HPC Linux Cluster
導入のためのシステム設計
利用の実際

2001年12月13日

株式会社　ソフテック
加藤　努

Cluster Conference
Tutorial Dec.13,2001

SofTek

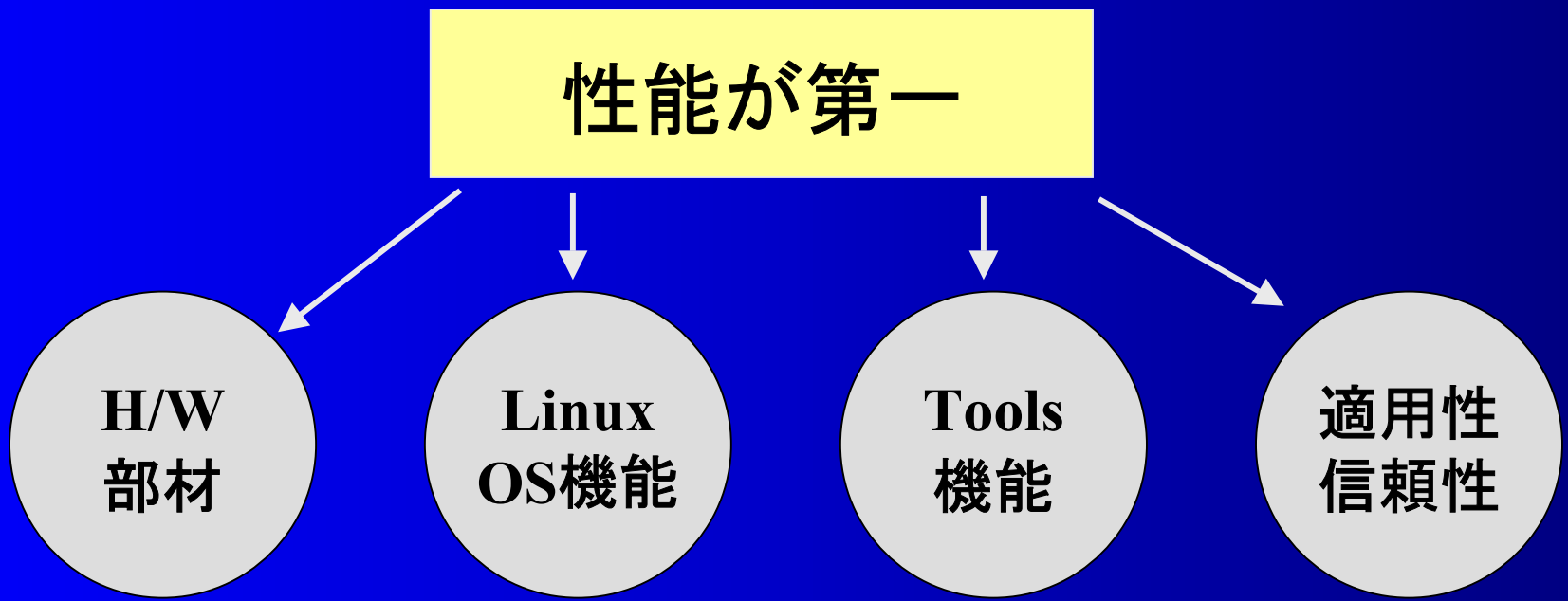
Linux PC Clusterは本当に使えるの？

- 性能はどうなの？
- 必要なハードウェア、周辺装置は、どうなの？
- 十分なソフトウェア環境の提供は、あるのか？
- ツール、ユーティリティは、揃っているの？
- 信頼性は、どうか？
- 専業の Cluster サービスプロバイダの有無

