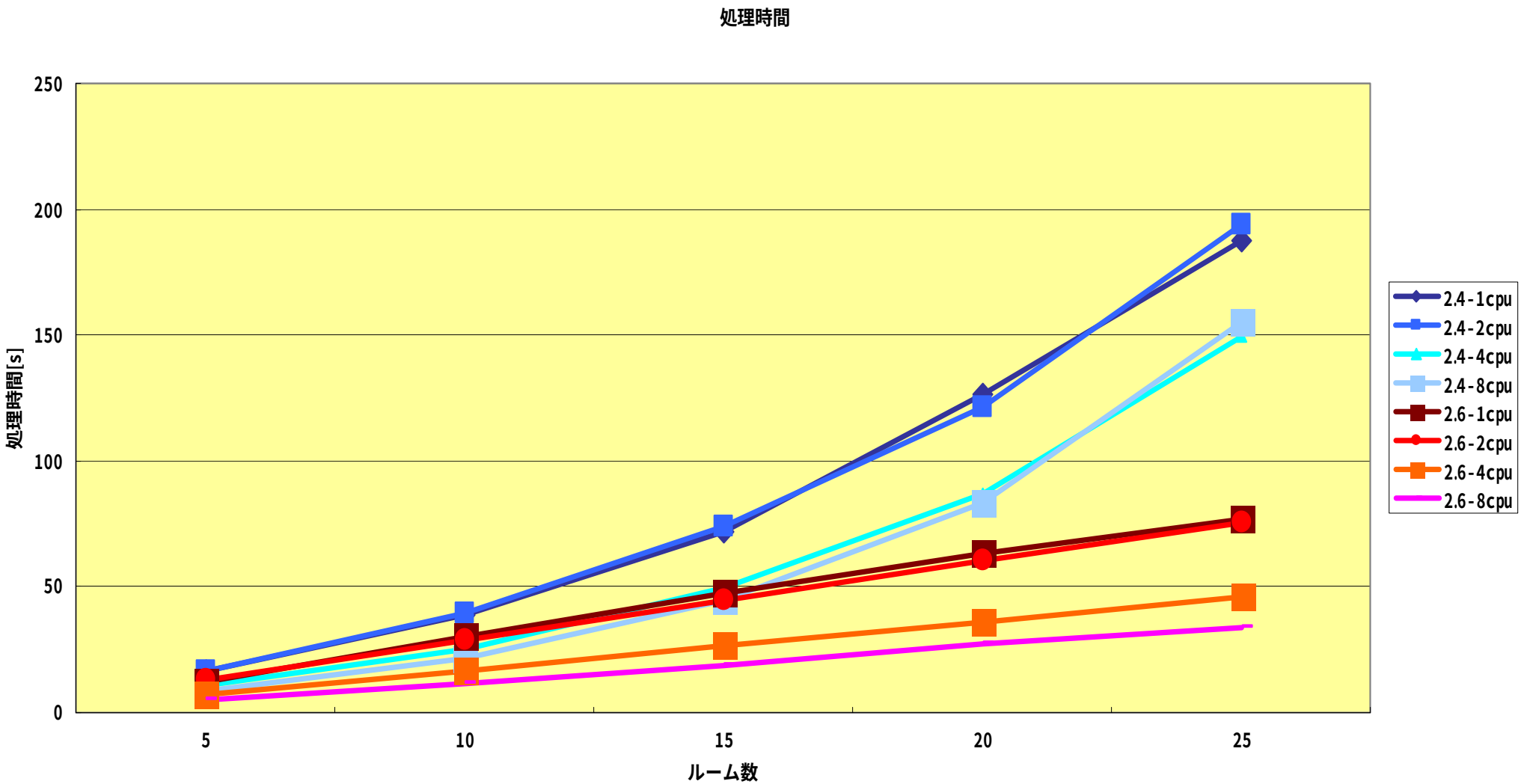
テスト内容

- ルーム数を変えた時のスループットと処理時間
  - ルーム数　　5～25
- 論理ＣＰＵ数（ＨＴ使用）を変えた時のスループットと処理時間
  - ＣＰＵ数　　1、2（1+HT）、4（2+HT）、8（4+HT）
- 測定回数、測定結果の算出方法
  - 測定回数　　3
  - 測定結果の算出方法　　3回測定した結果の平均値

