HPCユーザーから見た様々なCPUのカラー

堀越　将司
大阪大学レーザー核融合研究センター
mhori@ile.osaka-u.ac.jp

上島　豊
特定非営利活動法人　けいはんな文化学術協会
大規模データマネジメント研究会

Special　thanks：**日本原子力研究所：**　Timur Esirkepov
けいはんな文化学術協会：　若林　大輔
大阪大学レーザー研：　城崎　知至、福田　優子、西原　功修

目次

1.ベンチマークからみたCPUのカラー

2.大規模システムの現状

3. Itanium 2ベースの最適なHPC環境構築

第1章　ベンチマーク

Itanium2をはじめとする様々なCPUで
種々のベンチマークを行い、
それぞれのCPUの特徴をみる

比較マシン(逐次実行)

- Intel Itanium2
- NEC SX-6 **(非占有環境)**
- HP Alpha Server ES40
- **富士通** PRIMEPOWER **(非占有環境)**
- **日立** SR8000 **(非占有環境)**
- Intel Pentium4

Intel Itanium2

CPU:Intel Itanium2 1.5GHz/L3 6MB/4way

Memory:16GB

1CPU**使用時メモリバンド幅**6.4GB/s **ノード全体**6.4GB/s

OS:Red Hat Enterprise Linux AS2.1(Kernel 2.4.18-e.31smp)

Compiler:Intel Fortran Compiler ver8.0

理論ピーク性能　6GFLOPS

NEC SX-6

CPU:8GFLOPS/4way

Memory:24GB

1CPU**使用時メモリバンド幅**　32GB/s **ノード全体**128GB/s

OS:SUPER-UX R13.1

Compiler:NEC Fortran Compiler for SX Rev.285

理論ピーク性能　8GFLOPS

HP Alpha Server ES40-4

CPU:Alpha21264B　833MHz/4way

Memory：16GB

1CPU**使用時メモリバンド幅**　2.6GB/s　(**ノード全体**5.2GB/s)

OS:Compaq Tru64 UNIX V5.1

Compiler:Compaq Fortran Compiler V5.5

理論ピーク性能　1.666GFLOPS

AlphaPowered

富士通 PRIMEPOWER

CPU: SPARC 64V 1.3GHz/128way

Memory:256GB

1CPU**使用時メモリバンド幅**　8.4GB/s (**ノード全体**133GB/s)

OS:SUN Sloaris9

Compiler:Fujitsu Fortran Compiler V5.4

理論ピーク性能　2.6GFLOPS

日立 SR8000　F1

CPU: Power3-Ⅱ 375MHz/8way

Memory:8GB

1CPU**使用時メモリバンド幅**　4GB/s (**ノード全体**32GB/s)

OS:HI-UX/MPP

Compiler:Hitachi Fortran Compiler　V01-05-/A

理論ピーク性能　1.5GFLOPS

Intel Pentium4

CPU: Intel Pentium4　3.0GHz　/L2 512KB/FSB800 /HT off

Memory:2GB

1CPU**使用時メモリバンド幅**　6.4GB/s　/DDR400/　875P

OS:Redhat Linux 9(Kernel 2.4.20-8)

Compiler:Intel Fortran Compiler　ver8.0

理論ピーク性能　単精度SSE；　12GFLOPS

倍精度SSE2；　6GFLOPS

理論FLOPS値について

FLOPS （Floating point number Operations Per Second）
コンピュータの処理速度をあらわす単位の一つ。
処理速度が1FLOPSのコンピュータは、1秒間に1回の浮動小数点演算ができることを示す。

Itanium２の場合の理論FLOPS値
1クロックで4個の単/倍精度浮動小数点メモリ操作と、
2個の単/倍精度浮動小数点加算演算と
2個の単/倍精度浮動小数点乗算演算が行える。

⇒Itanium2　1.5GHzでは1.5*4=6GFlops（単精度、倍精度で同じ）。

Pentium4のSSE、SSE2について

複数のデータに対し、一度にまとめて同じ命令を実行して処理することを
SIMD（Single Instruction Multiple Data）という。

Pentium4はSSE（Streaming SIMD Extensions）という拡張命令により
4個の単精度浮動小数点の加算演算と4個の単精度浮動小数点の乗算演算の
データを2クロックでパック化し、一度に処理して結果を得ることができる。
また、Pentium4はSSE2（Streaming SIMD Extensions2）という拡張命令により
2個の倍精度浮動小数点の加算演算と2個の倍精度浮動小数点の乗算演算の
データを2クロックでパック化し、一度に処理して結果を得ることができる。

よって、Pentium4の場合の理論FLOPS値

⇒Pentium4 3GHzは単精度計算では3*4=12Gflops、
倍精度計算では3*2=6Gflops

参考文献：Intel Technology Journal Q1,2001

ベンチマークプログラム

1.**姫野ベンチ　Mサイズ**（理研姫野氏開発、公開ベンチマークソフト）

2.**流体コード3種類**（阪大堀越開発）

⇒2-1　CIP**法**

⇒2-2　Roe**法**+MUSCL**補間**+TVD　（**以下**APR**と略す**）

⇒2-3　PPM**法**

3.**粒子コード**（PIC**法**）（**原研**Timur**氏開発**）

4.**フォッカープランク**(FP)**コード**（**阪大城崎氏開発**）

1.姫野ベンチについて

・3次元非圧縮性流体コードのコア部分
　(ヤコビ反復法、単精度)

計算サイズ：Mサイズ　257×129×129
　　(使用メモリ300MB程度)

1.姫野ベンチ　比較マシン結果

- NEC　SX-6

  -C hopt

  ⇒ 3219.815MLFOPS

- HP　Alpha　Sever　ES40-4　Alpha 21264B 833MHz

  -O5 -fast -tune ev7-inline all -speculate all -unroll2

  ⇒ 612.5643MFLOPS

- **富士通**　PRIMEPOWER　SPARC64V 1.3GHz

  -Kfast_GP2=2 -Klargepage=2 -Kprefetch_cache_level=3 -Kstriping=11

  ⇒ 906.7856MFLOPS

- **日立**　SR8000　F1

  -Os -noparallel

  ⇒ 322.8880MFLOPS

- Intel Pentium4　3GHz

  -O3 -fast -Vaxlib -xKWN -tpp6 -prof_use

  ⇒ 762.2132MFLOPS

PentiumIII用オプション、IPOが有効

1.姫野ベンチ　Itanium2結果　その1

log 1:12: MFLOPS: **328.0752** time(s): 60.17871 -Vaxlib
log 2:12: MFLOPS: **321.9560** time(s): 60.04492 -Vaxlib -tpp1
log3 :12: MFLOPS: 750.6443 time(s): 59.90918 -O3 -Vaxlib
log 4:12: MFLOPS: 928.3233 time(s): 59.51953 -O3 -Vaxlib -tpp1
log 5:12: MFLOPS: 806.1193 time(s): 59.69824 -O3 -Vaxlib -fast
log 6:12: MFLOPS: 1068.569 time(s): 59.40625 -O3 -Vaxlib -fast -tpp1
log 7:12: MFLOPS: 750.6321 time(s): 59.91016 -O3 -Vaxlib -ipo
log 8:12: MFLOPS: 928.4756 time(s): 59.50977 -O3 -Vaxlib -tpp1 -ipo
log 9:12: MFLOPS: 750.5953 time(s): 59.91309 -O3 -Vaxlib -static
log10:12: MFLOPS: 928.3690 time(s): 59.51660 -O3 -Vaxlib -tpp1 -static
log11:12: MFLOPS: 750.4730 time(s): 59.92285 -O3 -Vaxlib -fno-fnalias
log12:12: MFLOPS: 928.2928 time(s): 59.52148 -O3 -Vaxlib -tpp1 -fno-fnal
log13:12: MFLOPS: 805.9874 time(s): 59.70801 -O3 -Vaxlib -fast -fno-fnalia
log14:12: MFLOPS: 1068.608 time(s): 59.14746 -O3 -Vaxlib -fast -tpp1 -fno-fnalias
log15:12: MFLOPS: 750.6443 time(s): 59.90918 -O3 -Vaxlib -fno-fnalias -static
log16:12: MFLOPS: 928.2471 time(s): 59.52441 -O3 -Vaxlib -tpp1 -fno-fnalias -static
log17:12: MFLOPS: 75
log18:12: MFLOPS: 92
log19:12: MFLOPS: 75
log20:12: MFLOPS: 92
log21:12: MFLOPS: 80
log22:12: MFLOPS: 1
log23:12: MFLOPS: 75
log24:12: MFLOPS: 92
log25:12: MFLOPS: 75
log26:12: MFLOPS: 928.2776 time(s): 59.52246 -O3 -Vaxlib -tpp1 -fno-alias –ipo

デフォルト-O2から-O3への変更で2～3倍の性能向上！

プロシジャ最適(-fastに含まれる-ipo)も有効

-tpp1がかなり有効
(Itanimu2であっても Itanium用のオプション)

デフォルトコンパイルオプション
328.0752 MFLOPS
からコンパイルオプションのチューニング
-O3 -fast -Vaxlib -tpp1 -fno-fnalias –ivdep_parallel -ftz
によって
1069.202 MFLOPSまで向上 (3.33倍)

1.姫野ベンチ　Itanium2結果　その2

ディレクティブ挿入　プリフェッチ命令　**!DIR$PREFETCH　配列名　(固定形式)**
!DEC$PREFETCH　配列名　(固定＆自由形式)

log 6 :12:　　MFLOPS:　1068.569
log_preftc 6:12:　　MFLOPS:　1068.587

log14 :12:　　MFLOPS:　1068.608
log_preftc14:12:　　MFLOPS:　1068.644

log22 :12:　　MFLOPS:　1068.591
log_preftc22:12:　　MFLOPS:　1068.55

log_idep6:　　MFLOPS:　1068.64
log_idep_pref6:12:　MFLOPS:　1068.9

log_idep22:　　MFLOPS:　1068.815
log_idep_pref14:12:　MFLOPS:　1068.991

log_idep14:　　MFLOPS:　1069.008
log_idep_pref22:12:　MFLOPS:　1068.679

プリフェッチ命令を追加してもそれほど性能は上がらない
⇒姫野ベンチはDoループ内添え字が線形なため、Intelコンパイラがディレクティ部なしでもデフォルトでprefetchをしてくれているため

1.姫野ベンチ　Itanium2結果　その3

ディレクティブ挿入　ループアンローリング命令　!DIR$UNROLL (n)　（固定形式）

!DEC$UNROLL (n)　（固定＆自由形式）

```
!DEC$ PREFETCH a, p, b, c, wrk1, bnd
!DEC$ UNROLL (126)
      DO K=2,kmax-1
        DO J=2,jmax-1
!DEC$ UNROLL (12)
          DO I=2,imax-1
            S0=a(I,J,K,1)*p(I+1,J,K)+a(I,J,K,2)*p(I,J+1,K)
   1             +a(I,J,K,3)*p(I,J,K+1)
   2             +b(I,J,K,1)*(p(I+1,J+1,K)-p(I+1,J-1,K)
   3             -p(I-1,J+1,K)+p(I-1,J-1,K))
   4             +b(I,J,K,2)*(p(I,J+1,K+1)-p(I,J-1,K+1)
   5             -p(I,J+1,K-1)+p(I,J-1,K-1))
   6             +b(I,J,K,3)*(p(I+1,J,K+1)-p(I-1,J,K+1)
   7             -p(I+1,J,K-1)+p(I-1,J,K-1))
   8             +c(I,J,K,1)*p(I-1,J,K)+c(I,J,K,2)*p(I,J-1,K)
   9             +c(I,J,K,3)*p(I,J,K-1)+wrk1(I,J,K)
            SS=(S0*a(
            GOSA=GO
            wrk2(I,J,K
```

```
!DEC$ PREFETCH wrk2
!DEC$ UNROLL (12)
      DO K=2,kmax-1
        DO J=2,jmax-1
!DEC$ UNROLL (12)
          DO I=2,imax-1
            p(I,J,K)=wrk2(I,J,K)
          enddo
        enddo
      enddo
```