ハードベンダー依存の過去の呪縛から逃れる
High-end UNIX server、スパコンと遜色のない機能性を Linux Cluster は備えている

HPC の Mission

- High Performance Computingとは

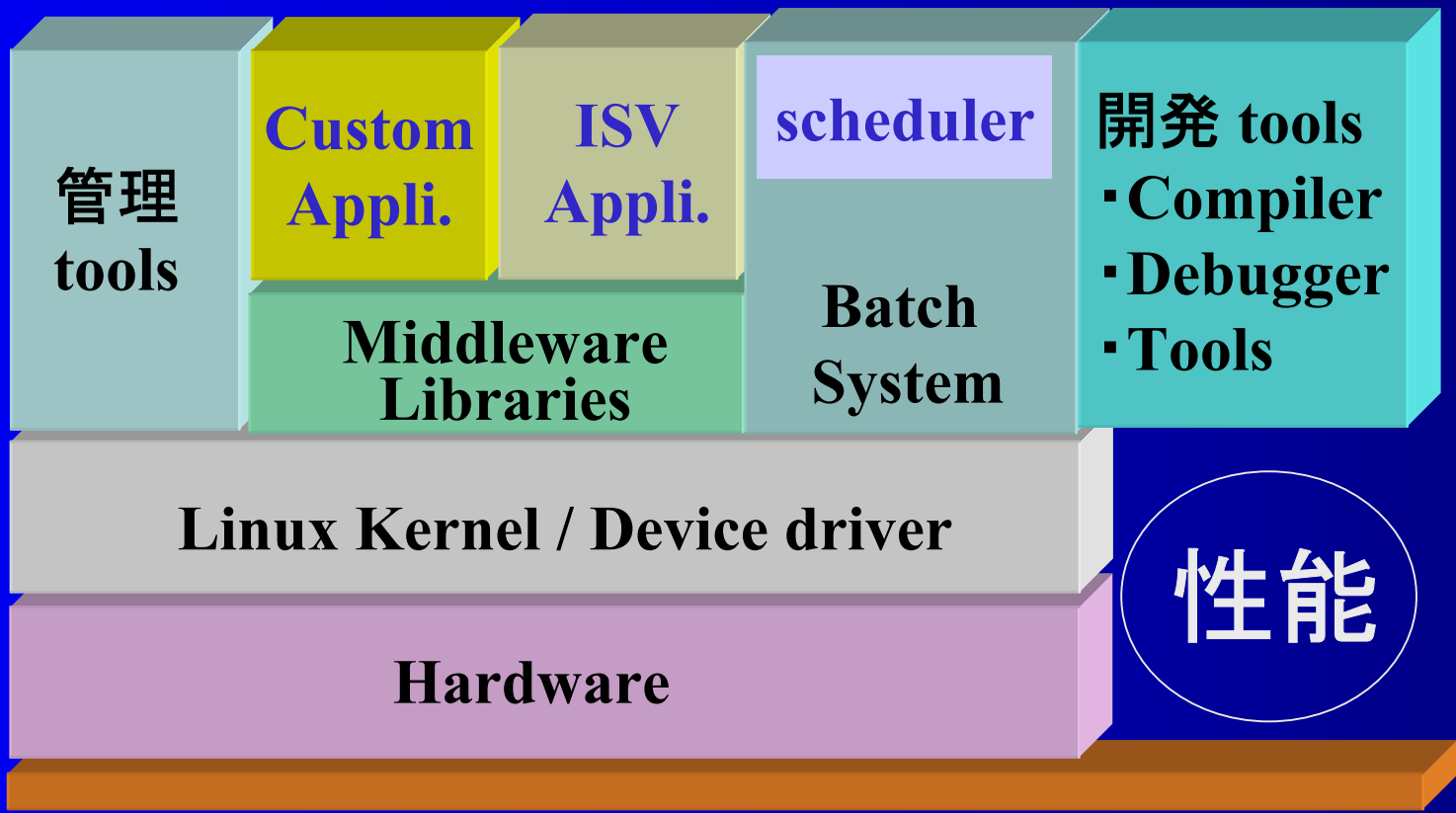


HPCは二つのカテゴリ

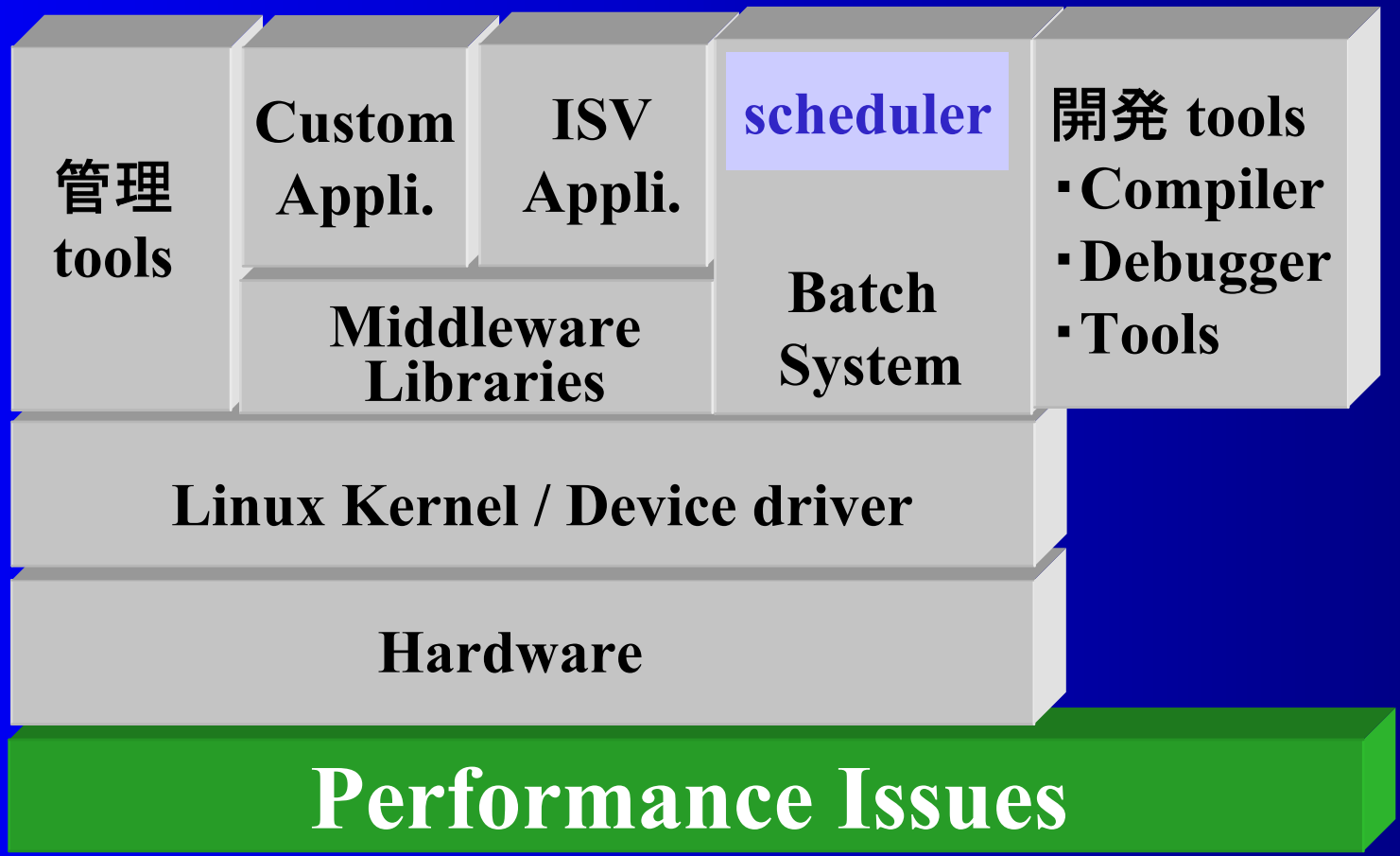
- 一つの処理をより高速に!
  - **High performance Computing**
  - 並列処理、並列プログラミング
  - 開発ツール、ライブラリとその機能性が重要
- 多数の処理をより高速に!
  - **High throughput computing**
  - 並行・分散処理を多数のノードで！
  - バッチシステム、Grid、専用ミドルウエアが重要