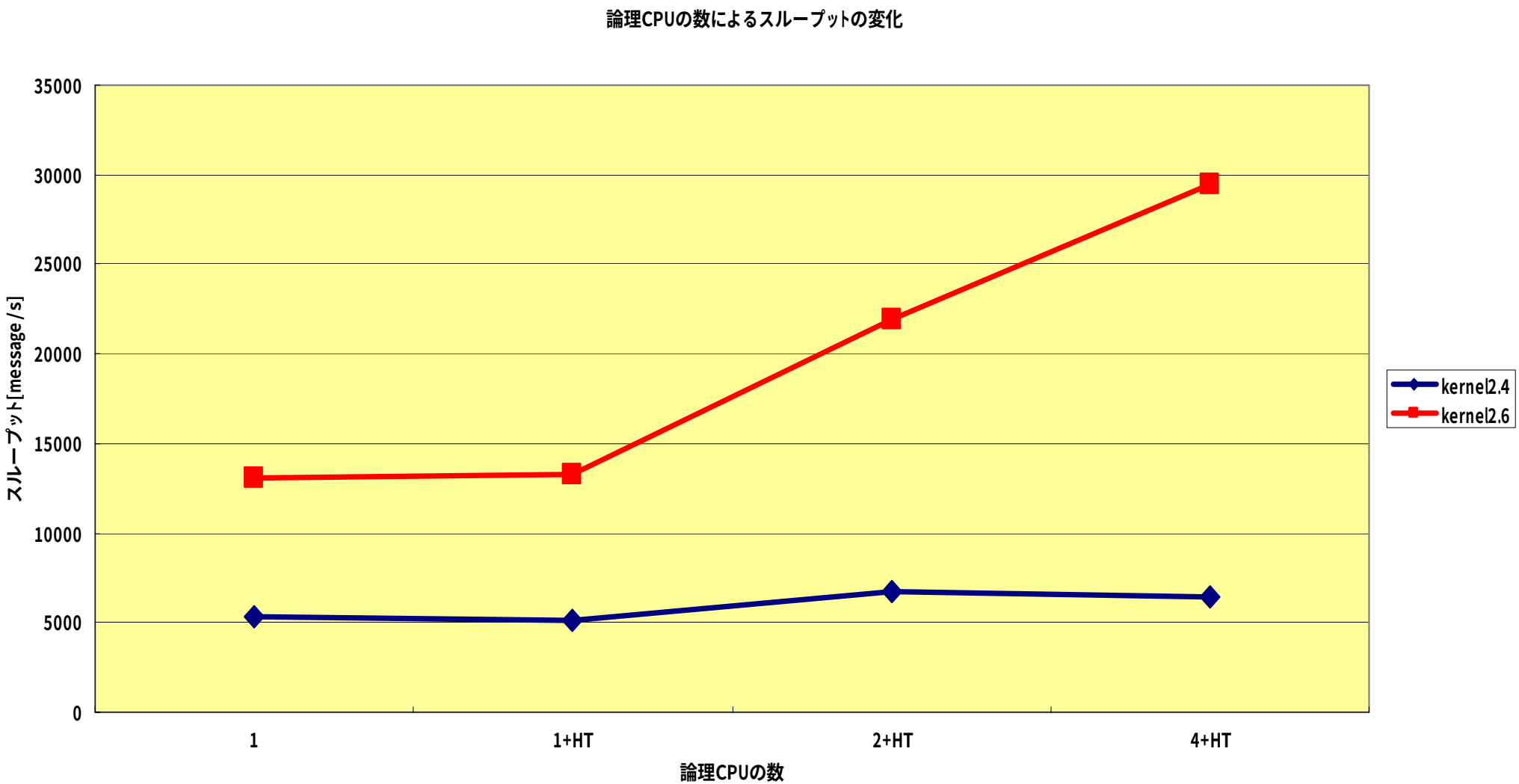
測定結果：スループット



測定結果：処理時間



分析： 論理ＣＰＵ数によるスループットの変化



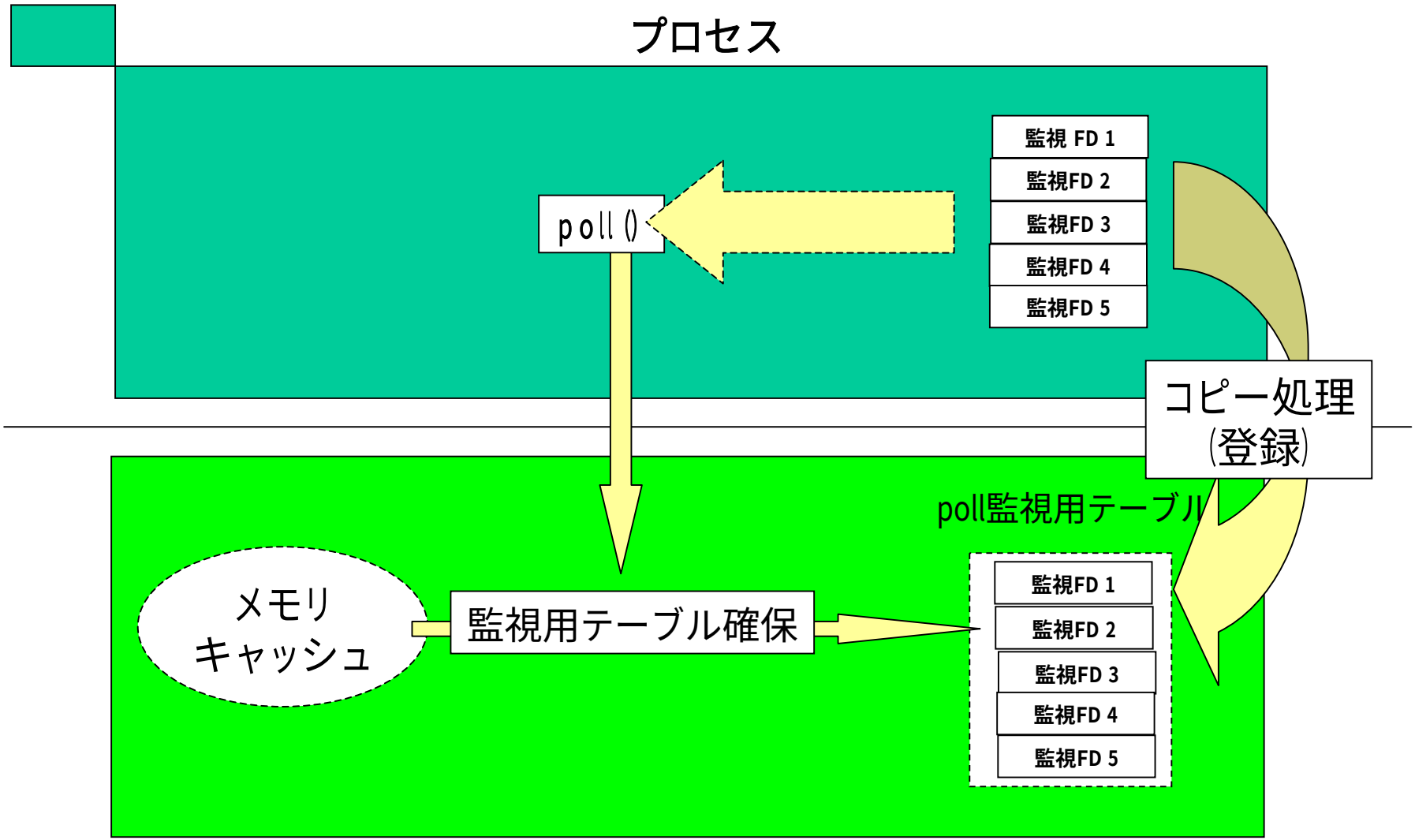
Webサーバ基盤の実力を探る
(eventpollの実力を探る)

