

M32Rマルチプロセッサへの Linux SMPカーネルの実装

○藤原隼人[†], 山本整[‡], 高田浩和[†], 坂本圭[†], 作川守[†], 近藤弘郁[†]

[†]株式会社ルネサステクノロジ

[‡]三菱電機株式会社

発表内容

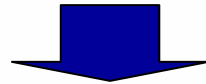
- 背景目的
- M32Rシングルチップ マルチプロセッサ
- Linux/M32Rへの
マルチプロセッサ機能の実装
- 性能評価
- まとめ
- Linux/M32Rプロジェクト紹介



背景・目的

組み込み機器

- 高性能化の要求により、CPUの性能向上が求められている
- 低消費電力も要求され、動作周波数を上げることは難しい



- マルチプロセッサ技術が注目されている
 - 周波数をあげずに性能を向上
 - 特にシングルチップマルチプロセッサは低消費電力を実現できる



M32R用LinuxでSMPサポート



M32Rシングルチップ マルチプロセッサ



- M32Rプロセッサ

- ルネサステクノロジ・オリジナルの32ビットRISCプロセッサ・コア
- 組み込み用途向けでコンパクト
 - シングルチップマルチプロセッサに適している



組み込み機器でもマルチプロセッサ技術は重要となる

- M32Rシングルチップマルチプロセッサ



M32Rシングルチップ マルチプロセッサ

- M32700

- CPUコア: 2基

- Dual-issue
 - 7stage pipe

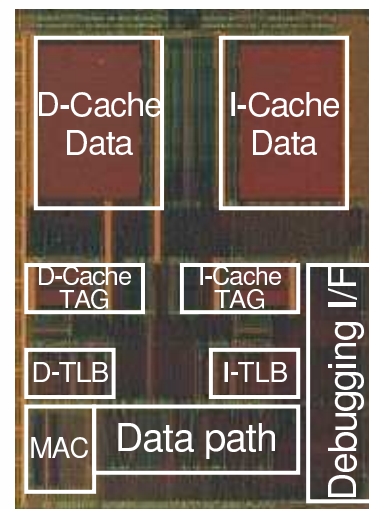
- 内蔵SRAM: 512KB

- キャッシュ:

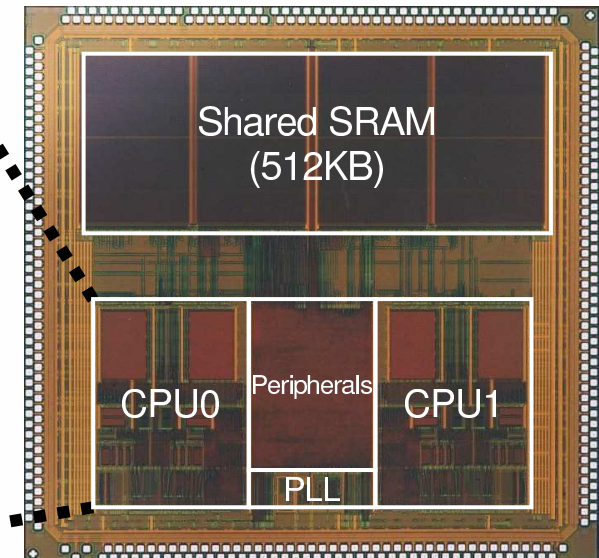
- 命令8KB
 - データ8KB

- TLB:

- 命令 32エントリ
 - データ 32エントリ



M32R CPU Core



Chip Photomicrograph

Satoshi Kaneko, et al.: "A 600 MHz single-chip multiprocessor with 4.8 GB/s internal shared pipelined bus and 512 kB internal memory," ISSCC Digest of Technical Papers, pp. 254-255 (2003).



M32Rシングルチップ マルチプロセッサ

- マルチプロセッサとして必要な機能をM32Rへ追加
 - 排他制御命令
 - ロックつきロード・ロック解除つきストア
 - CPU間割り込み (IPI: Inter-Processor Interrupt)
 - 他のCPUに割り込みを発生させる
 - キャッシュコヒーレンシ
 - 各CPUのキャッシュの一貫性を保つ



Linux/M32R SMPサポートの実装

SMPをサポートするための機能

- 同期機構
 - クリティカルリージョンを保護するために利用
 - スピンロック
 - セマフォ
 - アトミック操作
- CPU間通信
 - 他のCPUへ処理を要求
- 起動処理
 - システム中の全CPUを起動