HPC システムのアーキテクチャ

Linux Cluster => Open source + Open document



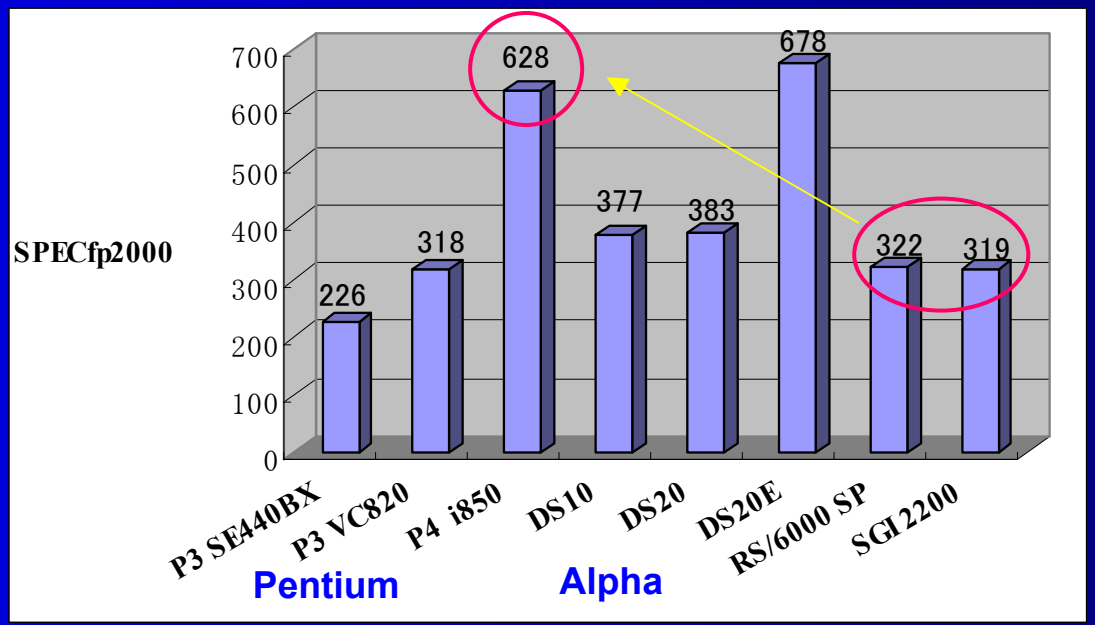
Pentium の性能について



SPECfp2000による実効性能比較

14 applications
- six FORTRAN77
- four Fortran90
- four C

SPEC95より実体を表す



SPECfp2000							
Intel-box			AlphaServer			IBM	SGI
P3 SE440BX	P3 VC820	P4 i850	DS10	DS20	DS20E	RS/6000 SP High node	SGI 2200 R12000
800 MHz	1 GHz	1.8 GHz	600 MHz	500 MHz	833 MHz	375 MHz	400 MHz
226	318	628	377	383	678	322	319
Ratio 1.00	1.41	2.78	1.67	1.69	3.00	1.42	1.41