― Webサーバベンチマークによる検証 ―

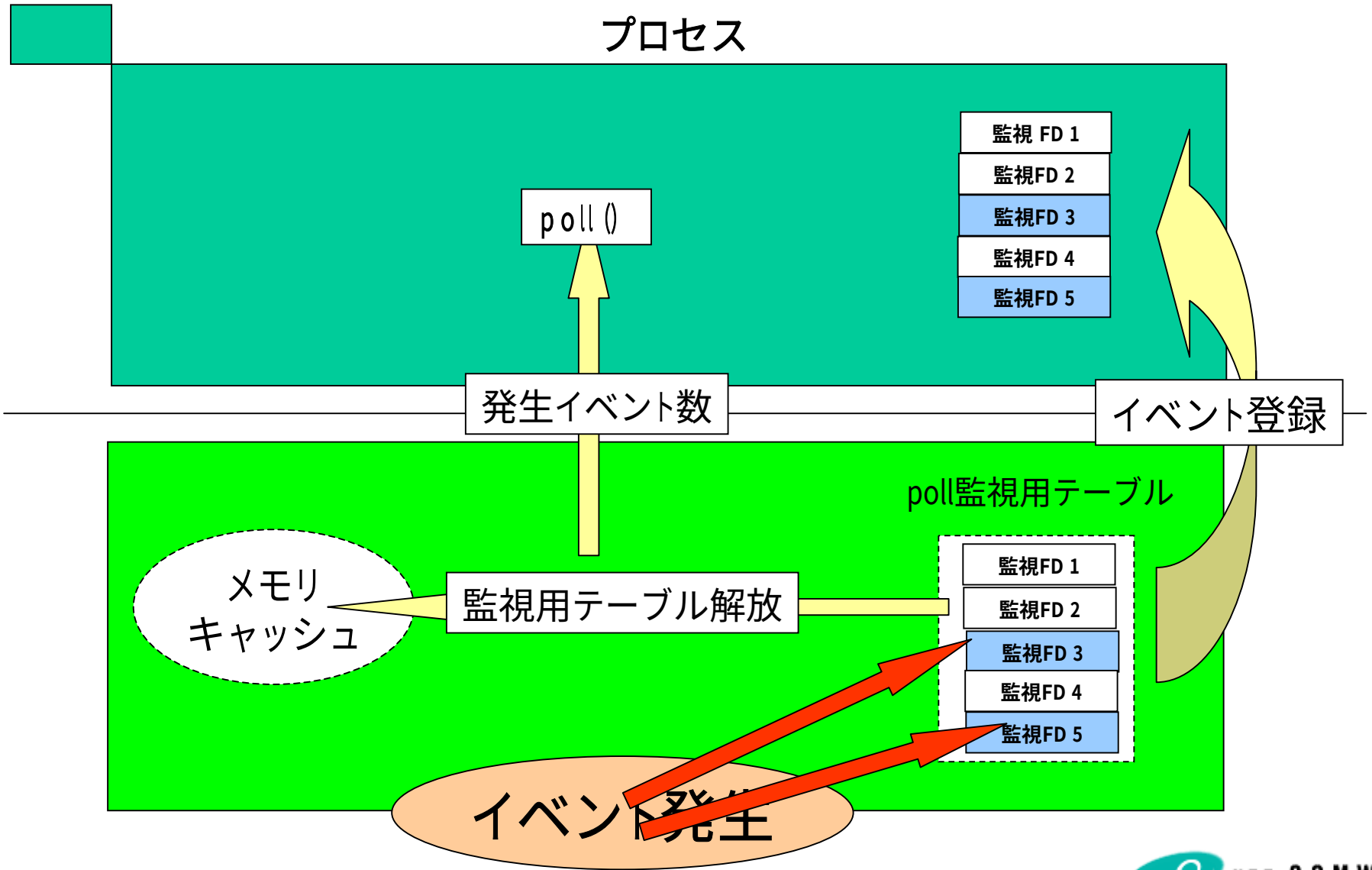
eventpoll (epoll) とは？

- コネクション (ディスクリプタ) の多重監視を行う poll/select にかわる新たなシステムコール
- poll/select に比べ、オーバーヘッドが少なく、多数のコネクションを効率的に監視可能
- コネクションの多重監視を行うインターネットサーバ (Webサーバ、Proxyサーバ) に効果大

pollによるソケットの多重監視 (1)

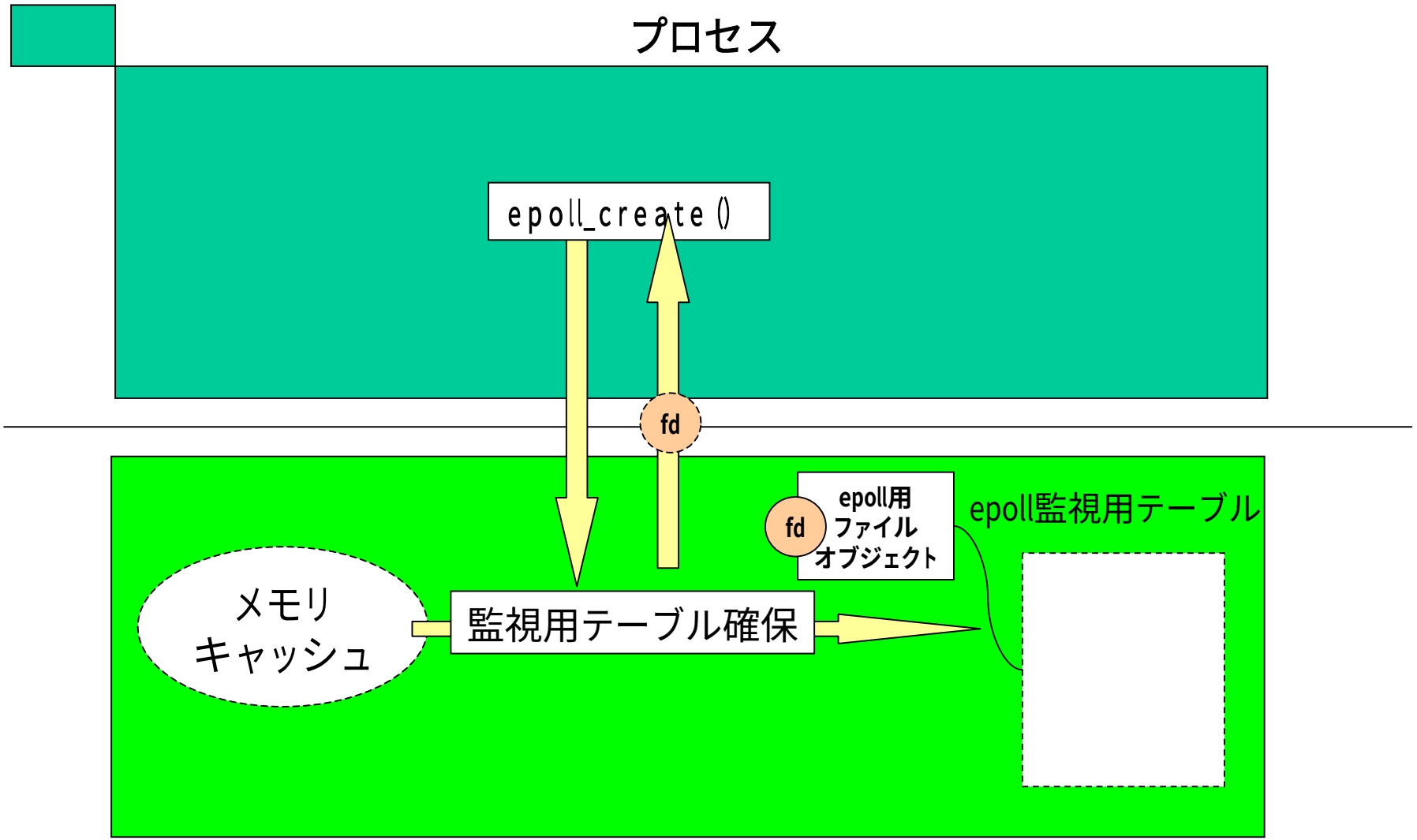


pollによるソケットの多重監視 (2)