マルチプロセッサ機能の実装

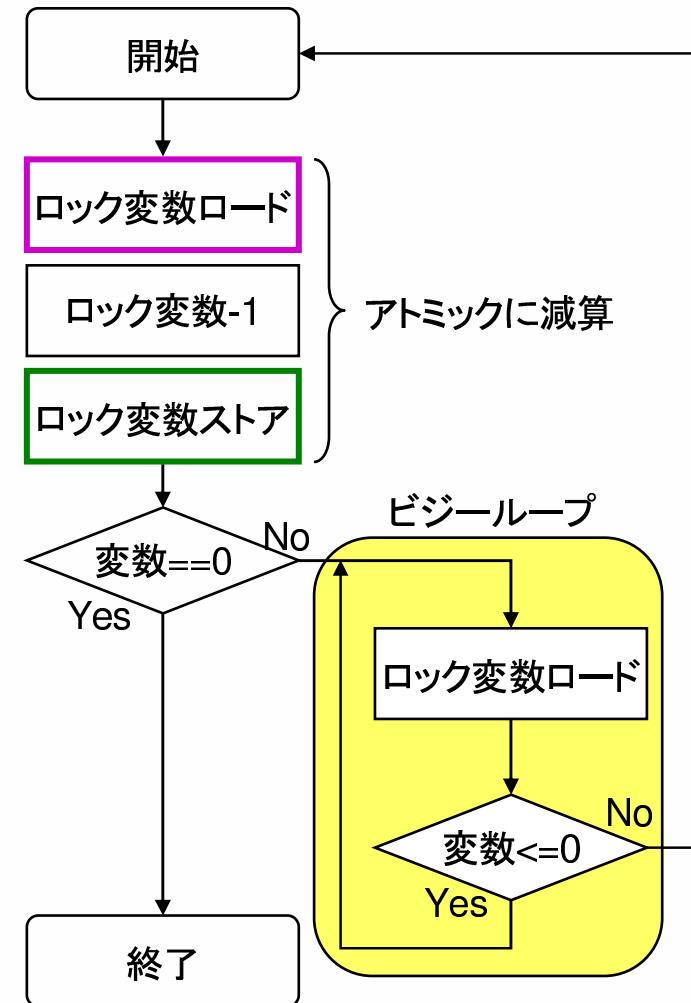
• 同期機構

–クリティカルリージョンを保護

割り込み禁止にだけでは
他のCPUから同じ資源にアク
セスされる可能性がある

• スピンロック

- ロック変数==1: 開放状態
- ロック変数<=0: ロック状態
- すでにロックがかかっているならばスピンして解除を待つ
- アンロックはロック変数に1を書く
- ロックつきロードストア命令
LOCK/UNLOCKを用いて実装

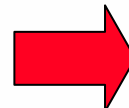


マルチプロセッサ機能の実装

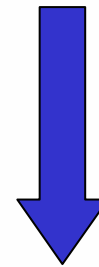
- ビジーループ部分を別セクションに移動

- 命令キャッシュのヒット率を改善
- LmbenchのLocal Communication latenciesテストで約10%の性能向上

```
1 : mvfc  r5 , psw
   clrpsw #0x40  # disable int.
   lock   r4 , @r0 # }
   addi   r4 , #-1 # } atomic op.
   unlock r4 , @r0 # }
   mvtc   r5 , psw
   beqz   r4 , 3f
2 : ld    r4 , @r0 # } busy loop
   blez   r4 , 2b  # }
   bra    1b
3 : XXX           # } critical
   ⋮             # } region
```



```
1 : mvfc  r5 , psw
   clrpsw #0x40  # disable int.
   lock   r4 , @r0 # }
   addi   r4 , #-1 # } atomic op.
   unlock r4 , @r0 # }
   mvtc   r5 , psw
   bnez   r4 , 2f
```



```
.subsection 1
.text .lock
2 : ld    r4 , @r0 # } busy loop
   blez   r4 , 2b  # }
   bra    1b
```

```
.previous
XXX           # } critical
   ⋮             # } region
```