ディレクティブ挿入なし　　**ディレクティブ挿入あり**

1069.202 MFLOPS　⇒　1224.192 MFLOPS　（さらに1.15倍）

ループアンローリング数指定は有効

1.姫野ベンチ　Itanium2結果　その4

プロファイラの使用

手順1：–prof_genでコンパイル＆実行

手順2：–prof_useでコンパイル＆実行

ディレクティブなし

-O3 -fast -Vaxlib -tpp1 -fno-fnalias -ivdep_parallel -ftz

1069.202 MFLOPS

-O3 -fast -Vaxlib -tpp1 -fno-fnalias -ivdep_parallel -ftz *-prof_use*

2119.135MFLOPS

ディレクティブあり

-O3 -fast -Vaxlib -tpp1 -fno-fnalias -ivdep_parallel -ftz

1224.192 MFLOPS

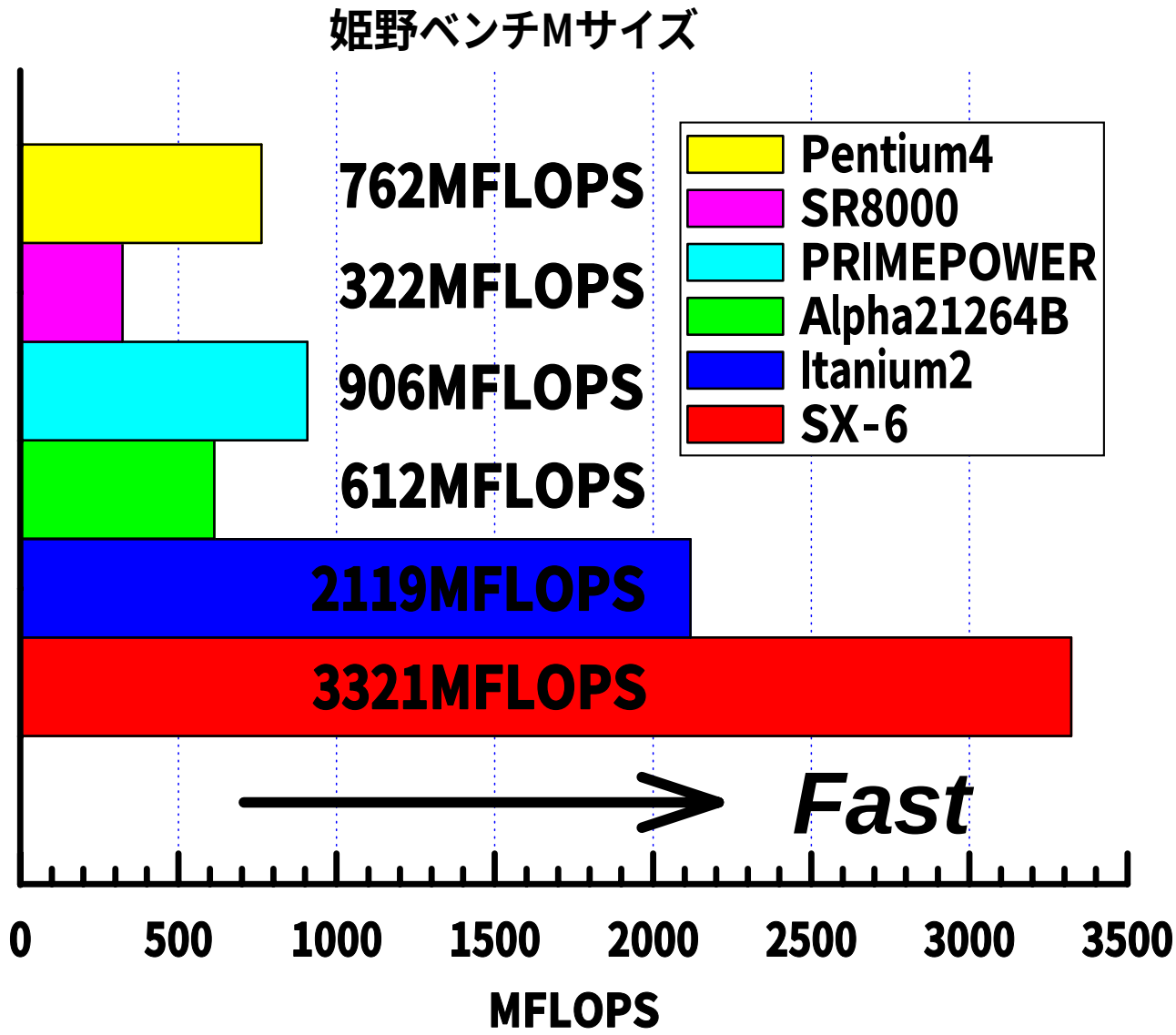
-O3 -fast -Vaxlib -tpp1 -fno-fnalias -ivdep_parallel -ftz *-prof_use*

2116.932MFLOPS

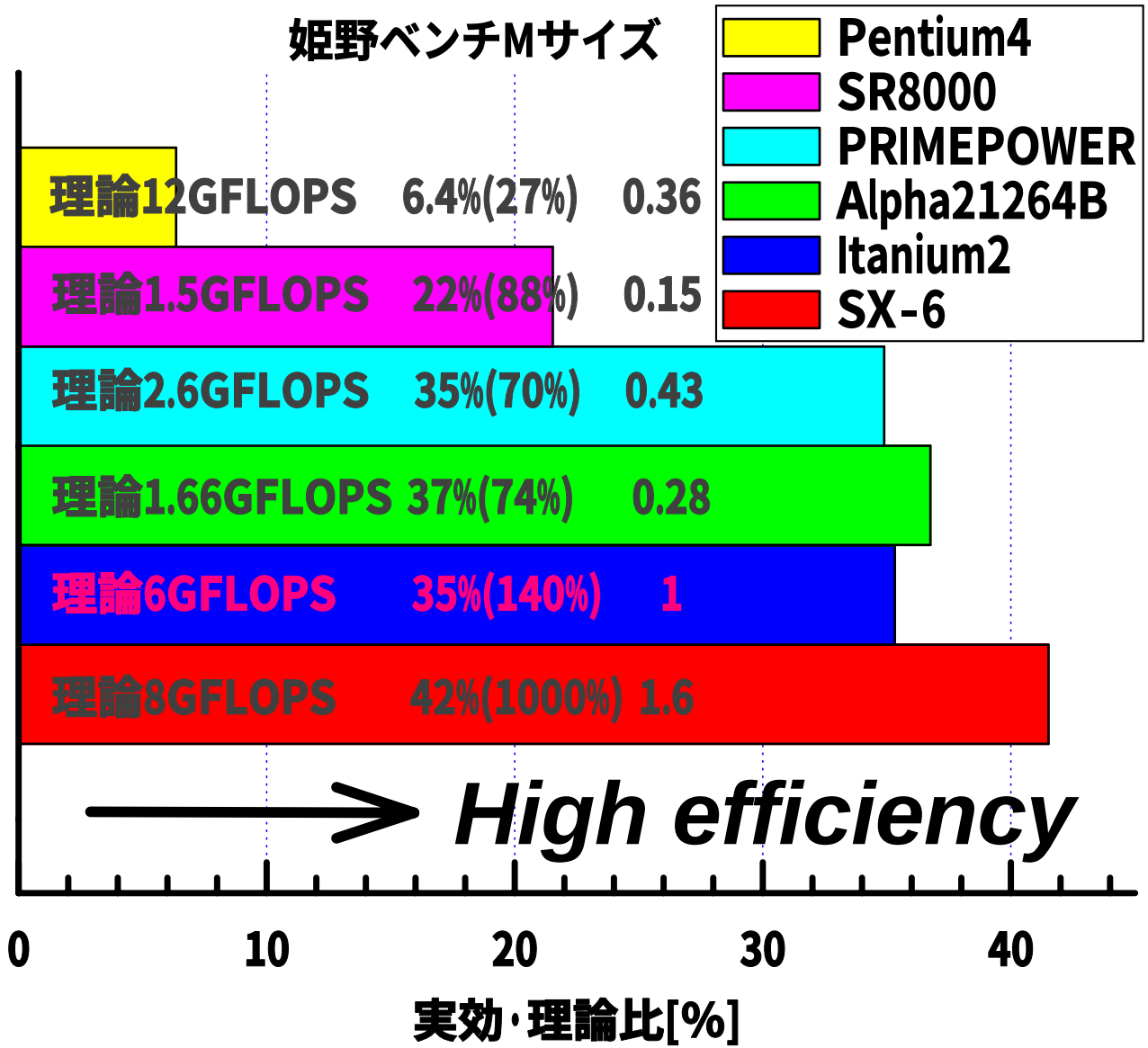
ただし、プロファイラでそれほど速度の向上が得られない場合もあるので注意。例えば、データがキャッシュが収まる場合は、スケジューリング命令等そのものがメモリ帯域を圧迫する

⇒例. 姫野ベンチSサイズ

姫野ベンチにおいて単一ＣＰＵでＧＦＬＯＰＳを出せるのは、Itanium2とSXのみである。



姫野ベンチにおいて演算効率の面では、PRIMEPOWER,Alpha,Itanium2が横一線である。



2.流体コードについて

・3次元圧縮性流体コード（陽解法、倍精度）

CIP法(空間3次精度、時間1次精度)

APR(空間2次精度、時間2次精度)

PPM法(空間3次精度、時間2次精度)

いずれもベクトル化率99%以上のベクトル機向けコード

計算サイズ：128×128×128（使用メモリ1GB前後）

初期条件：衝撃波管問題

計算ステップ：50

2-1 CIP法 比較マシン結果

- NEC SX-6

  -C hopt

  ⇒ 41.4791**秒** (**ベクトル化率** 99. 8%、MFLOPS 3237.72136)