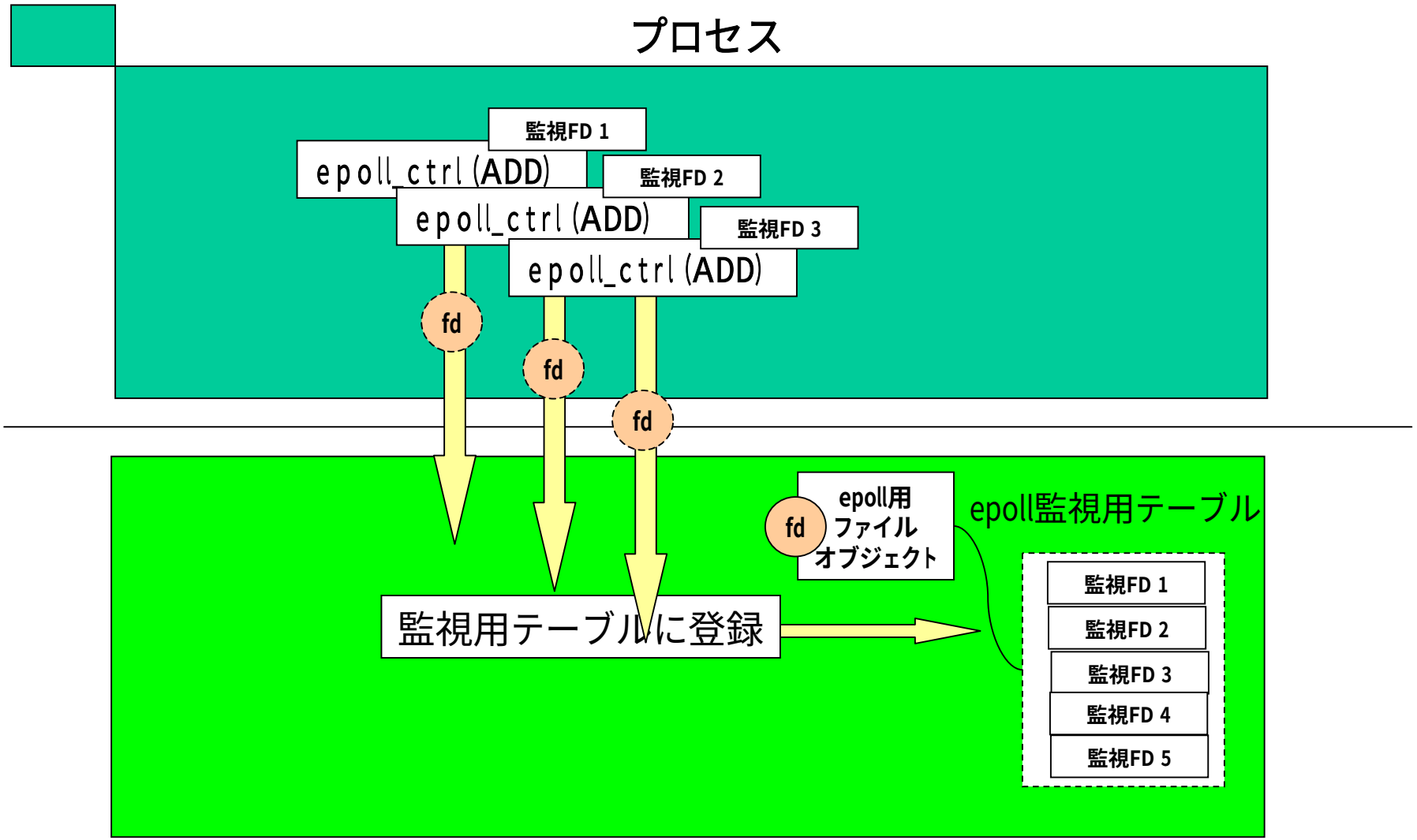


pollによるソケットの多重監視 (3)

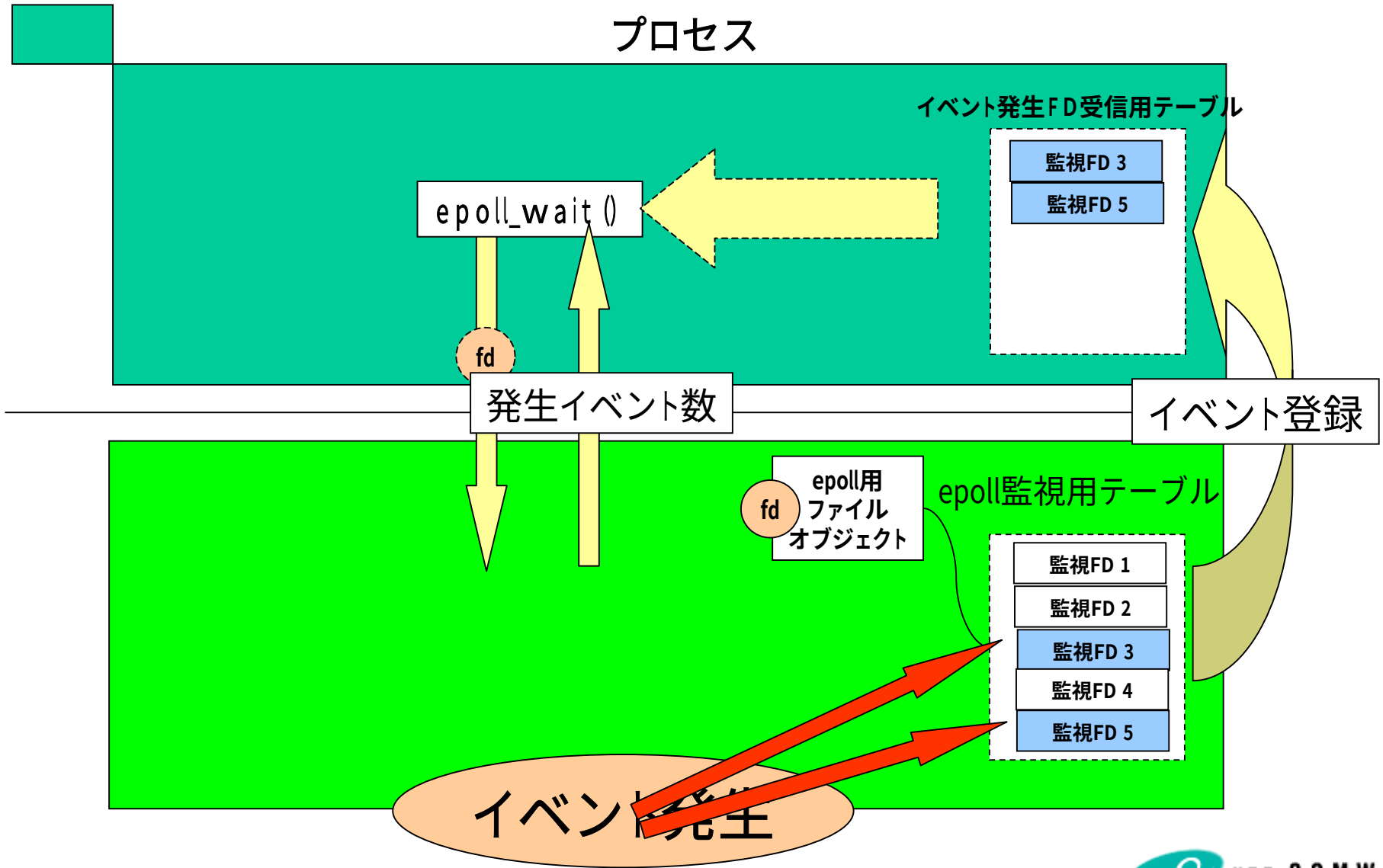
epollによるソケットの多重監視 (1)