マルチプロセッサ機能の実装

• CPU間通信

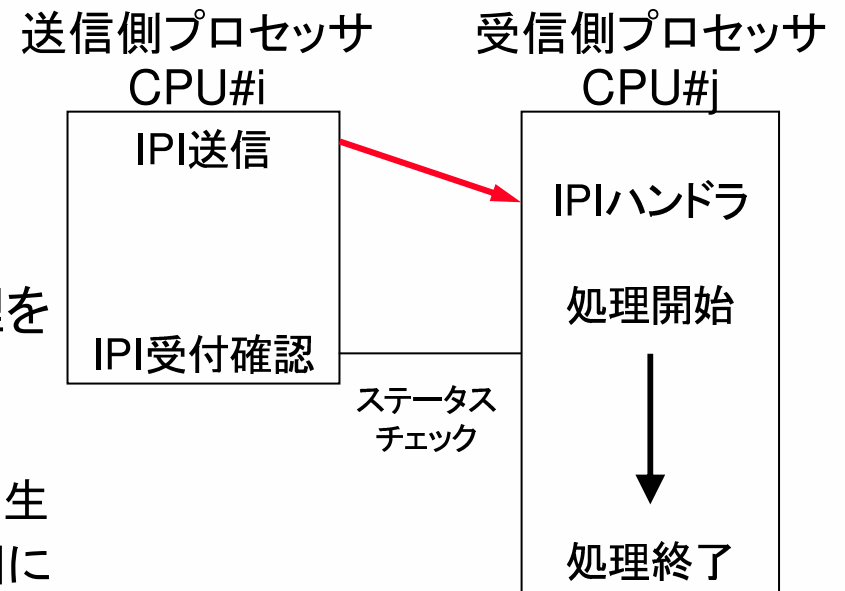
–他のCPUへ処理を要求

- 再スケジューリング
- キャッシュフラッシュ
- TLBフラッシュ

など

–CPU間割り込み(IPI)を用いて処理を要求

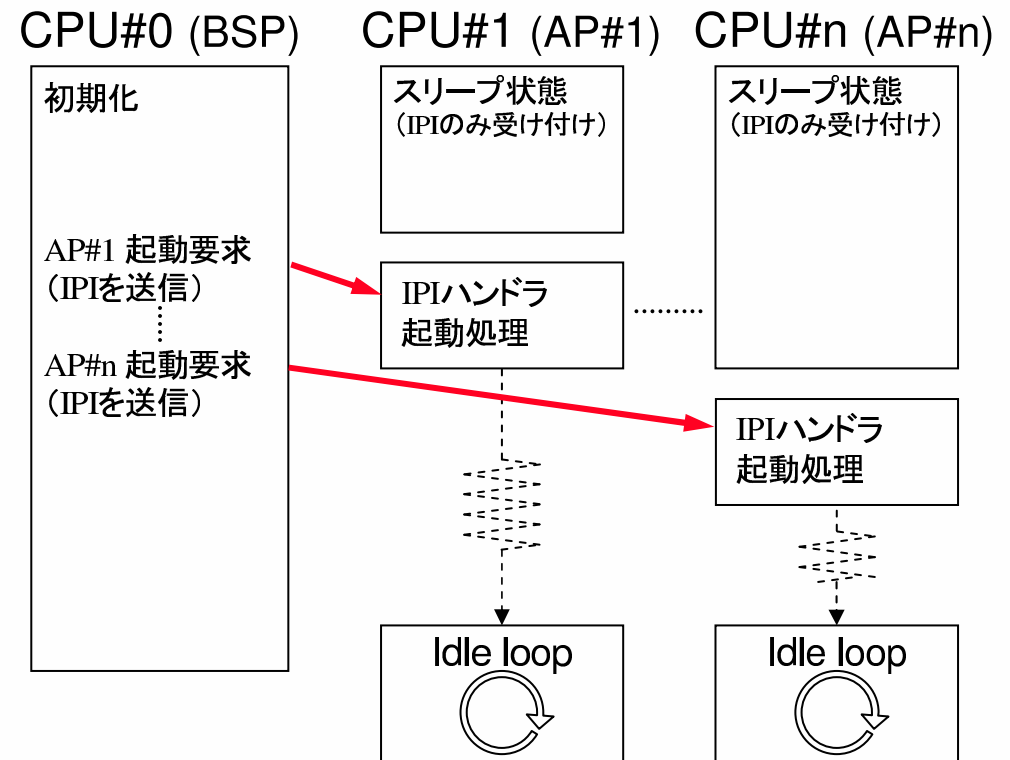
1. 送信側プロセッサがIPIを送信
2. 受信側プロセッサで割り込みが発生
3. 受信側プロセッサは割り込み要因に応じた割り込みハンドラを実行し、処理を行なう
4. 送信側プロセッサは割り込みが受け付けられた事を確認して終了