- HP Alpha Sever ES40-4 833MHz

  -O5 -fast -tune ev7 -inline all -speculate all

  ⇒ 919.7336**秒**

- **富士通** PRIMEPOWER SPARC64V 1.3GHz

  -Kfast_GP2=2 -Klargepage=2 -Kprefetch_cache_level=3 -Kstriping=2

  ⇒ 423.8105**秒**

- **日立** SR8000 F1

  -Oss -noparallel

  ⇒ 1147.6452**秒**

- Intel Pentium4

  -O3 -Vaxlib -xKWN **-tpp6** -ipo -prof_use

  ⇒ 257.3315**秒**

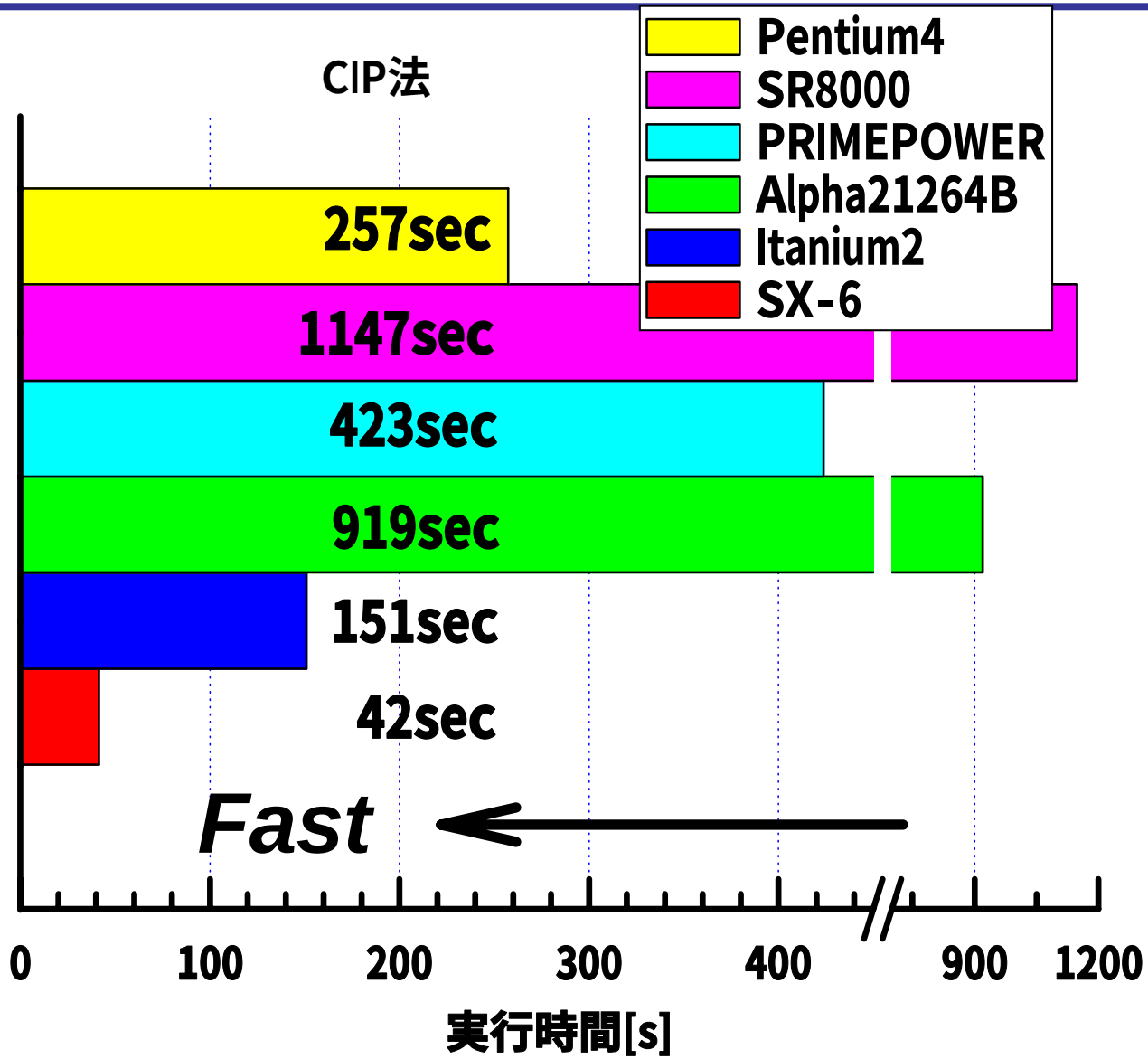
PentiumIII用オプション、IPOが有効

2-1 CIP法 Itanium2結果

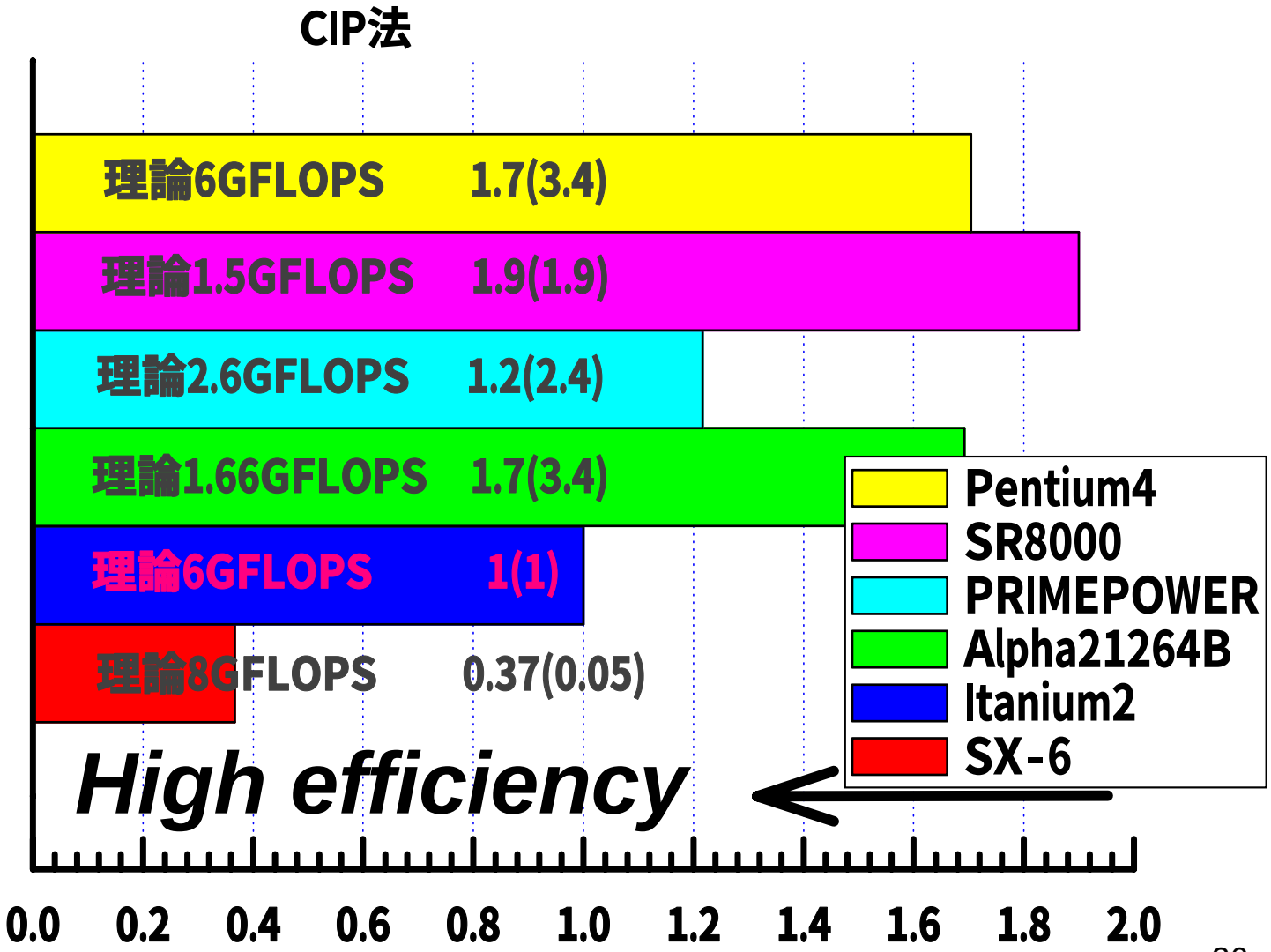
- ifort -Vaxlib (**デフォルト**、-O2 -tpp2)
  ⇒290.3583**秒**
- ifort -O3 -Vaxlib
  ⇒170.1200**秒**
- ifort -O3 -Vaxlib -ipo
  ⇒160.1990**秒**
- ifort -O3 -Vaxlib -ipo -fno-alias
  ⇒ 160.1482**秒**
- ifort -O3 -Vaxlib -ipo -fno-alias
  (**ディレクティブ** !DIR$ PREFETCH)
  ⇒ 160.1359**秒**
- ifort -O3 -tpp1 -Vaxlib -ipo -fno-alias -prof_use
  (**ディレクティブ** !DIR$ PREFETCH、**プロファイラ使用**)
  ⇒ 150.9816**秒**

Itanium用オプション、IPOが有効

CIPでは、Itanium2およびPentium4が格段の性能 (SX6の28-17%) をはじき出した。



Itanium2は、他のCPU（SXは除く）に比べると理論・実効性能は倍近く効率がよい。



2-2 APR法 比較マシン結果

- NEC SX-6

  -C hopt –pi auto

  ⇒ 244.4059**秒** (**ベクトル化率** 99.6%、3402MFLOPS)

- HP Alpha Sever ES40-4 833MHz

  -O5 -fast -tune ev7 -inline all -speculate all

  ⇒ 5607.1805**秒**

- **富士通** PRIMEPOWER SPARC64V 1.3GHz

  -Kfast_GP2=3 -Klargepage=2 -Kprefetch_cache_level=3 -Kstriping=2

  ⇒ 5046.3893**秒**

- **日立** SR8000 F1

  -Oss -noparallel

  ⇒ 15529.2028**秒**

- Intel Pentium4

  -O3 -Vaxlib -xKWN -tpp6 -fno-fnalias -ipo

  ⇒ 1164.8366**秒**

PentiumIII用オプション、IPOが有効

2-2 APR法 Itanium2結果

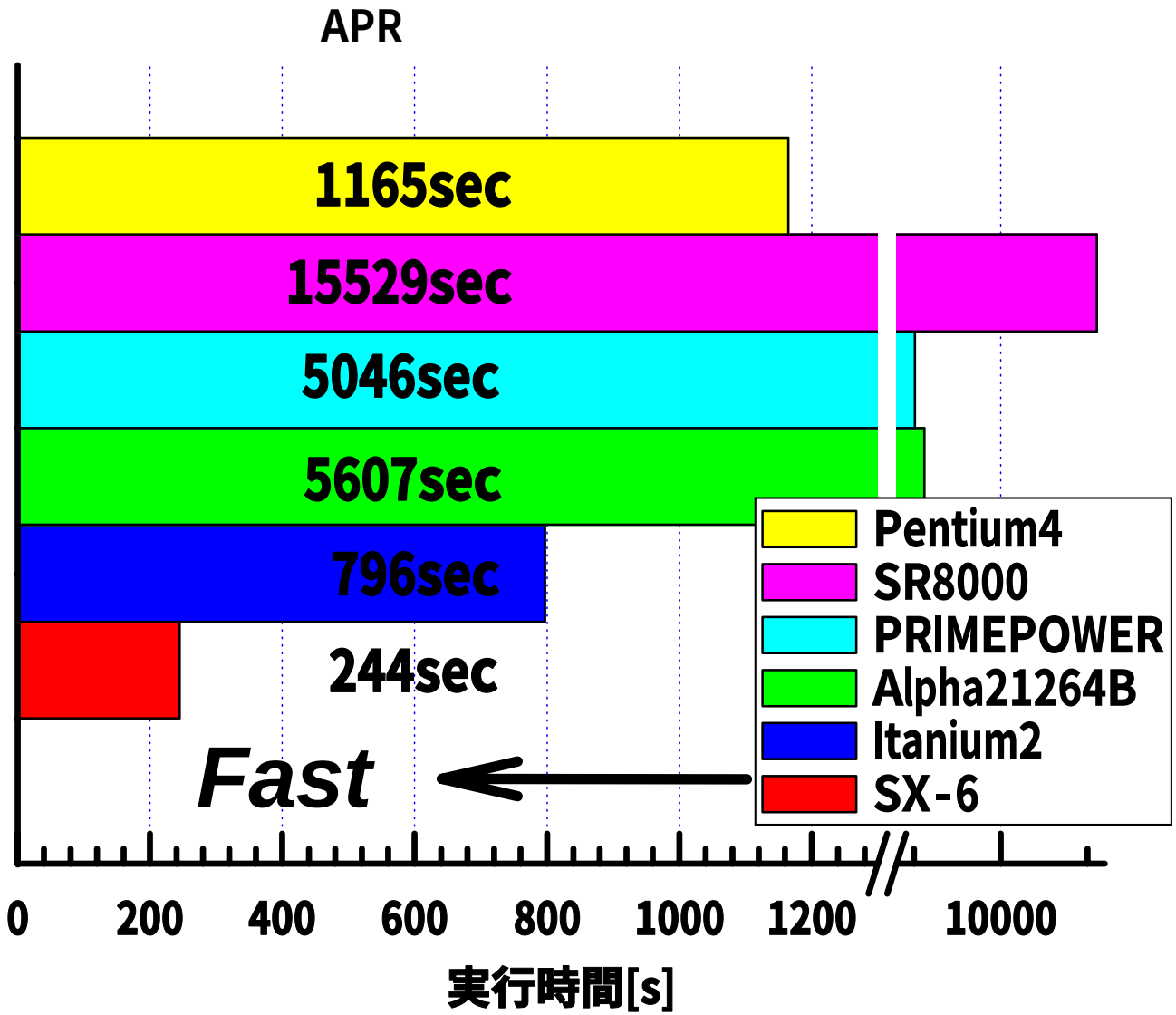
- ifort -Vaxlib (デフォルト、-O2 -tpp2)
  ⇒1334.5990秒