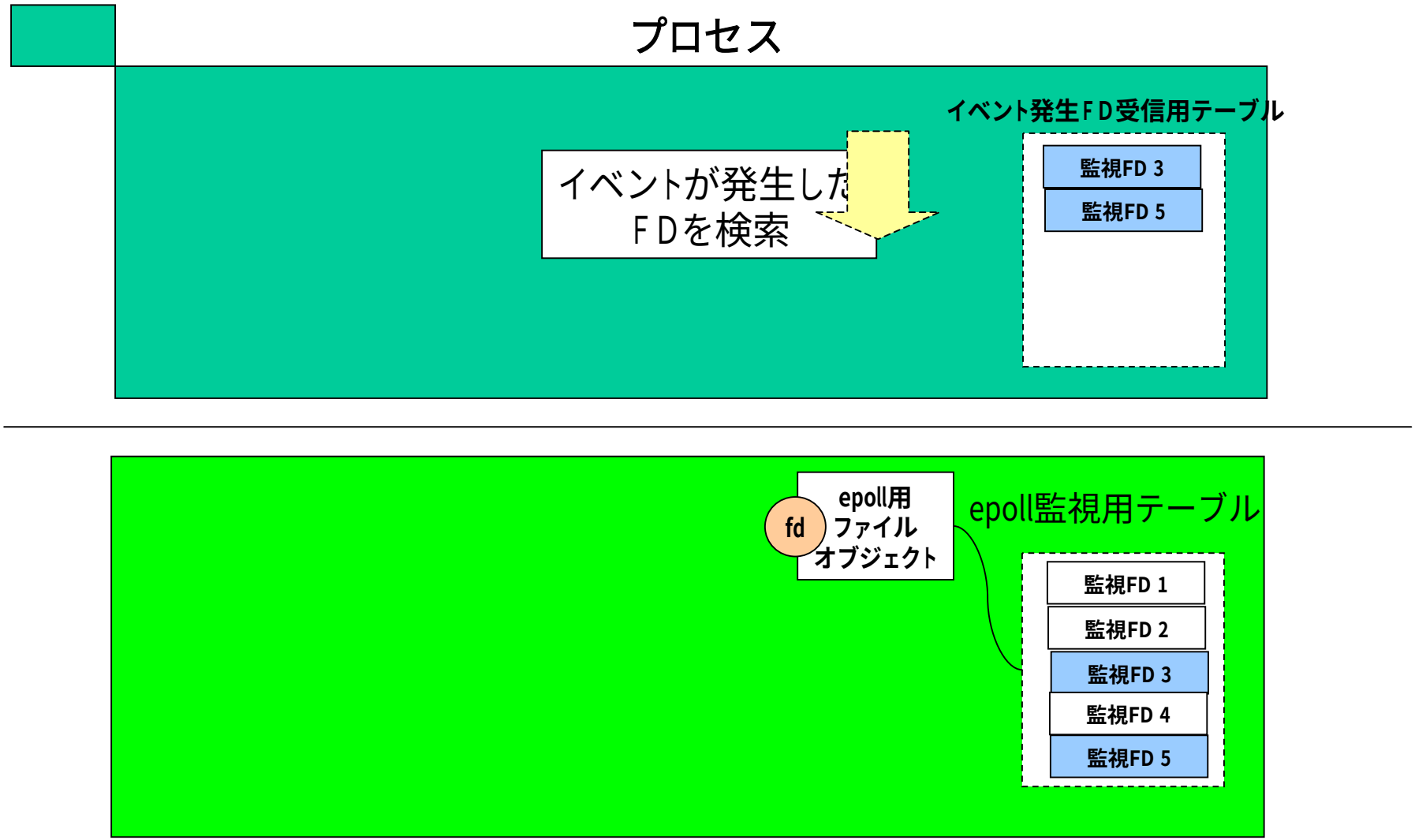
epollによるソケットの多重監視 (2)



epollによるソケットの多重監視 (3)



epollによるソケットの多重監視 (4)



Webサーバのベンチマーク検証

- userver、httperfによる性能測定

userverとは

- 実験用に設計されたシンプルなWebサーバ
- select/poll/epollをサポート
- 多彩なコマンドラインオプション
- バージョン 0.3.1

Userver Project
http://www.hpl.hp.com/research/linux/userver/index.php

httperfとは

- Webサーバ性能計測ツール
- HTTP／1．1対応
- 現実的なアクセスパターンのトラフィックを生成
  - 単純な連続アクセス
  - 一定間隔で複数リクエストを同時に送信
- バージョン　0．8

http://www.hpl.hp.com/personal/David_Mosberger/httperf.html

httperfとは：テストパラメータと測定内容

テストパラメータ

要求接続レート	1秒毎に接続しにいくコネクション数
TCPコネクション数の合計	接続するＴＣＰコネクションの合計数
TCPリクエストの合計	リクエストの合計数
バースト間隔	バースト間隔の時間
バースト当りのリクエスト数	バースト間隔毎にまとめて送信するリクエスト数

測定内容

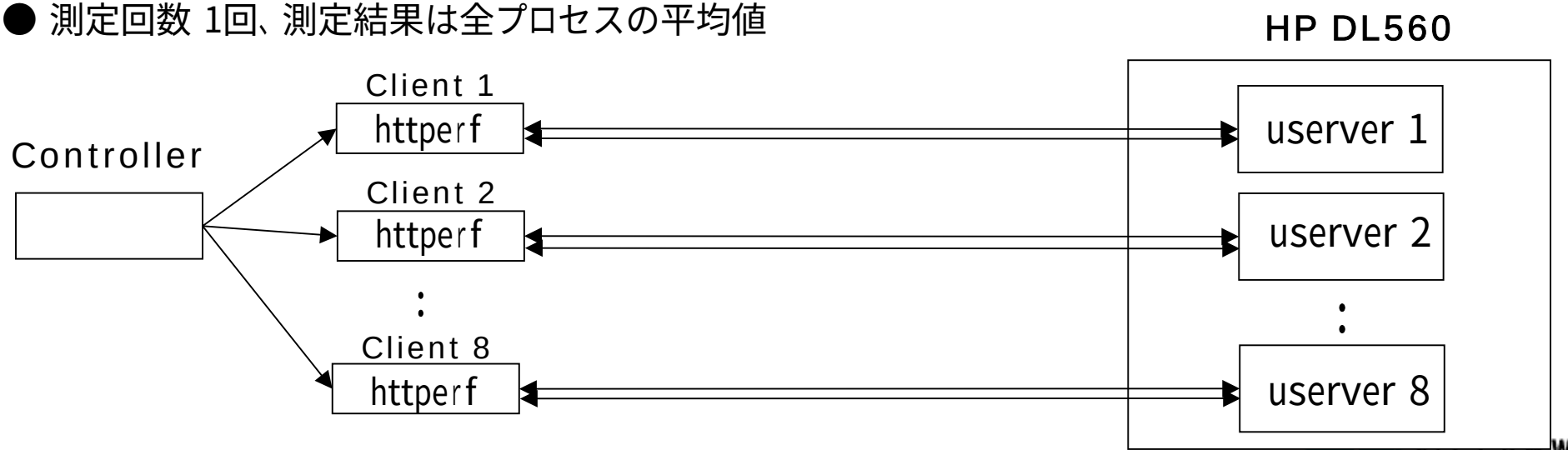
リプライレート	1秒当りの応答メッセージ数
レスポンスタイム	応答メッセージのレスポンス時間