マルチプロセッサ機能の実装

• 起動処理

- システムの全CPUを起動する
- BSPがIPIを使用してAPを起動
 1. BSPが起動しハードウェアおよびLinuxの初期化を行う
 2. BSPが各APに対して起動を要求するIPIを発行する
 3. APが動作を開始し、初期化後アイドルループに入る



BSP: Bootstrap Processor
AP: Application Processor



性能評価

- LMbench

- OS・ハードウェアの基本的な性能を測定

- Bandwidth benchmarks
 - キャッシュやメモリ, 通信の帯域幅を測定
 - Latency benchmarks
 - コンテキストスイッチ, ネットワーク, システムコール, メモリ等の遅延を測定

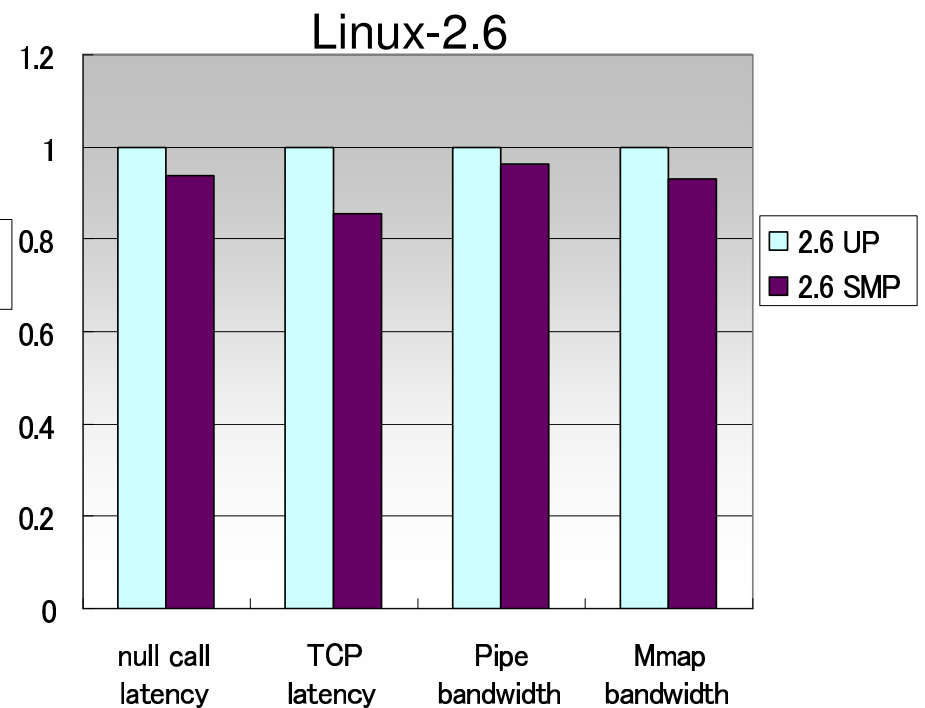
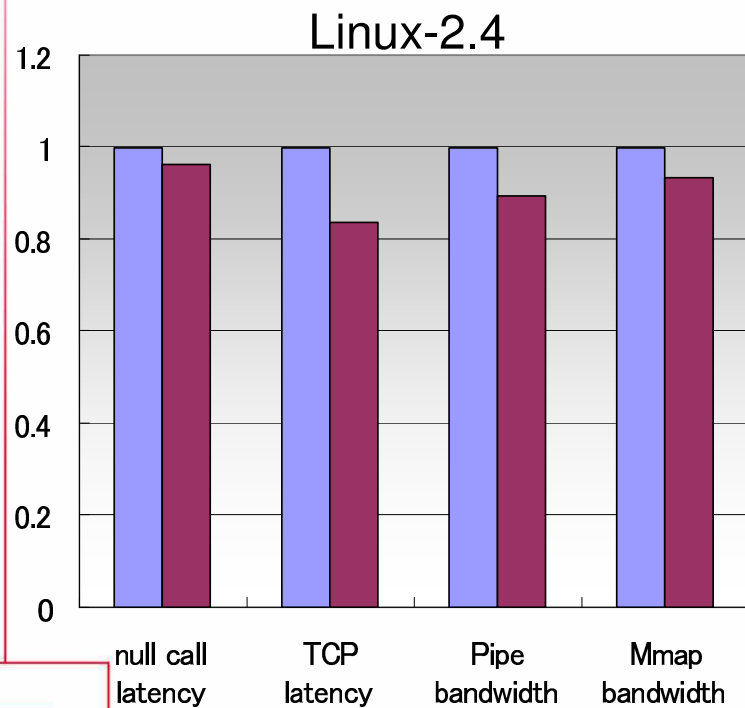
- 測定環境