- ifort -O3 -Vaxlib
  ⇒875.1882秒

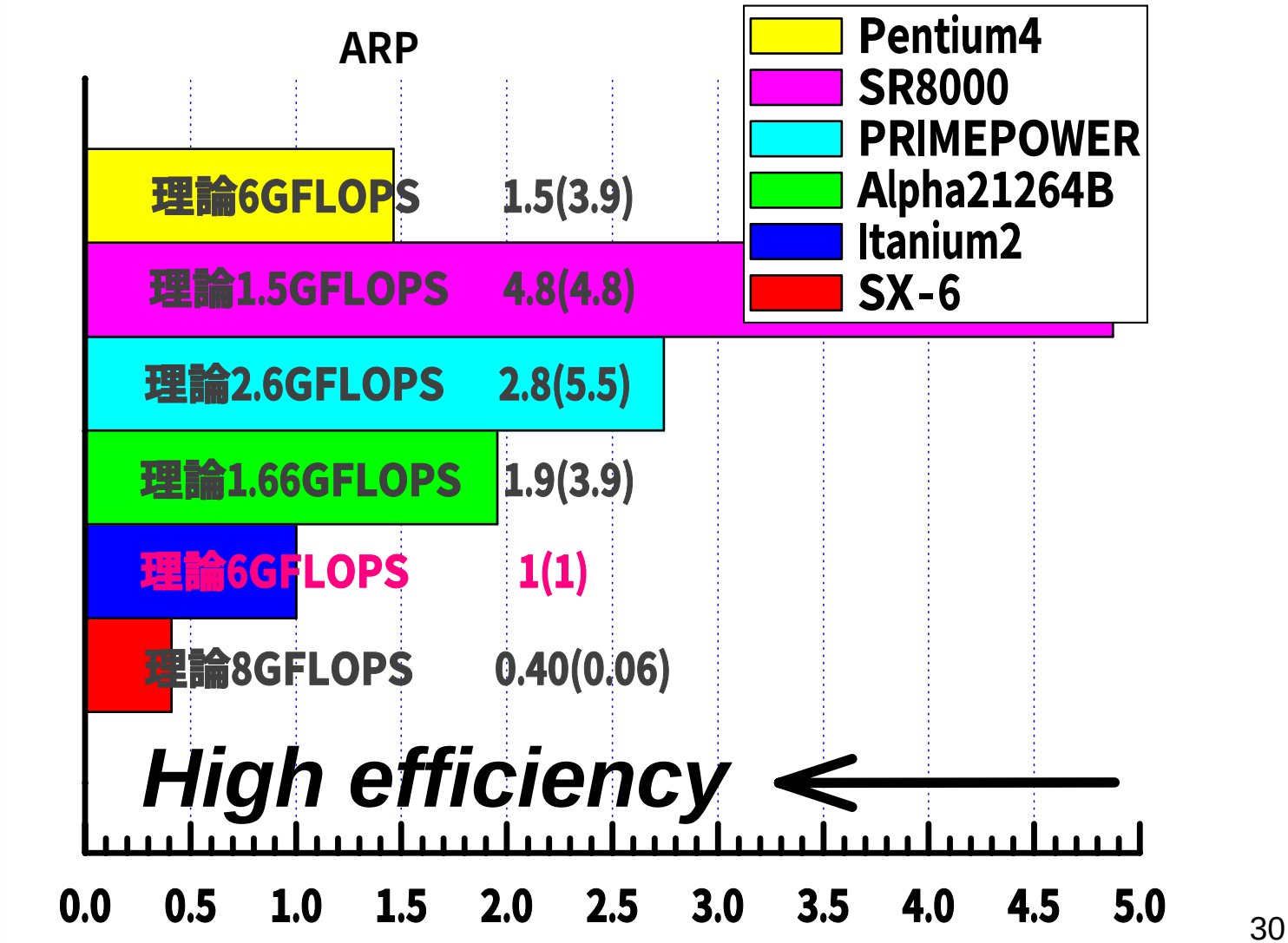
- ifort -O3 -Vaxlib -fno-fnalias -ipo
  ⇒ 797.3473秒