テスト内容：8サーバ起動

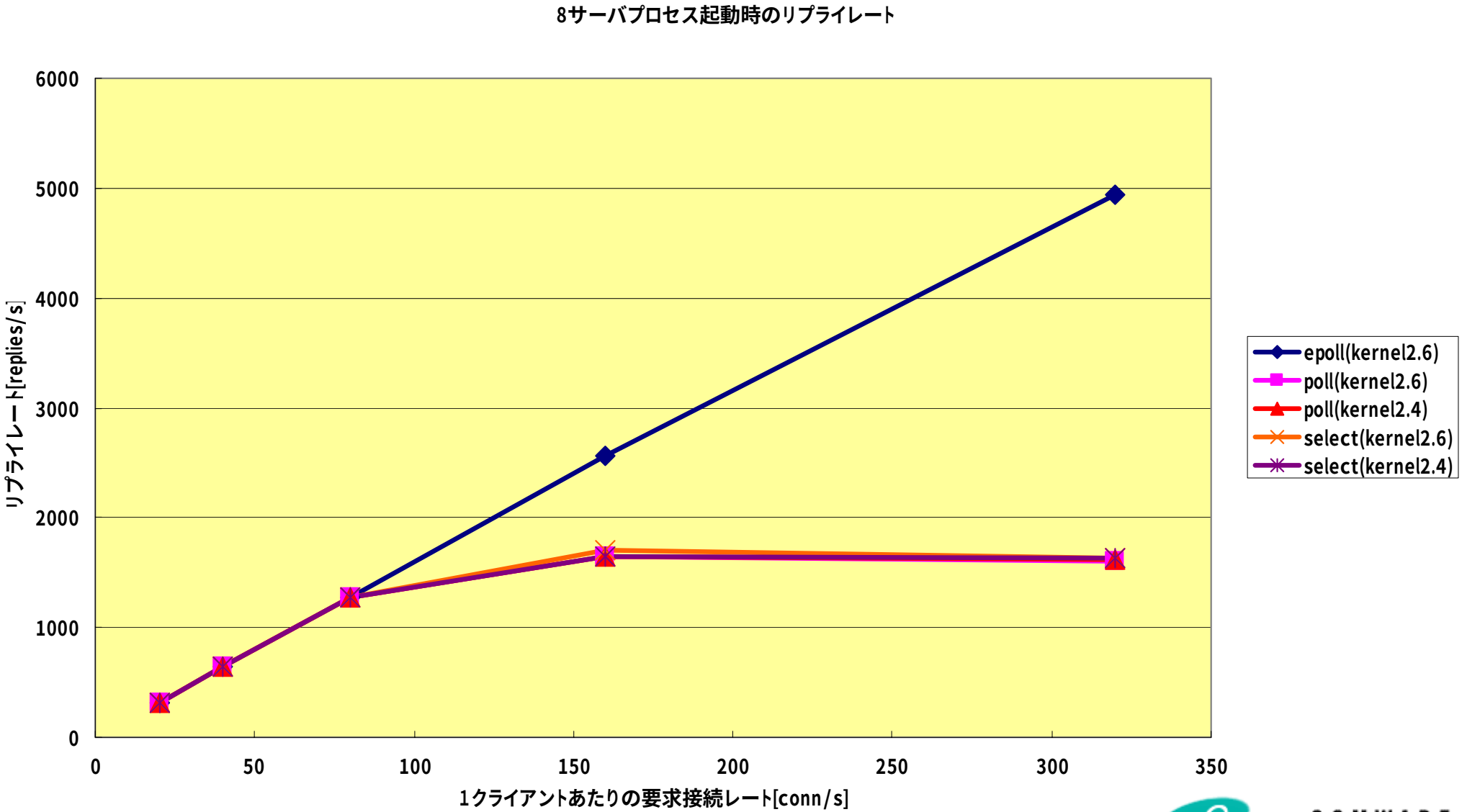
● 1クライアント（httperf）あたりのテストパラメータ

テストパラメータ	1	2	3	4	5
要求接続レート（数／秒）	20	40	80	160	320
TCPコネクションの合計	600	1200	2400	4800	9600
リクエスト数の合計	6	6	6	6	6
バースト間隔（秒）	30	30	30	30	30
バースト当りのリクエスト数	2	2	2	2	2

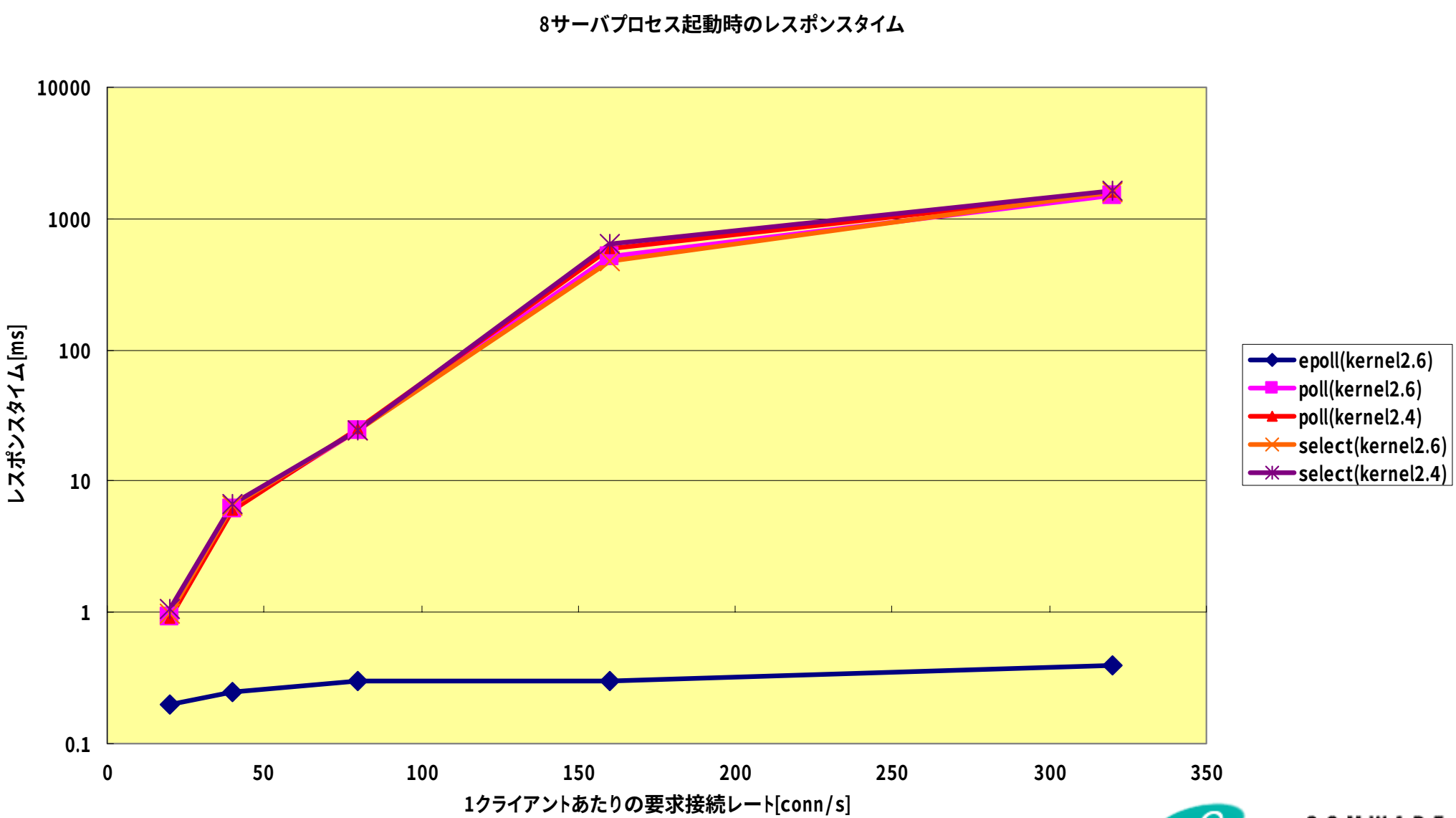
● 1回あたりの測定時間は 60秒間、HTMLファイルサイズは 500バイト
● 測定回数 1回、測定結果は全プロセスの平均値