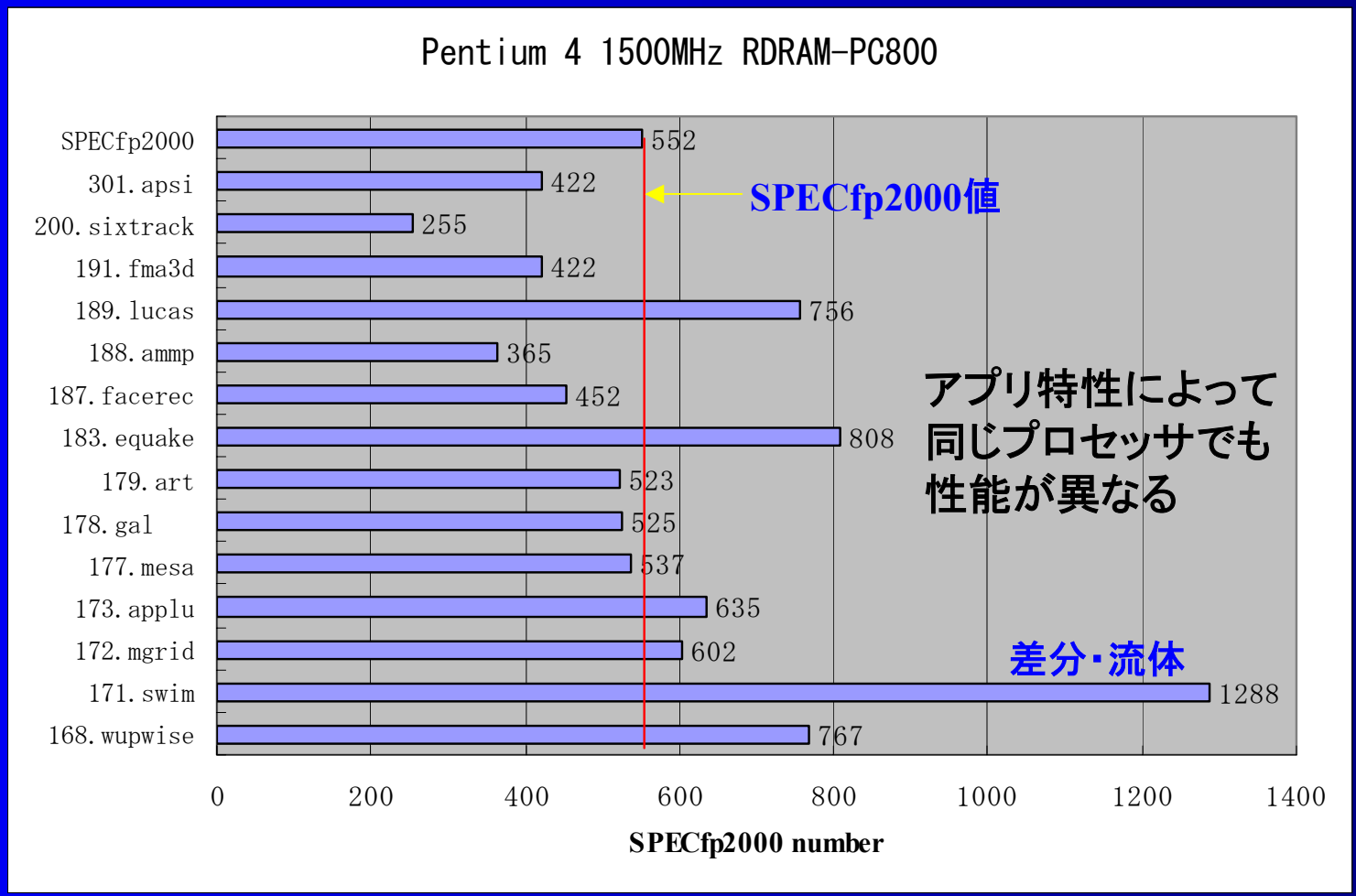
接近

SPECfp2000で使用するアプリケーション

様々なアプリケーションの性能の幾何学平均値
特性により、プロセッサ上での性能が大きく異なる

SPECfp2000	Language	Resident	Virtual	description
Code name		size(Mbytes)	size(Mbytes)	
168.wupwise	F77	176	177	Quantum chromodynamics
171.swim	F77	191	192	shallow water modeling
172.mgrid	F77	56	56.7	Multi-grid solver in 3D potential field
173.applu	F77	181	191	partial differential equations
177.mesa	C	9.5	24.7	3D graphics library
178.galgel	F90	63	155	Computational Fluid dynamics
179.art	C	3.7	5.9	Image recoginition/neural network
183.equake	C	49	51.1	Seismic wave propagation
187.facerec	F90	16	18.5	Image processing
188.ammp	C	26	30	Computational chemistry
189.lucas	F90	142	143	Number theory/primality testing
191.fma3d	F90	103	105	Finite-element crash simulation
200.sixtrack	F77	26	59.8	Nuclear physics accelerator design
301.apsi	F77	191	192	Meteorology:pollutant distribution

SPECfp2000：個々のプログラムの性能



Sun Ultra5 (300MHz) の性能を 100 とした相対性能

SPECfp2000による価格性能比

Processor	MHz	L2 cach	SPECint2K	SPECfp2K	Base Price¥	¥/SPECfp	Notes
Pentium 4	1800	256	586	628	320,000	510	(Dell,512MB,20GB)
Pentium III (Coppermine)	1000	256	428	314	185,000	589	(Dell,512MB-SDRAM,20GB)
AMD Athlon (Thunderbird)	1300	256	491	374	240,000	642	(GW2K, PC133 512MB-SDRAM,20GB)
AMD Athlon (Thunderbird)	1330	256	539	445	300,000	674	(価格推定、512MB-DDR SDRAM,20GB)
UltraSPARC III	750	8192	395	421	1,210,000	2,874	(Sun Blade1000)
Alpha (21264)	833	8192	533	678	2,000,000	2,950	(価格推定,DS20E)
UltraSPARC II	480	8192	234	291	1,210,000	4,158	(AXd
PA-8700	750	N/A	603	581	3,000,000	5,164	(価格推定)

Processor	¥/SPECfp	Ratio
Pentium 4	510	1.0
Pentium III (Coppermine)	589	1.2
AMD Athlon (Thunderbird)	642	1.3
AMD Athlon (Thunderbird)	674	1.3
UltraSPARC III	2,874	5.6
Alpha (21264)	2,950	5.8
UltraSPARC II	4,158	8.2
PA-8700	5,164	10.1