Itanium2用オプション、IPOが有効。
Loop unrollingディレクティブで更なる
向上の可能性あり

APRでは、SRの遅さ（Itanium2の20倍）が目立った。



PRIMEPOWERとSRが、APRを苦手としているのに対し、Alphaは、CIPと同性能を出している。



2-3　PPM法　比較マシン結果

- NEC　SX-6

  -C hopt　–pi auto

  ⇒　609.3589**秒**（**ベクトル化率** 99.4%、1991MFLOPS）

- HP　Alpha　Sever　ES40-4　833MHz

  -O5 -fast -tune ev7 –arch ev7

  ⇒ 25465.1878**秒**

- **富士通**　PRIMEPOWER　SPARC64V 1.3GHz

  -Kfast_GP2=3 -Klargepage=1 -Kprefetch_cache_level=3 -Kstriping=2

  ⇒ 53610.3754**秒**

- **日立**　SR8000　F1

  -Oss -noparallel -pvfunc=3 -predicate –noscope

  ⇒ 92497.9832**秒**

- Intel Pentium4

  -O3 -Vaxlib -xKWN -tpp6 -ipo

  ⇒ 2696.7019**秒**

PentiumIII用オプション、IPOが有効

2-3　PPM法　Itanium2結果

- ifort -Vaxlib　(**デフォルト**、-O2 -tpp2)
  ⇒5894.5673**秒**

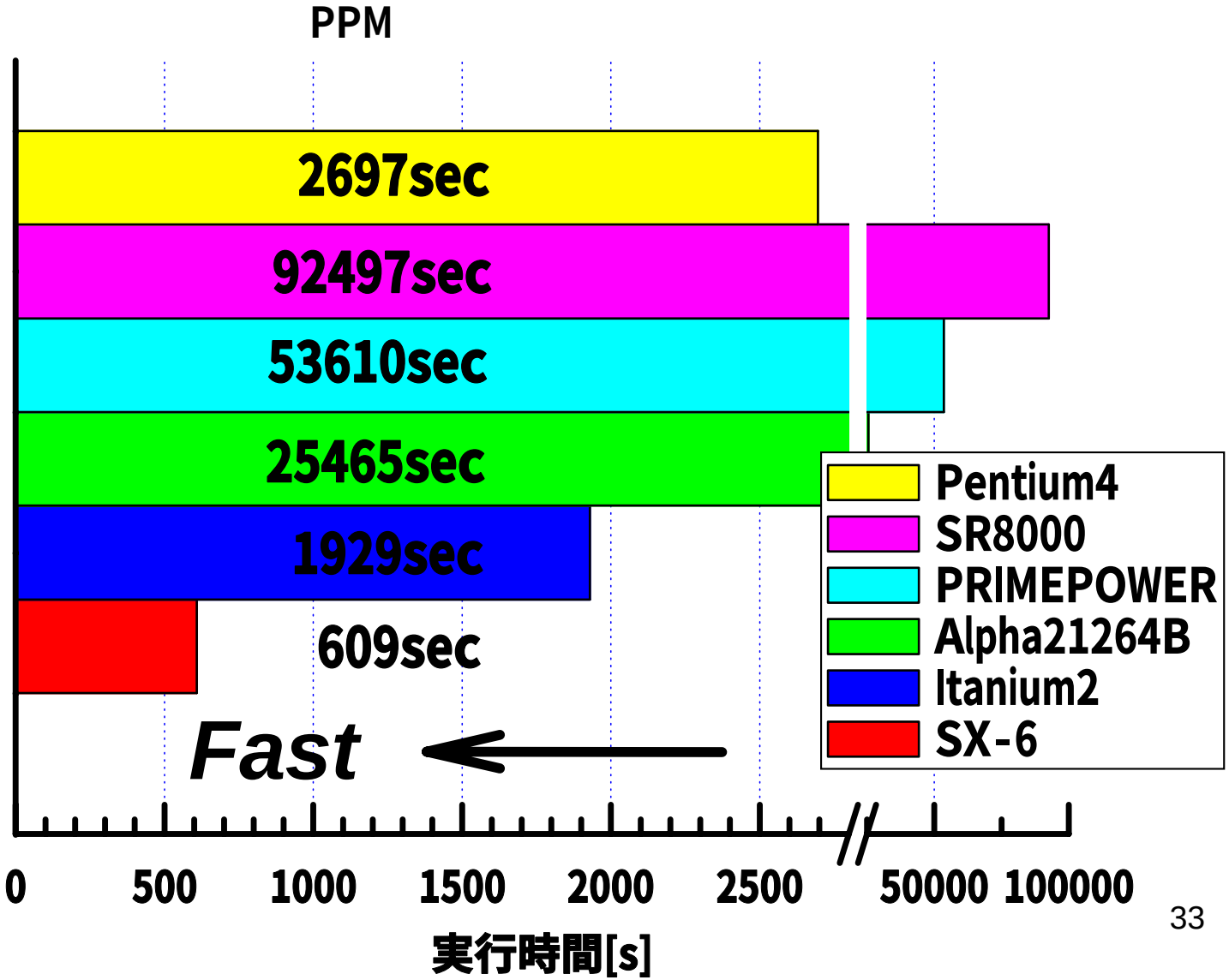
- ifort -O3 -Vaxlib
  ⇒3324.6899**秒**

- ifort -O3 -Vaxlib -ipo
  ⇒1931.5548**秒**

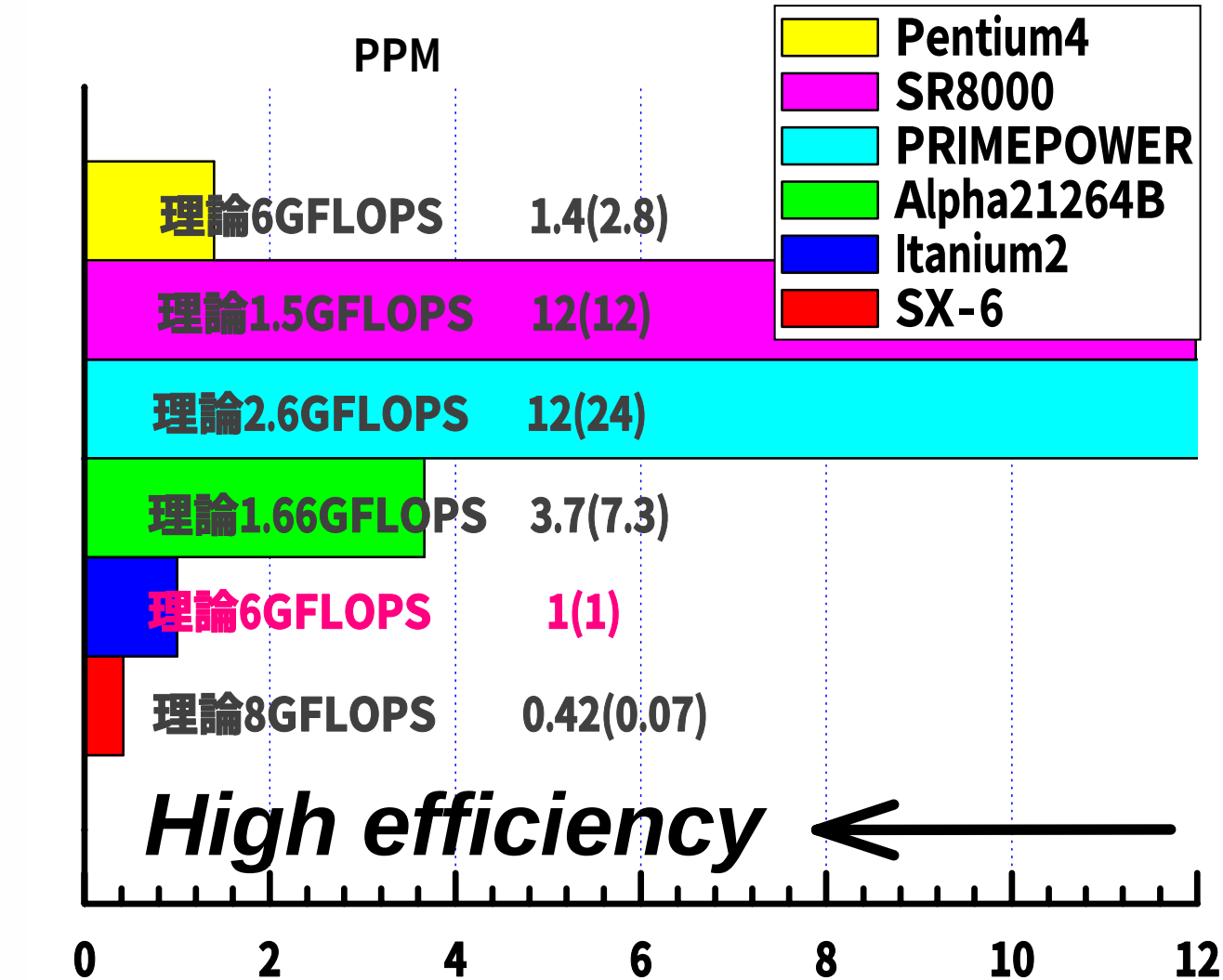
- ifort -O3 -Vaxlib -fno-alias
  ⇒ 1929.5491**秒**

Itanium2用オプションが有効。
Loop unrollingディレクティブで更なる向上の可能性あり

PPMでは、スパコン系の13-45倍、SXの30%の性能をItanium2がマークした。



PPMでは、SR, PRIMEPOWERの非効率性が目立った。



理論ＦＬＯＰＳ（クロック周波数）を仮想的に固定した場合の実行時間比

Pentium4をもとに考えたCPUのカラー　～流体編～

Itanium2：VLIW、EPIC（命令並列、コンパイラ任せ）が有効に働いている。
max,min,absといった数学関数が少し弱い。
メモリのアクセスは高速。（専用のロード/ストア　パイプ）
SXに対する速度ファクターは3.2~3.6程度。
種々のコードで安定した実効性能がある。
理論FLOPSベースでもクロックベースでも非常に速い。
Alpha：多重同時命令発行は有効でない。数学関数はItanium2並。
ONキャッシュが小さめなので複雑なループが弱い
PRIMEPOWER：多重同時命令発行は有効でない。メモリ周りはコピーも遅い。
数学関数も遅い。ONキャッシュが大きいので
複雑なループはAlphaより多少速い。
SR8000：多重同時命令発行は有効でない。AlphaやPRIMEPOWERより遅い。
ONキャッシュが大きいので速いはずだが。。除算が遅い。
コンパイラが除算の乗算化をしない、レジスタ数に合わせた最適化を
しない。メモリのランダムアクセスは優秀。数学関数は速い。
SX：メモリアクセスが非常に高速なため1CPUでは他の追随を許さない。

粒子コードについて

・2次元PICコード（倍精度）

ベクトル化率95%以上のベクトル機向けコード

計算サイズ：4000×4000（使用メモリ1.5GB程度）

初期条件：真空領域に挟まれたプラズマにレーザーを入射、粒子数2,000,000

計算ステップ：100

3　PIC法　比較マシン結果

- NEC　SX-6
  -C hopt　-Wf"-pvctl listvec loopcnt=100000000"
  ⇒　43.5613**秒**（**ベクトル化率** 95. 8%、MFLOPS 2047.605）
- HP　Alpha　Sever　ES40-4　Alpha 21264B 833MHz
  -fast -tune ev7
  ⇒ 195.2215**秒**
- **富士通**　PRIMEPOWER　SPARC64V 1.3GHz
  -Kfast_GP2=2 -Klargepage=2 -Kprefetch_cache_level=3 -Kstriping=2
  ⇒　244.6536**秒**
- **日立**　SR8000　F1
  -Oss -noparallel
  ⇒　116.3513**秒**
- Intel Pentium4　3GHz
  -O3 -fast -Vaxlib -xKWN
  ⇒　75.09854**秒**