- CPU: M32700 (single-chip-multiprocessor)
 - freq: CPU 200MHz, BUS 100MHz
 - OS: Linux-2.4.25, Linux-2.6.4 UP/SMP
 - Benchmark: LMbench 3.0-a3



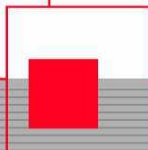
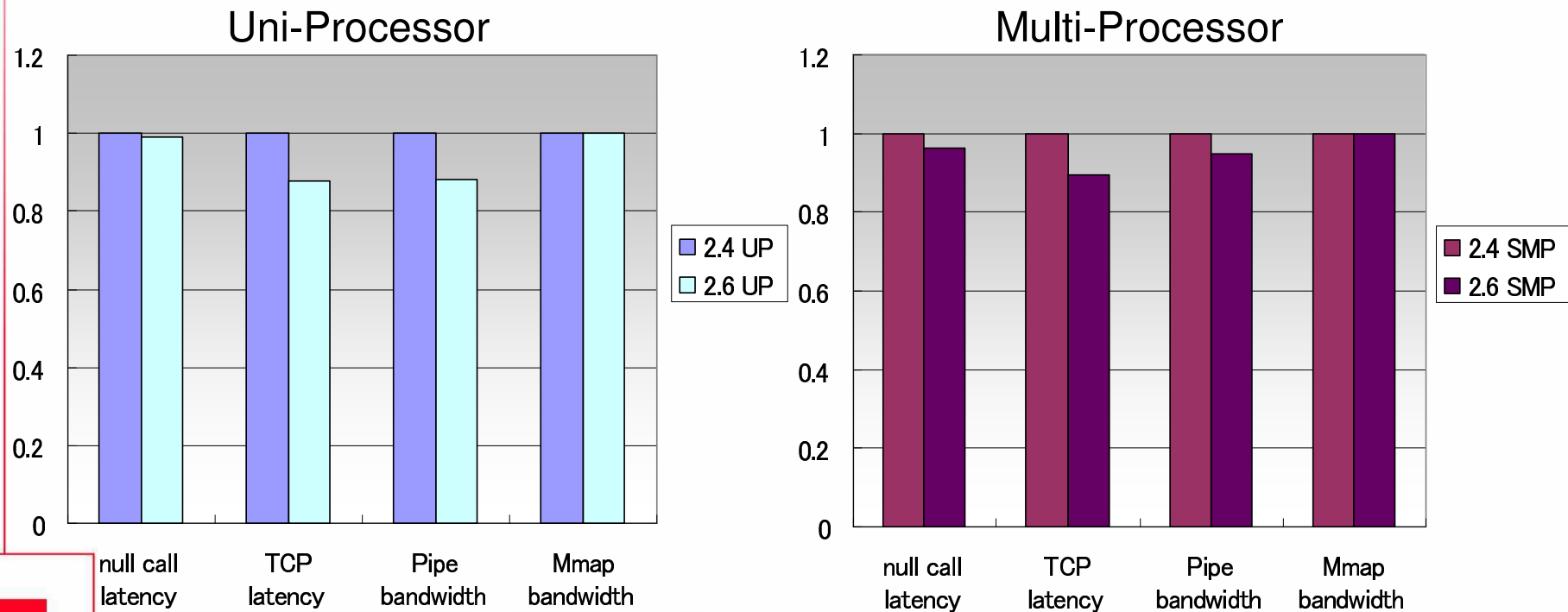
性能評価 - Lmbench UP vs. SMP