Pentium 4の実効性能の目安

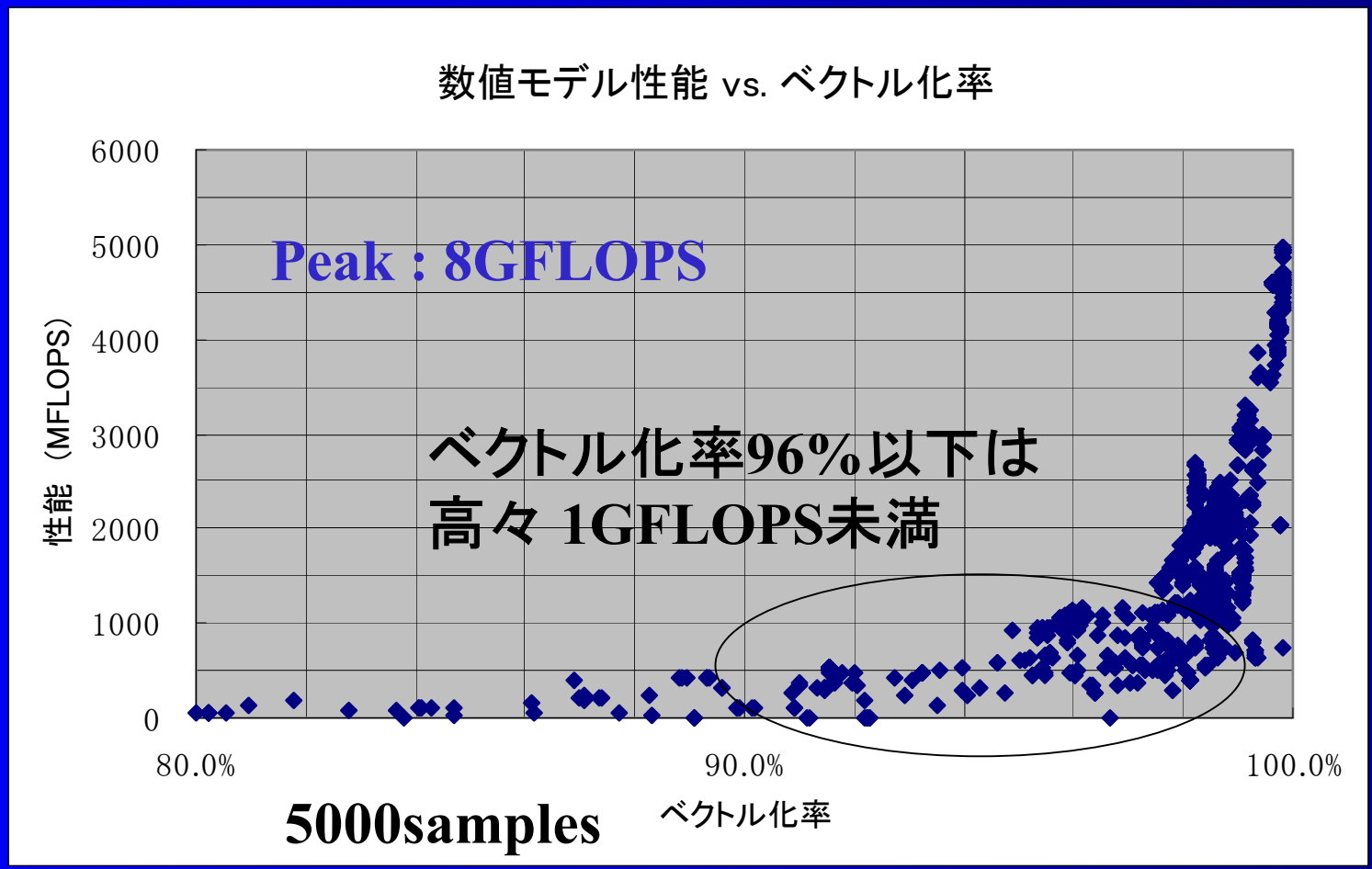
Pentium 4 : 1.8GHz RDRAM PC800

実アプリケーションの実効性能

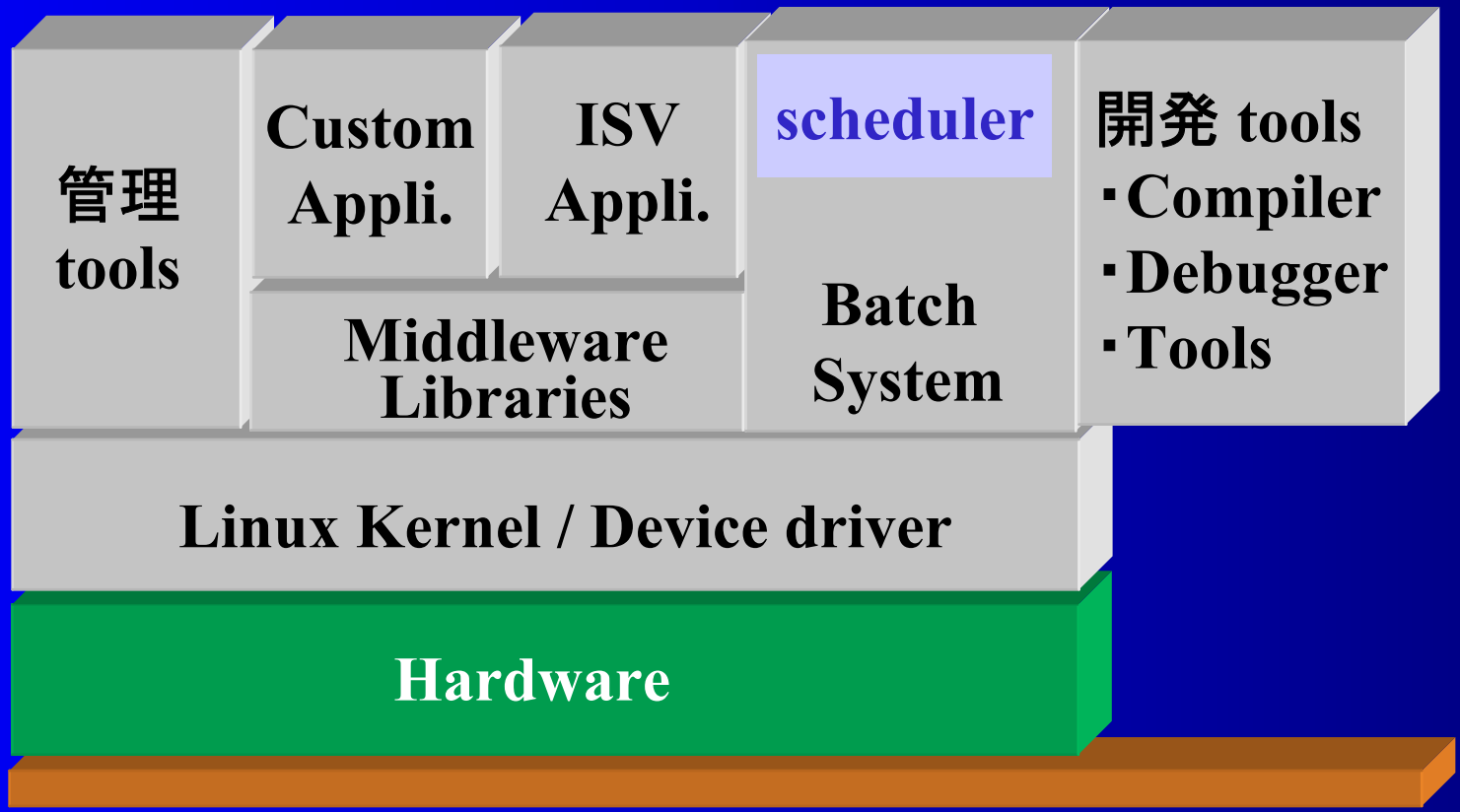
倍精度計算　：　200 ～300 MFLOPS
単精度計算　：　400 ～570 MFLOPS

約10年前のベクトル・スパコンの性能並
1CPU：2 ～3 億円のプロセッサと同じ

最近のベクトル・スパコン性能の実態



HPC PC Cluster の「ハードウェア」



ハードウェア(1)

- 筐体
  - タワー型（安価）、ラックマウント型（高価）
  - Size：1U～2U, 4U～5U (Pentium 4)
- プロセッサ
  - Pentium III , Pentium 4, AMD Athlon
  - Single CPU or Dual CPU
- メモリ
  - SDRAM , DDR SDRAM , Rumbus DRAM
  - SDRAM (CAS Latency 2 or 3)

実効メモリバンド幅 (STREAM)

- STREAM benchmark
  - Cache 外アクセスでの実効的なメモリ転送レート

Processor	MHz	STREAM (MB/s)	RATIO	RAM
Pentium 4	1500	2144	3.94	RDRAM PC800
Alpha (21264) DS40E	833	1429	2.63	SDRAM
Ultra SPARC III	750	890	1.64	SDRAM
AMD Athlon (Thunderbird)	800	586	1.08	SDRAM
Pentium III (Coppermine)	733	544	1.00	SRDRAM PC133

ハードウェア(2)

- PCI バス
  - 32bit/33MHz (chipset : i815,i820,i845,i850)
  - 64bit/66MHz (chipset : i860,Serverworks)
- ハードディスク
  - IDE （大容量可能、低価格、DMAの設定で高速）
  - SCSI（9GB、18GB、36GB、価格高い、高速）
  - Hardware RAID 0,1,5 (File server用途、可用性）
- RAID5
  - Ultra 160 SCSI対応 ATA-RAID
  - 500GB で100万円前後