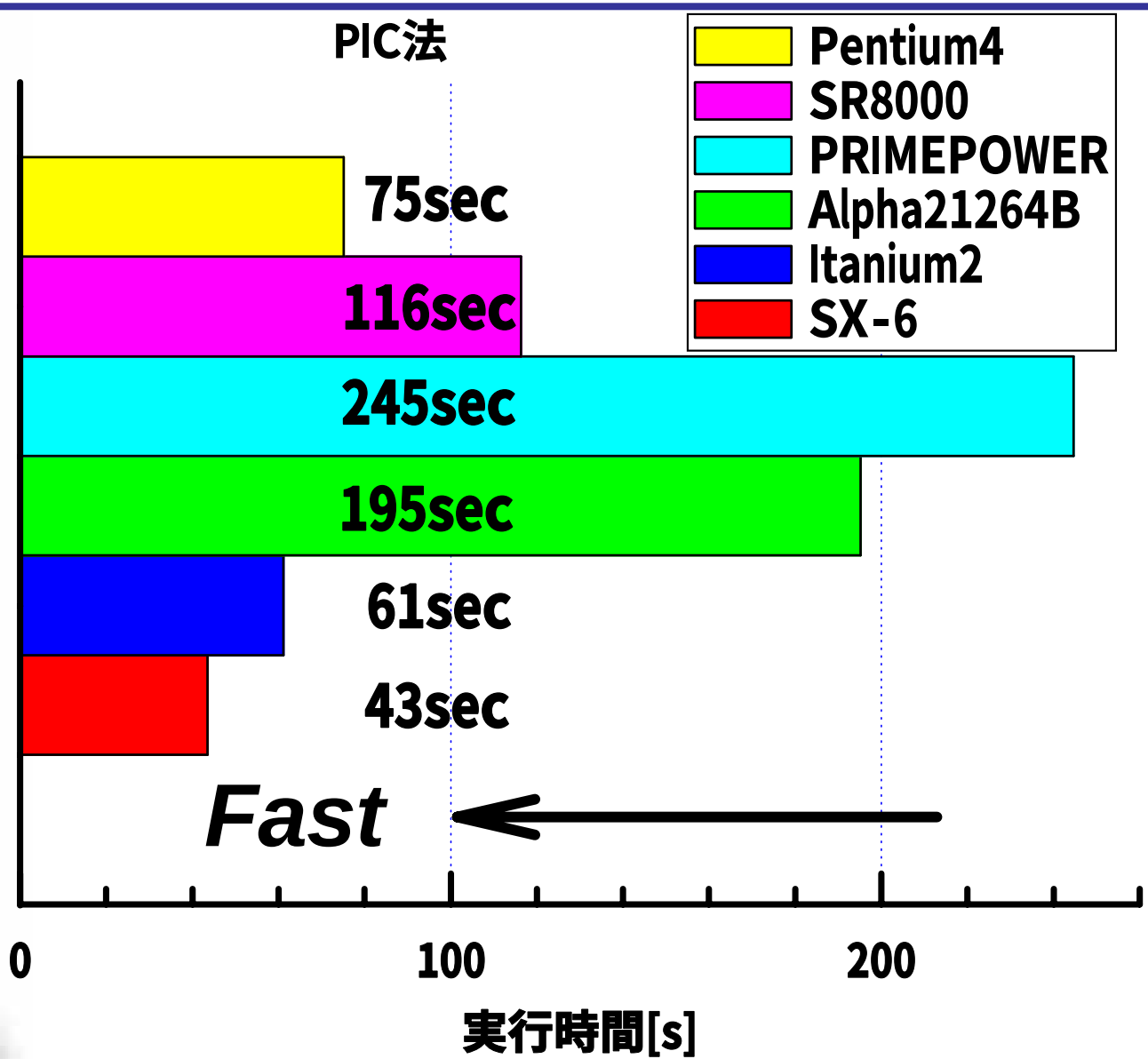
Pentium4用オプション、IPOが有効

3　PIC法　Itanium2結果

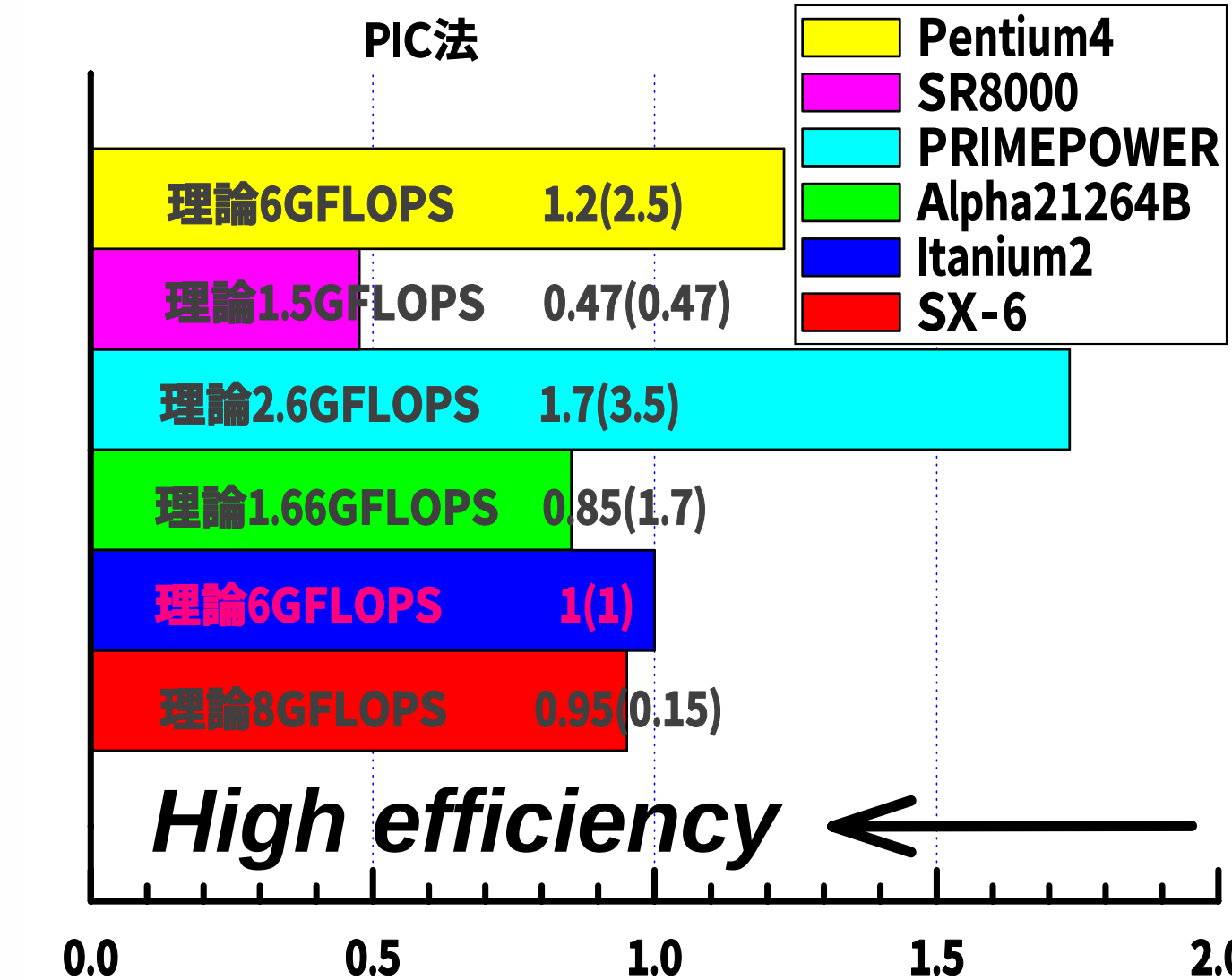
- ifort -Vaxlib　(**デフォルト**、-O2 -tpp2)
  ⇒257.2682**秒**
- ifort -O3 -Vaxlib
  ⇒63.2322**秒**
- ifort -O3 -Vaxlib -ipo -fno-alias
  ⇒63.1727**秒**
- ifort -O3 -Vaxlib -fast -fno-alias -prof_use
  (**プロファイラ使用**)
  ⇒ 61.0910**秒**
- ifort -O3 -Vaxlib -fast -fno-fnalias -prof_use
  (**プロファイラ使用**)
  ⇒ 61.0023**秒**

Itanium2用オプション、IPOが有効

ランダムメモリアクセスが多い、PICでは、Itanium2がSXに肉薄した性能をマークした。



SR, Alphaが得意でＳＸが不得意であるが、Itanium2は堅調な性能を出している。



Pentium4をもとに考えたCPUのカラー　～粒子編～

Itanium2：理論FLOPSとクロックベースでは同時4命令発行(SR8000)に負ける。
実行時間ではPentium4に肉薄されている。
SXに対する速度ファクターは1.4程度。

Alpha：多重同時命令発行は有効。
理論FLOPSに対して比較的効率がよい。

PRIMEPOWER：多重同時命令発行が有効であるコードだが、
ランダムメモリアクセスが遅いため性能を生かしきれない。

SR8000：同時4命令発行が非常に有効。メモリのランダムアクセスも優秀。
理論FLOPSに対して最も効率がよい。

SX：理論FLOPS ベースの効率では、Itanium2と同程度

フォッカープランクコードについて

- 1次元フォッカープランク(FP)コード（倍精度）
  高速電子FP＋バルクプラズマ流体＋輻射を計算
  ベクトル化率98%以上のベクトル機向けコード

計算サイズ：1000（使用メモリ400MB程度）

初期条件：高密度プラズマに相対論的高速電子を入射

計算ステップ：12タイムステップ＋各ステップでの収束計算

4　FPコード　比較マシン結果

- NEC　SX-6

  -C hopt　-Wf"-pvctl listvec loopcnt=100000000"

  ⇒ 17.2988**秒**（**ベクトル化率** 98. 7%、2911MFLOPS）

- HP　Alpha　Sever　ES40-4　Alpha 21264B 833MHz

  -O5 -fast -tune ev7 -inline all -speculate all

  ⇒ 224.1667**秒**

- **富士通**　PRIMEPOWER　SPARC64V 1.3GHz

  -Kfast_GP2=2 -Klargepage=2 -Kprefetch_cache_level=3 -Kstriping=4

  ⇒ 387.1140**秒**

- **日立**　SR8000　F1

  -Os -noparallel

  ⇒ 218.5404**秒**

- Intel Pentium4　3GHz

  -O3 -fast -Vaxlib -xKWN -fno-alias

  ⇒　83.6965**秒**

Pentium4用オプション、IPOが有効

4　FPコード　Itanium2結果

- ifort -Vaxlib -noautomatic　(デフォルト、-O2 -tpp2)
  ⇒73.05956秒

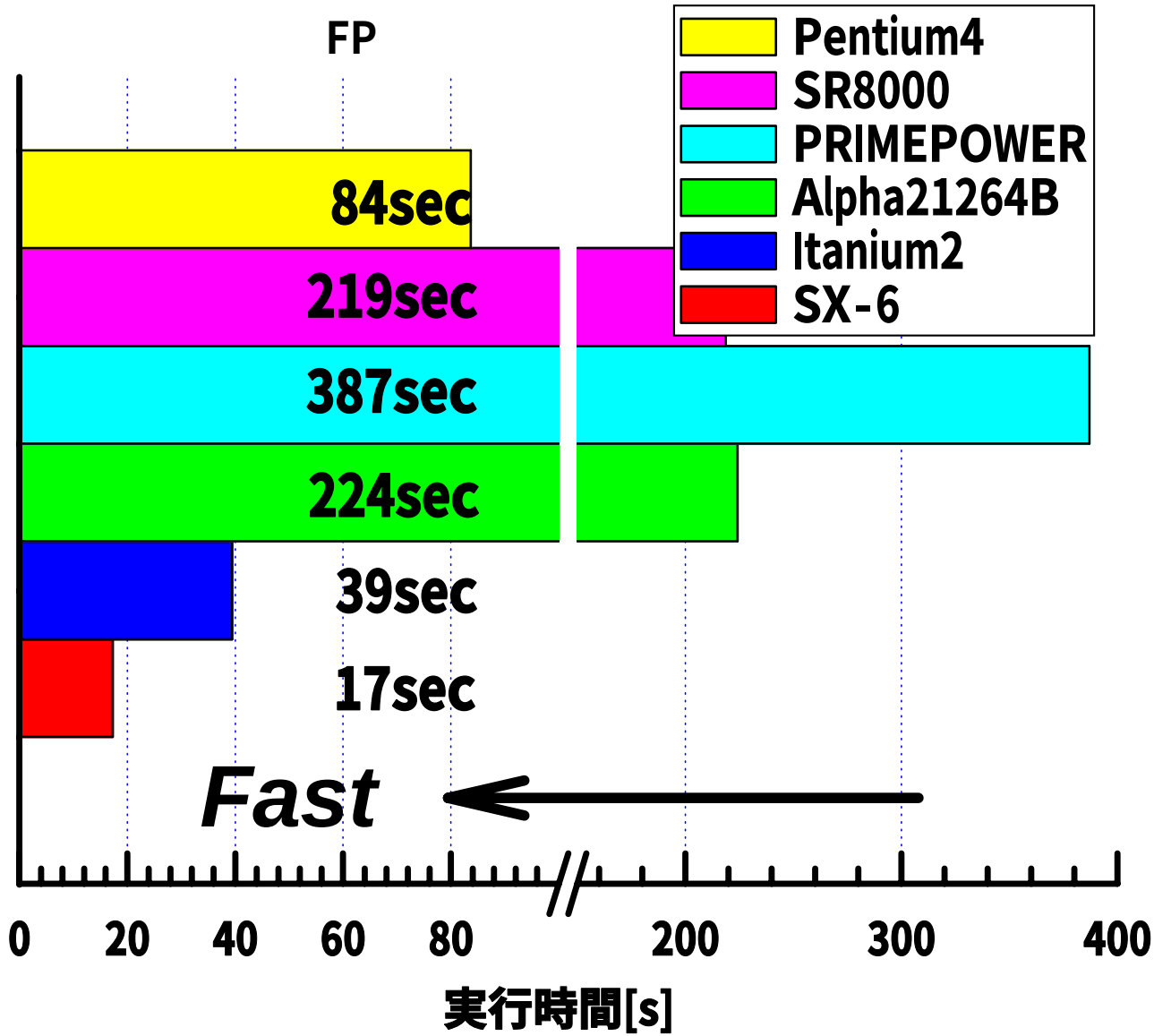
- ifort -Vaxlib -O3 -noautomatic
  ⇒40.5828秒

- ifort -O3 -fast -Vaxlib -noautomatic
  ⇒39.9900秒

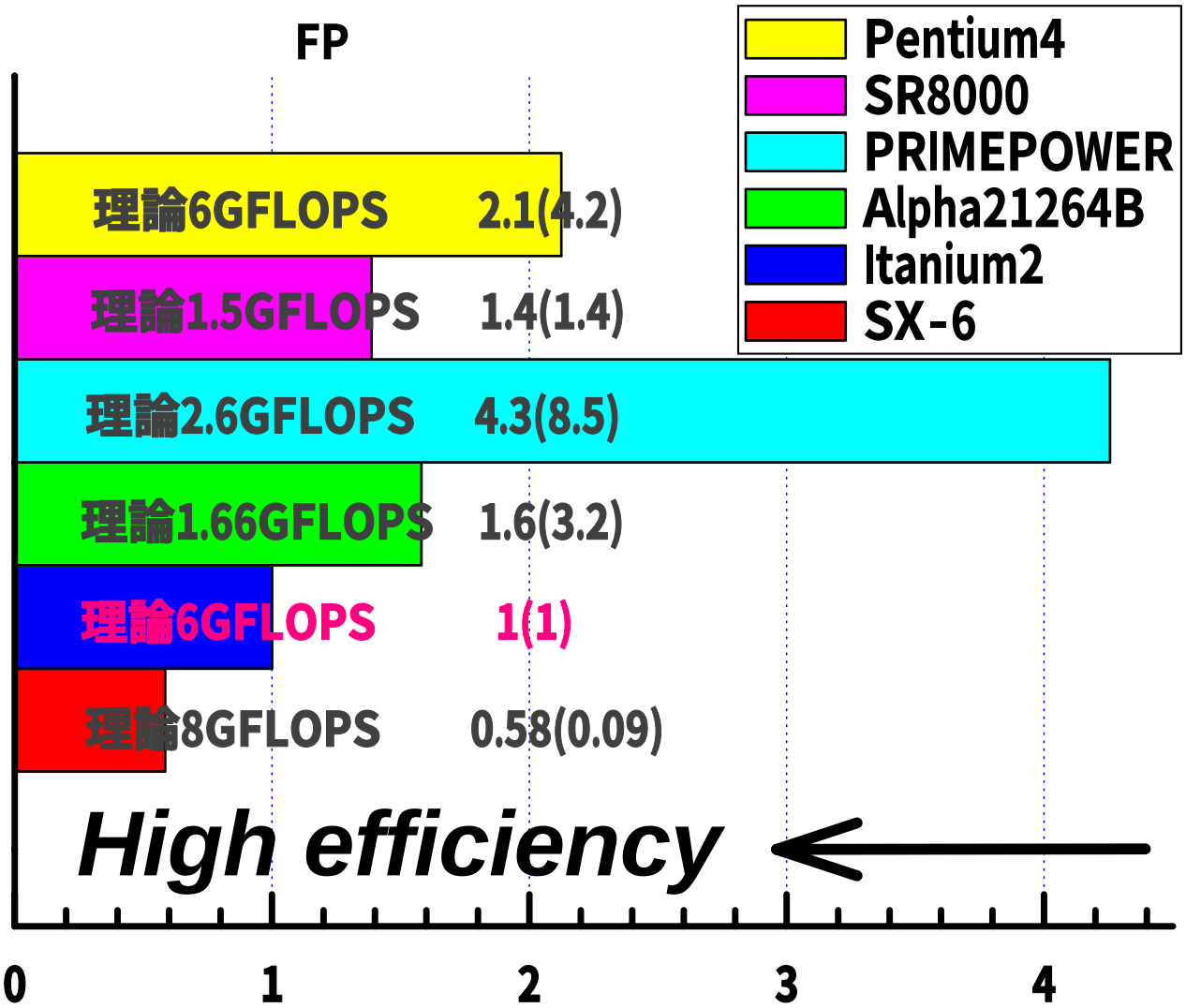
- ifort -O3 -fast -Vaxlib -noautomatic -prof_use
  ⇒38.7282 秒