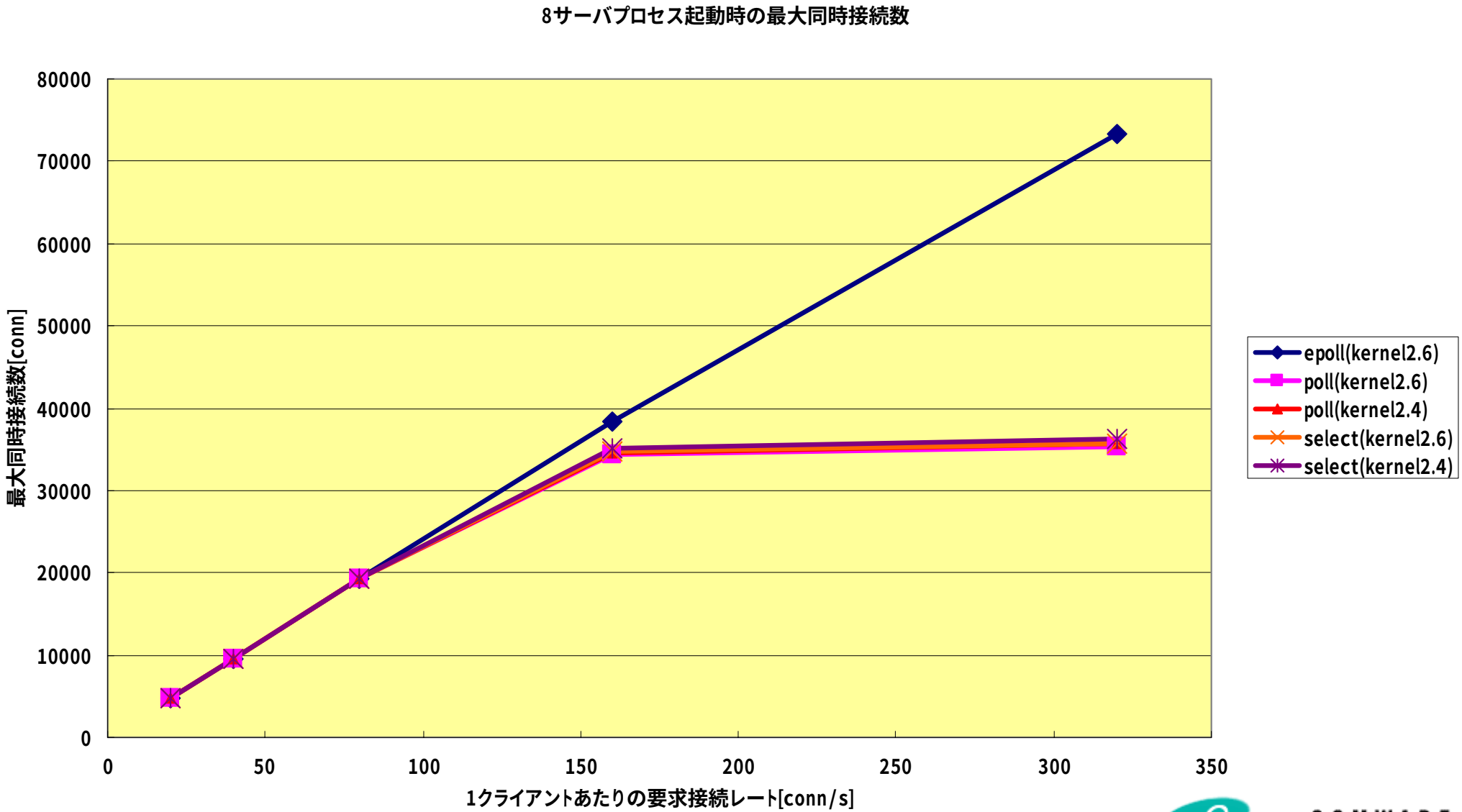
測定結果：8サーバ：リプライレート



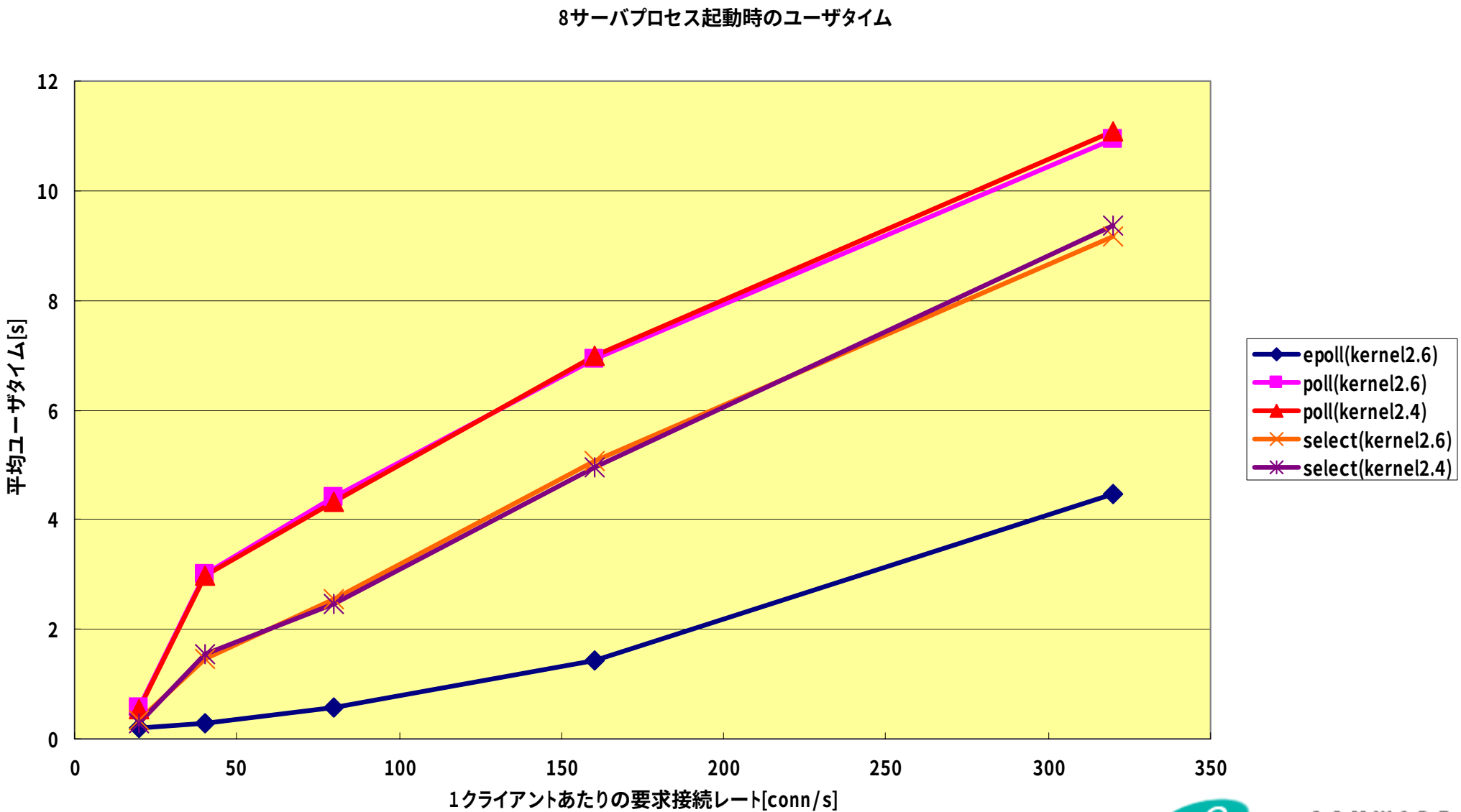
測定結果：8サーバ：レスポンスタイム



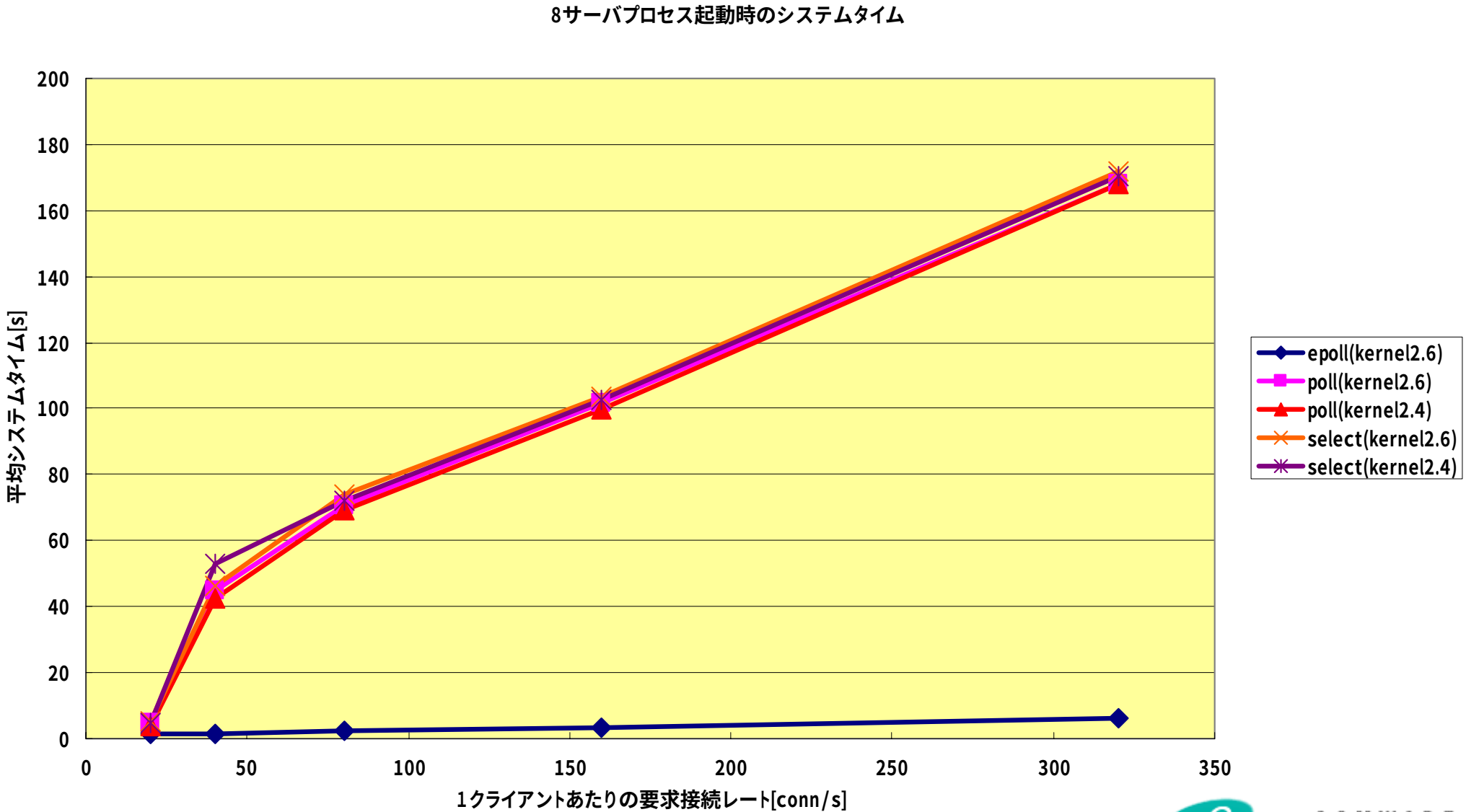
測定結果：8サーバ：同時接続数



分析：8サーバ：ユーザ時間



分析：8サーバ：システム時間



まとめ

- 期待していた以上の性能
- その他のハイエンドシステム向け機能強化にも期待大
  - マルチプロセッサ対応強化
  - ６４ビットＣＰＵ対応の拡充
  - ストレージ機能強化
    - ジャーナリングFS、動的ファイルシステム拡張
  - データベース向け機能強化
    - Direct I/O、AIO、readv/writev、HugeTLB
  - 可用性の向上
    - スナップショット、ハードウェアホットプラグ

ご静聴ありがとうございました