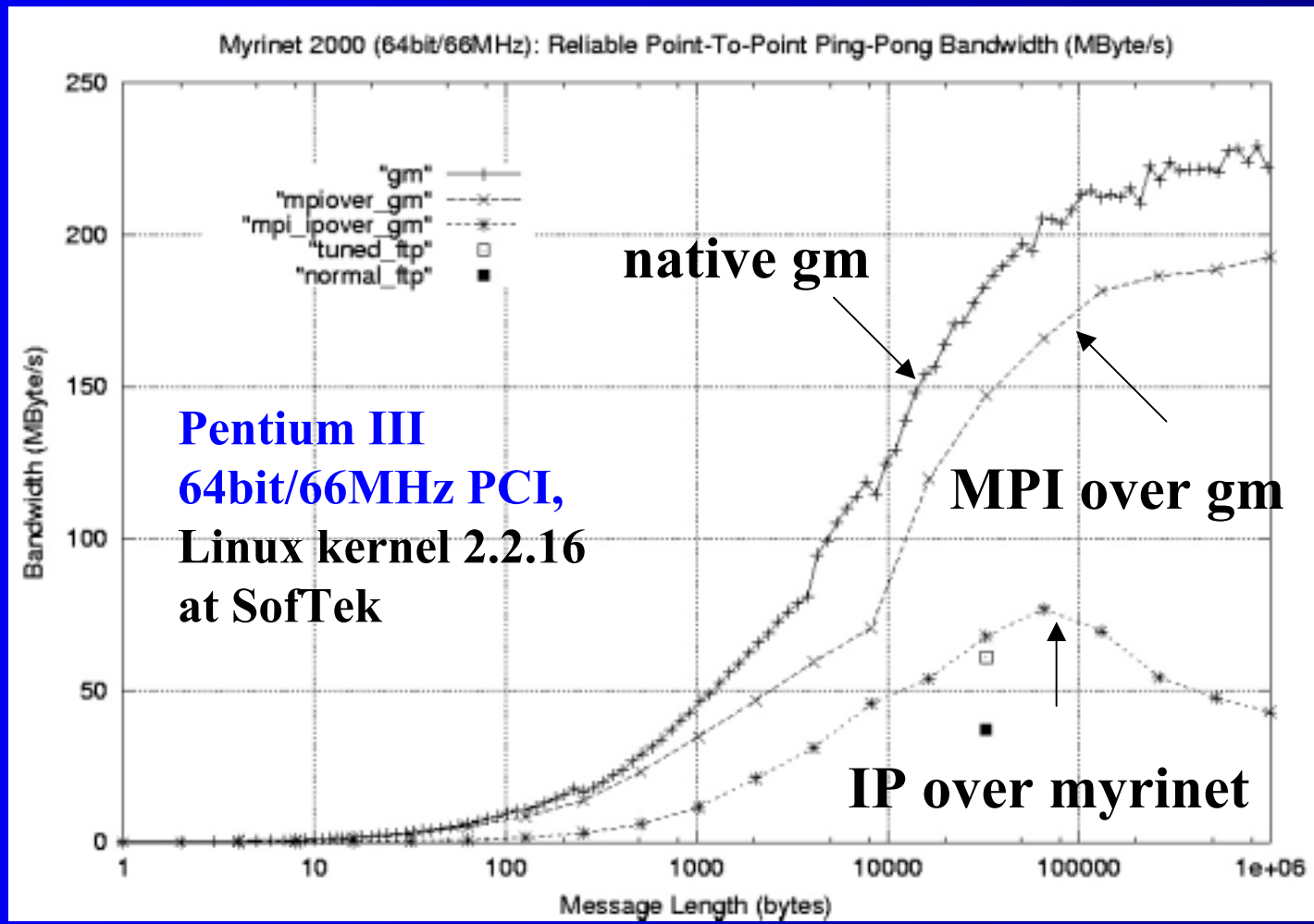
ハードウェア(3)

- ネットワーク装置
  - Fast Ethernet　　(TCP/IP or 低遅延通信S/W)
  - Gigabit Ethernet (TCP/IP or 低遅延通信S/W)
  - Myrinet2000 (PCI 32bit/33MHz , 64bit/66MHz)
  - Giganet cLAN (VIA)
- 周辺機器
  - Network Attached Storage(NAS) for Linux
  - Backup Server for Linux
  - Console切替器