Itanium2用オプション、IPOが有効

FPは、CIPと同じような傾向を持つが PRIMEPOWERが極端に遅くなっている。



FPは、PRIMEPOWER以外であれば、CPU依存性の少ないコードだといえる。



Pentium4をもとに考えたCPUのカラー　～FP編～

Itanium2：VLIW、EPIC（命令並列、コンパイラ任せ）が有効に働いている。
SXに対する速度ファクターは2.3程度。
理論FLOPSベースでもクロックベースでも非常に速い。

Alpha：多重同時命令発行はまあまあ。ONキャッシュが小さい。

PRIMEPOWER：メモリアクセスの遅さが全体性能を削ぎ落としている。

SR8000：多重同時命令発行が有効。理論FLOPSベースやクロックベースで
Itanium2により若干遅い程度。ONキャッシュの大きさも有効に
働いている。

SX：メモリアクセスが非常に高速なため速い。

各ＣＰＵのカラーと計算機選定のベストマッチ

・Pentium4~Itanium2~Alphaは、同じような傾向を示し、システムで同じFLOPS値をそろえた場合の倍精度浮動小数点演算が主体の実効性能比は、

Pentium4:Itanium2:Alpha、0.63:1.00:0.51

同様にクロック周波数をそろえて比較をすると

Pentium4:Itanium2:Alpha、0.32:1.00:0.26

となり、Itanium2の圧倒的性能とクロックアップを含めた将来性が期待できる。

・SR8000は、PIC（天体、プラズマ）を中心とするランダムアクセス系の演算は非常に高速である。ただし、流体などは苦手としている。
しかし、ループの最適化、除算の乗算化など自動で行われていない為の結果なので、チューニングのやりがいがあるCPUともいえる。

・PRIMEPOWERは、姫野ベンチマーク、CIPはItanium2に匹敵するほど高速であり、流体系の計算に強く思えるところもあるが、APR,PPMのように不得意も多い。
128waySMPという巨大共有メモリを使えるという点は他にないメリットである。

第2章　大規模システムの現状

意外と知られていない？

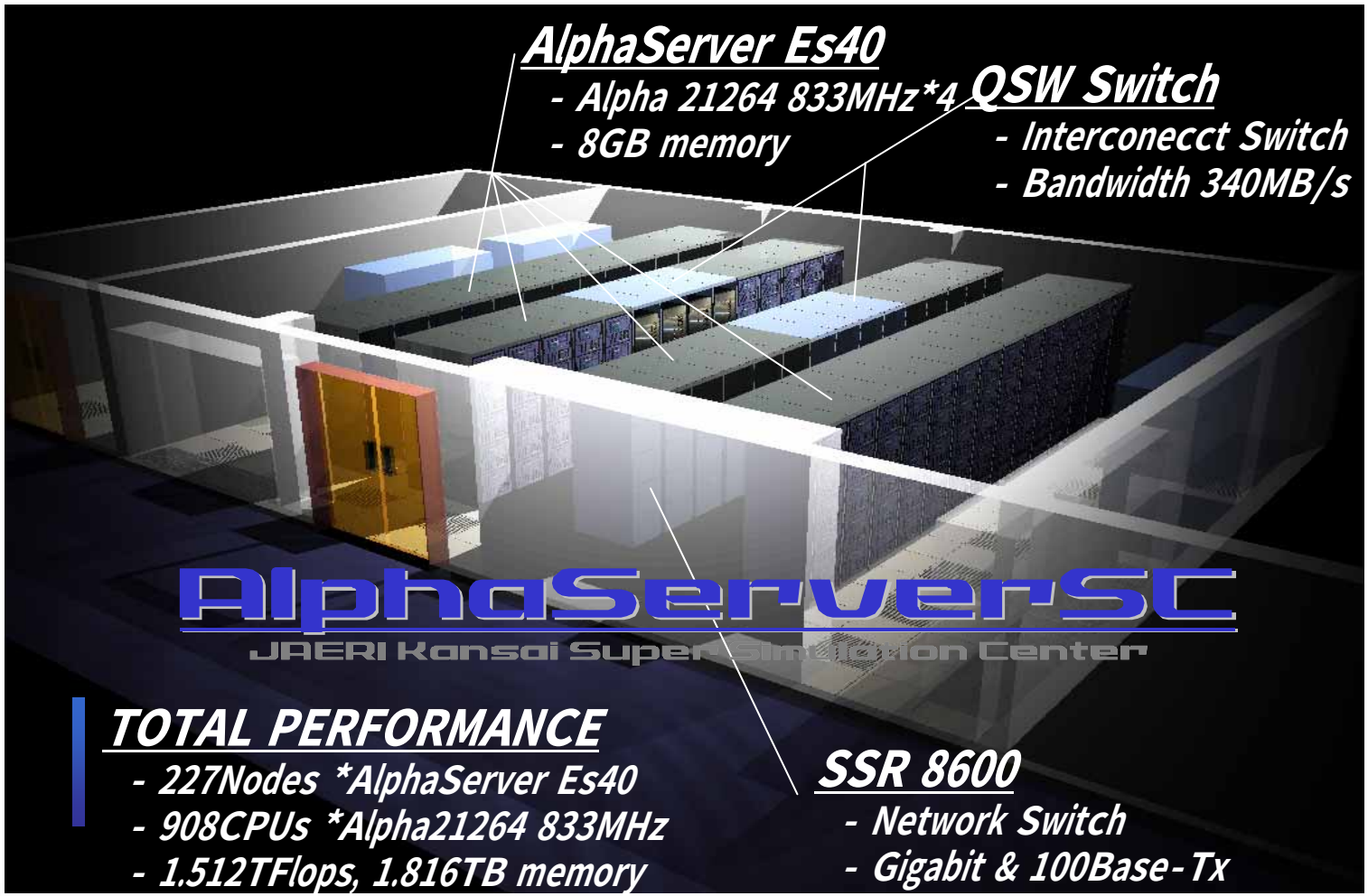
全CPUをすべて使用するとネットワーク周りでパフォーマンスの大幅な低下を起こす

⇒解決策はあるのか？

全CPUをすべて使用するとネットワーク周りでパフォーマンスの大幅な低下を起こす（大規模システムで顕著）

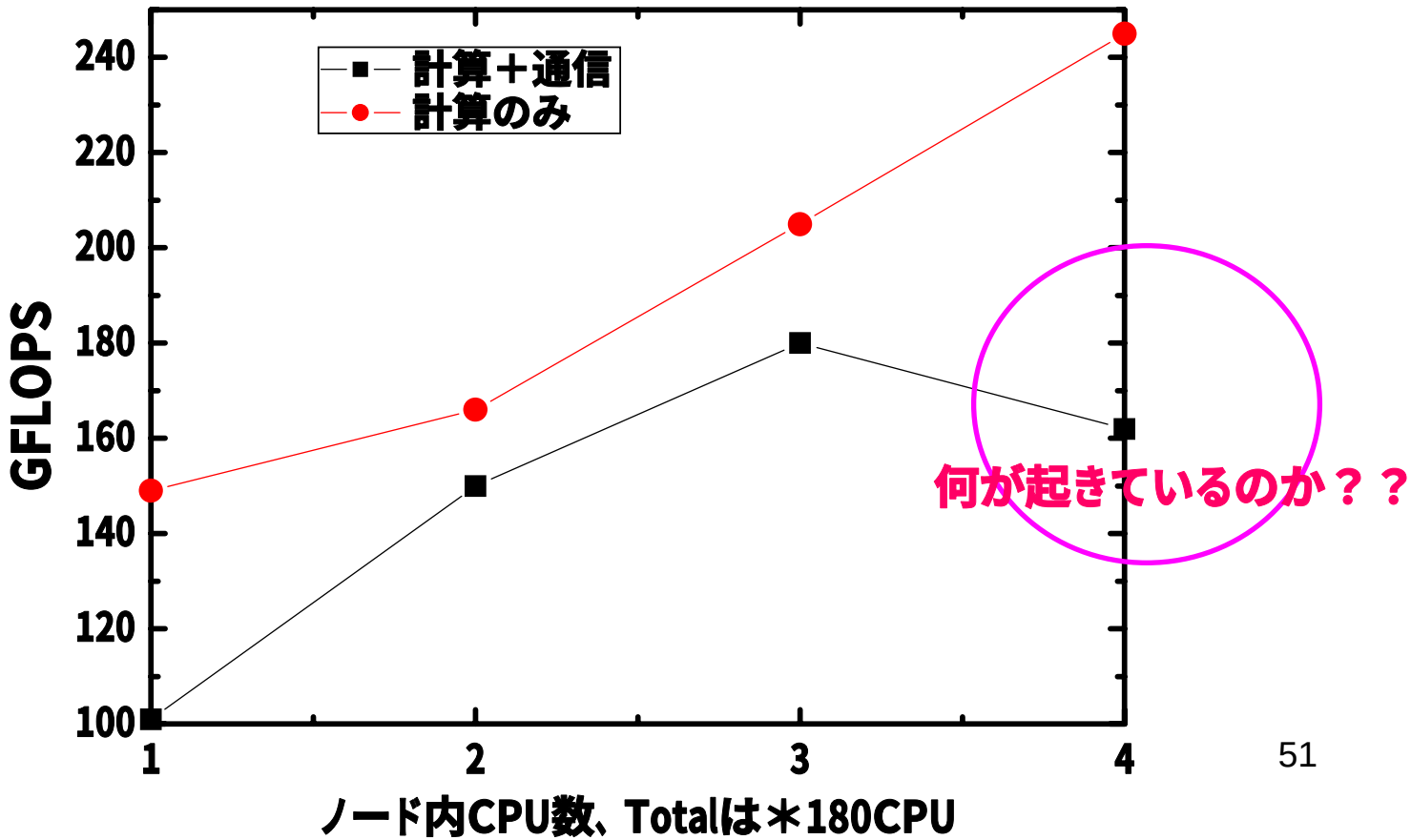
事例　姫野ベンチXLサイズ

日本原子力研究所　Alpha　Sever　180ノード720CPU　にて



姫野ベンチXLサイズにおけるパフォーマンスの大幅な低下

	スコア	実効・理論性能比	通信しない場合
180node*4CPU:	162 GFLOPS	13.5%	245 GFLOPS 20.4%
180node*3CPU:	180 GFLOPS	20.0%	205 GFLOPS 22.7%
180node*2CPU:	150 GFLOPS	25.0%	166 GFLOPS 27.6%
180node*1CPU:	101 GFLOPS	33.7%	149 GFLOPS 50.0%