- SMPでのオーバーヘッドを評価
- UPに対するSMPの速度低下は10%~20%で抑えられている



性能評価 - LMbench 2.4 vs. 2.6

- カーネルバージョンによる性能比較
- 2.6カーネルでは2.4に比べ速度が低下している項目が見受けられ、原因の調査・チューニングが必要



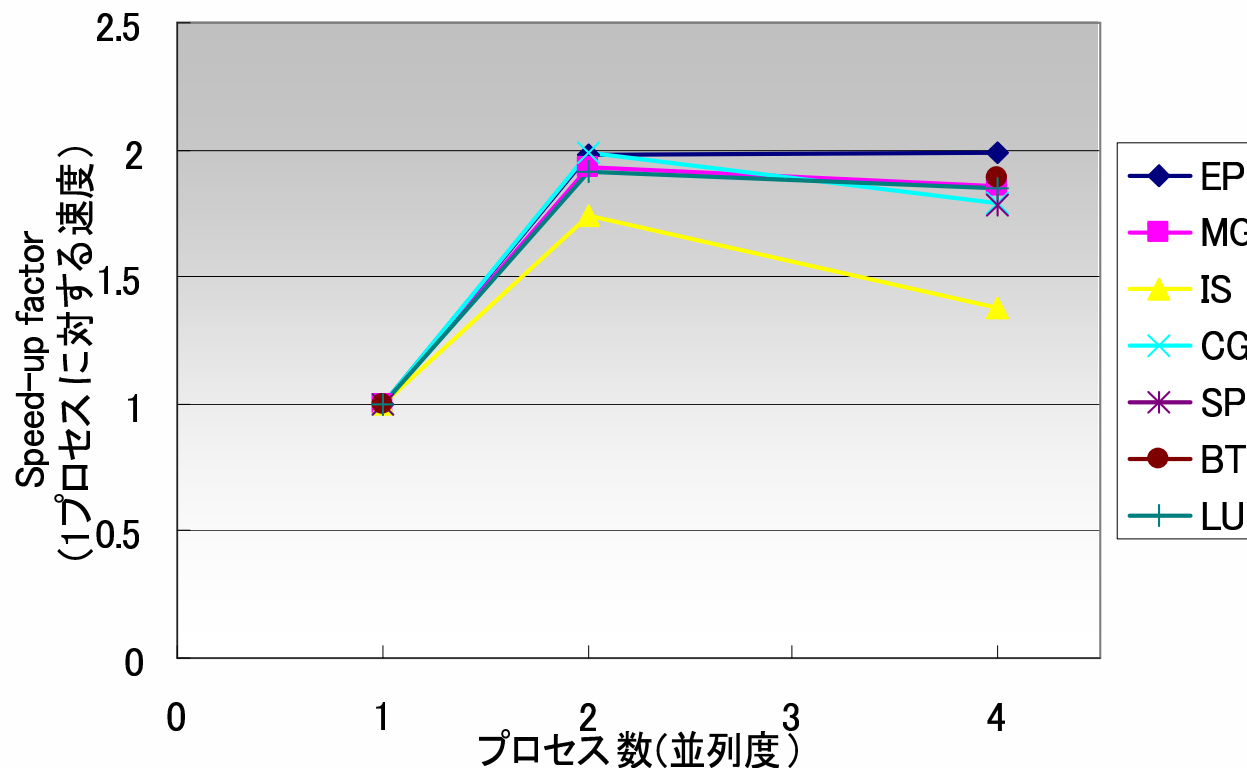
性能評価

- NPB: NAS Parallel Benchmark
 - 並列コンピュータの並列計算効率を測定
 - MPIを用いた並列処理のベンチマーク
 - 測定環境
 - CPU: M32700 (CPUコア数: 2)
 - freq: CPU 200MHz, BUS 100MHz
 - OS: Linux-2.4.19 SMP
 - MPI: MPICH 1.2.5 ch_shmem
 - Benchmark: NPB 2.4
 - 測定項目
 - EP, MG, CG, IS, LU, SP, BT
 - CLASS=S



性能評価

- NPB: NAS Parallel Benchmark



–2プロセッサで約2倍の速度

- 高いスケーラビリティが得られている

まとめ

- Linux/M32R: M32R用GNU/Linux環境
 - Linux/M32RをSMPに対応
 - 同期機構などのSMPに必要な機能を実装
 - M32Rシングルチップ マルチプロセッサで動作
 - 性能評価
 - Lmbench, NPBを用いて評価
 - SMP動作時でもカーネルの速度低下は少ない
 - 高いスケーラビリティが得られている





RENESAS

株式会社ルネサステクノロジ