全ノード全CPU使用時のパフォーマンスの大幅な低下の原因

Alphaサーバシステムは
超高速インターコネクト（bandwidth340MB/s、Latency5μs）を使用しているため通信時のバンド幅、レイテンシは問題ではなかった。

⇒では何か？大規模システムの宿命？

1.通信時には、ドライバ読み込みなどの多数のプロセスが立ち上がる

2.Alphaサーバシステムは32ノードで1パーティションであり、
パーティションごとの管理等をするミドルウエアがパーティションの
先頭と最後尾にある

3. 加えてシステムカーネルが存在する

これらが複雑に絡み合い、通信時の性能を削ぎ落としている

全ノード全CPU使用時のパフォーマンスの大幅な低下の解決策

解決策1．　全ノードにおいて、1CPUを遊ばせる
⇒しかしながら、遊ばせるCPUが多すぎる
　(180ノード時で180CPU)

通信時に最も負荷のかかるCPUはどれか？
⇒全CPU使用時における各パーティションの先頭と最後尾のノードのCPU
　　(カーネル＋ミドルウエア＋通信)

解決策2．　各パーティションの先頭と最後尾のノードを
　　プログラム実行に使用しない
⇒遊ばせるCPUはパーティション毎で済む
　(32ノード毎のパーティションで180/32　*2=12ノード程度
　12ノード*4CPU=48CPU)

第3章　Itanium 2ベースの最適なHPC環境構築

ポイント

大規模計算においては、ネットワークがボトルネックになることが多い
⇒多数の小さいパケット（レイテンシが重要）
多数の大きなパケット（バンド幅が重要）
ネットワーク選定の重要性を考える

ネットワークのコストは非常に高価
⇒コストにしめるCPUの割合は小さい
⇒割高なItanium2でもシステム全体でのコストパフォーマンスは
Pentium4、Xeonを**上回る？**

ネットワークにおけるバンド幅とレイテンシ

価格はおおよそのもの

	Bandwidth	MPI Latency
Quadrics QsNetⅡ cost~??	~900MB/s	～4.5 μs
Infiniband cost ~$2000/port	~800MB/s	～7μs
Quadrics QsNet cost ~$3500/port	~350MB/s	～5.5 μs
Myrinet 2000 cost ~$1700/port	~ 240MB/s	～7μs
Myrinet cost ~$1500/port	~150MB/s	～15μs
Gigabit Ethernet cost ~$1000/port	~100MB/s	～85μs

1μs＝CPU

大規模システムでは
CPUのコストは大きくない

現在のネットワーク/インターコネクトのトレンド

小規模システムGigabit Ethernet
　Pentium４　～16CPU
中規模システムMyrinet、Myrinet 2000
　Pentium４、Xeon　～64?CPU
大規模システムMyrinet 2000、Infiniband
　Pentium４、Xeon　～256?CPU
超大規模システムInfiniband、QsNet、特注仕様
　Xeon、Alpha　　～数千?CPU

TOP500の3位～6位にみるシステム性能

3	Virginia Tech United States/2003	X 1100 Dual 2.0 GHz Apple G5/Mellanox Infiniband 4X/Cisco GigE / 2200 Self-made	10280 17600	58% 4.7GFLOPS/CPU
4	NCSA United States/2003	Tungsten PowerEdge 1750, P4 Xeon 3.06 GHz, Myrinet / 2500 Dell	9819 15300	64% 3.9GFLOPS/CPU
5	Pacific Northwest National Laboratory United States/2003	Mpp2 Integrity rx2600 Itanium2 1.5 GHz, Quadrics / 1936 HP	8633 11616	74% 4.5GFLOPS/CPU
6	Los Alamos National Laboratory United States/2003	Lightning Opteron 2 GHz, Myrinet / 2816 Linux Networx	8051 11264	71% 2.9GFLOPS/CPU

Linpakは通信はそれほど問題とならない

⇒Itanium2は理論・実行性能が最も良い

⇒Itanium2はCPUあたりのスコアもG5と同程度

Itanium2はより少ないCPUでハイスコアが狙え、また高実行性能が得られるためコストパフォーマンスがよい

TOP500の3位～6位にみるシステム性能
ASCIプロジェクトは含まない⇒財源に限度あり

3	Virginia Tech United States/2003	X 1100 Dual 2.0 GHz Apple G5/Mellanox Infiniband 4X/Cisco GigE / 2200 Self-made	10280 17600	58% 4.7GFLOPS/CPU
4	NCSA United States/2003	Tungsten PowerEdge 1750, P4 Xeon 3.06 GHz, Myrinet / 2500 Dell	9819 15300	64% 3.9GFLOPS/CPU
5	Pacific Northwest National Laboratory United States/2003	Mpp2 Integrity rx2600 Itanium2 1.5 GHz, Quadrics / 1936 HP	8633 11616	74% 4.5GFLOPS/CPU
6	Los Alamos National Laboratory United States/2003	Lightning Opteron 2 GHz, Myrinet / 2816 Linux Networx	8051 11264	71% 2.9GFLOPS/CPU

逆に言うと、Itanium2は他のCPUほどノードを必要としないためシステムコストに占めるCPUの割合を小さくできる。
また、ノードが少ないということはネットワークコストの占める割合を大きくすることができる。ランニングコストも小さい。
⇒高速インターコネクトを用いたシステムの構築も可能

まとめ

システムの価格を概算し、Itanium2の価格性能比を考える

各CPUを使ったシステムの価格比較

今までのベンチマークによりItanium2が非常に高性能であることがわかった。

従来型のスーパーコンピュータであるSR，PRIMEPOWER、Alphaは、１ＴＦで10億程度の価格であり、その他のものも考慮してシステム換算すると100万円/GFLOPS程度である。

各CPUを使ったシステムの価格比較　続

これに対し、Pentium4で大規模システムを作ると、
１ＣＰＵあたり20万円＋ネットワーク20万円＋α＝60万円程度であり、実効性能効率を考慮すると20万円/GFLOPS。

Itanium2に関しても、ネットワーク込みで480万円程度で4wayのシステムが購入可能なので、20万円/GFLOPSとなる。

従来型のスパコン	100万円/GFLOPS
Pentium4	20万円/GFLOPS
Itanium2	20万円/GFLOPS

さらにIntel Compiler(Linux)は非商用であれば基本的にライセンス料が掛からない。

中･大規模のシステムで有利なItanium2

先ほどの議論で、Pentium4、Itanium2が従来型のシステムと比較して価格性能比が高いことを示した。

小規模のクラスタの場合は、ネットワークSW価格がほとんど無視できるが、ノード数が中規模･大規模になるとSW価格が急激に高騰するか、ネットワークSW自身が市場に存在しなくなる。(特注仕様もしくは、カスケード接続)

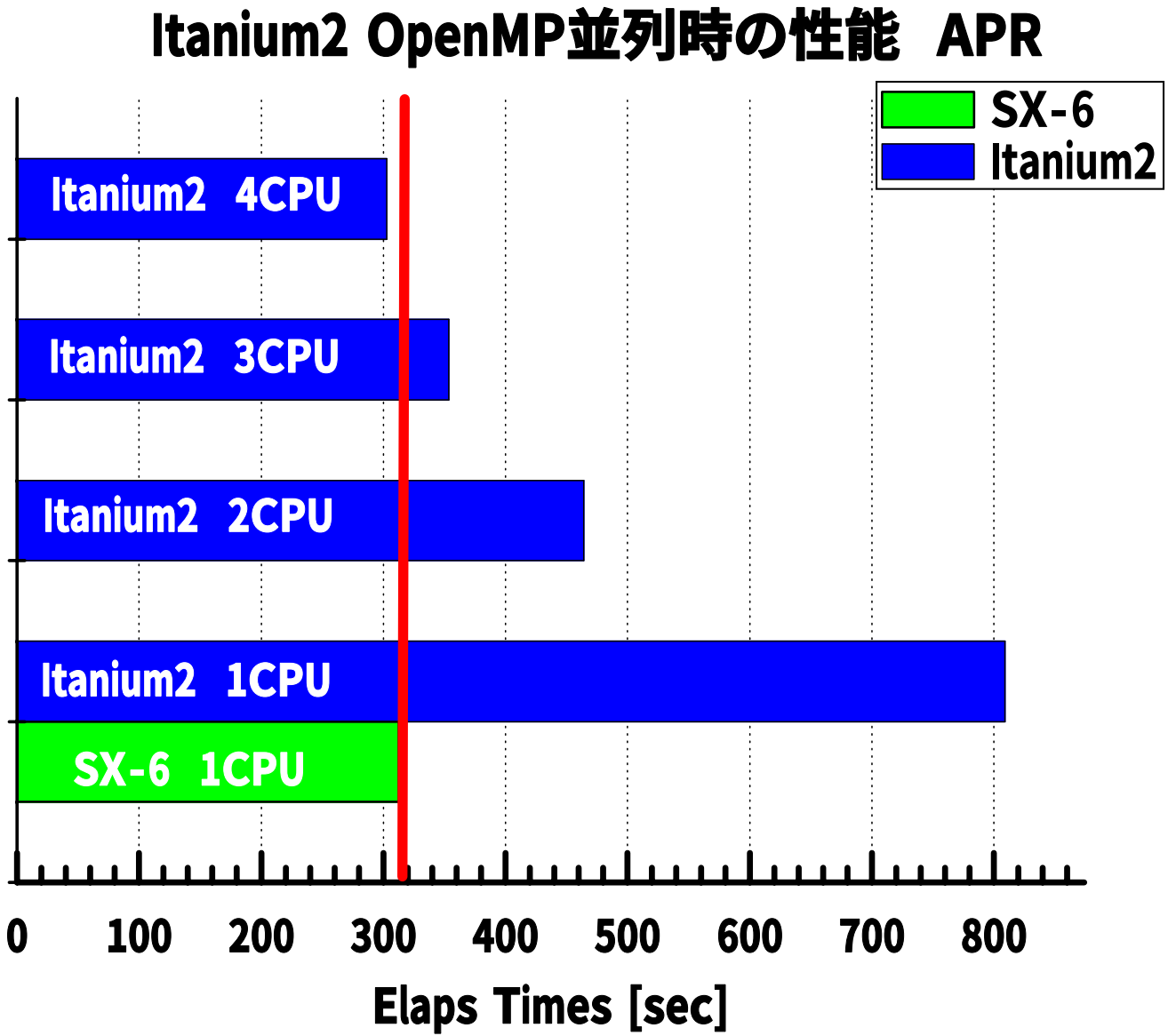
中･大規模のシステムで有利なItanium2　続

Pentium4(6GFLOPS)、Itanium2 (6GFLOPS*4)では、1ノードの理論性能比が4倍となり、Dual Xeon (6GFLOPS*2)のノードを使っても1ノードの理論性能比が2倍程度になる。

したがって、中規模･大規模のシステム構成には、Itanium2が非常に有利であると考えられる。

さらに、Itanium2システムでは、共有メモリ領域が非常に大きく取れるため非MPIコードの高速化も同時に期待できる。将来的にも、まだまだ高クロック化が見込まれる面も魅力的である。

期待される並列時のパフォーマンス
Itanium2 4way　V.S.　SX-6 1CPU



お知らせ

第2回大規模データマネージメントコンファレンス

開催日時：2004年3月17日-18日
開催場所：けいはんなプラザ ５Ｆ　黄河
（京都府相楽郡精華町）

参加費　：一般5,000円(事前振込3,000円)
　　学生1,000円
主催　　：けいはんな文化学術協会

詳細は、http://ldm-rg.com/ にて
お問い合わせは ldm-rg@nifty.com まで