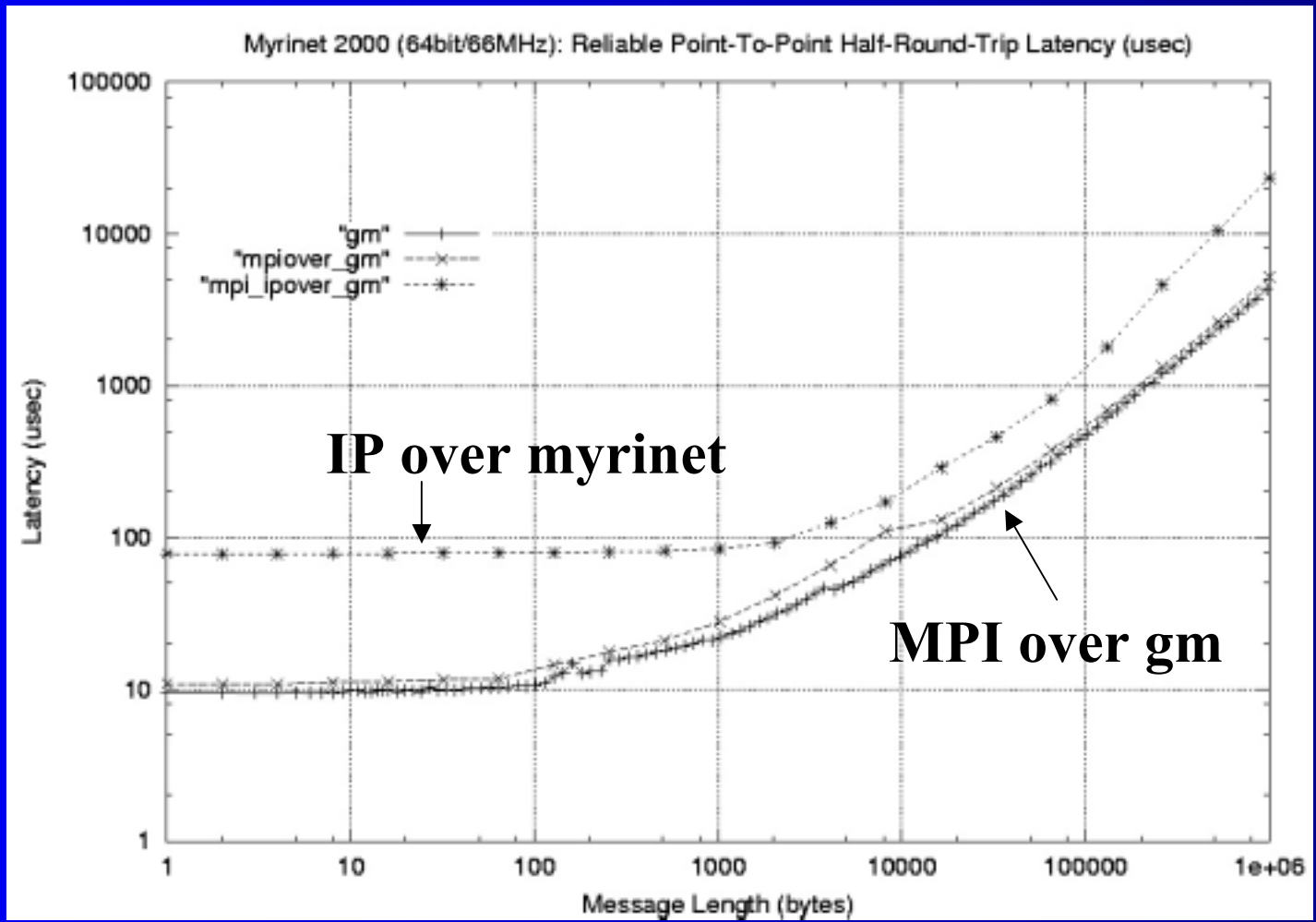
各通信媒体の特性

	Fast Ethernet	Gigabit Ethernet	Myrinet 2000	Giganet (cLAN)
MPI Bandwidth	11 (MB/sec)	20-40 (MB/sec)	140-200 (MB/sec)	105 (MB/sec)
MPI Latency	150～ (usec)	120～ (usec)	10～18 (usec)	20 (usec)
MPI support	MPICH MPI/Pro	MPICH MPI/Pro	MPI over GM	MPI/Pro MVICH
Cost NIC+Switch (32node)	22.5万円+ 25万円 (TCP/IP)	100万円+ ?? 高い!! (TCP/IP)	700万円+ 300万円 (GM)	???? (VIA)

Myrinet2000 Bandwidth



Myrinet2000 Latency



通信 FastEthernetかMyrinetか？

- 通信媒体の違いと性能の関連：アプリに依存する
- 8 Pentium III 550MHzを使用（8CPU-MPI並列時）

（実効性能比）	100Base	Myrinet	
1)LU(Linpack)	1	1.23	相対的に通信量少ない
2)CG(Sparse)	1	1.66	Myrinet買っても損しない
3)MM5	1	1.1	通信の最適化Good!
4)ATM simulation	1	13.0	通信遅延小のMyrinet

1) Dense matrix computation (CPU intensive & Minimum Comm.)

2) Sparse matrix computation (適度な通信量あるのでMyrinet有利)

3) Regional気候モデル（数値モデル自体が並列最適化良：good scalability)

4) 離散事象シミュレーション（ネットワーク解析）

A lot of small messages (tiny messages)==>Myrinet有利(Latency)

PCクラスタの設計（1）

- 小規模（PC4～16台）クラスタ
  - 1台を Master Node として共用
    - Gateway / NIS /NFS server　機能
    - 開発用フロントエンド機能
    - バッチシステムの Master Server
  - その他を Slave node
    - 並列計算、バッチノード要素
  - ネットワーク
    - FastEthernet 2系統望ましいが、1系統でもよし
    - 必要であれば、Myrinet-Net を必要ノード数のみ実装
    - Network Switch は16 port用で十分

PCクラスタの設計(2)

- 中規模(16～32台)クラスタ
  - 1台を Master Node として独立あるいは共用
    - Dual Processor であればなお良し
    - Gateway / NIS /NFS server　機能
    - 開発用フロントエンド機能、バッチシステムのMaster Server
  - Reliability 重視であれば RAID を MASTER に接続
  - ネットワーク
    - FastEthernet 2 系統望ましいが、1 系統でもよし
    - 必要であれば、Myrinet-Netを必要ノード数のみ実装
    - Network Switch は32 port用で、Backplane Busが高速なもの

PCクラスタの設計(3)

- 大規模(64台以上)クラスタ
  - 1台を Master Node として独立あるいは共用
    - Dual Processor であればなお良し
    - Gateway / NIS機能
    - 開発用フロントエンド機能、バッチシステムのMaster Server
  - NFS server としてNASを独立(メーカ品は高い、安価でも信頼できる製品は市場に存在する) ➔ Gigabit SWへ
  - ネットワーク
    - FastEthernet
    - 必要であれば、Myrinet-Netを必要ノード数のみ別実装
    - Network Switch はFE 32 +32 → Gigabit SWのTree 構造

設置性と価格、熱対策と信頼性

- 設置性を考えたら、やっぱりラックマウントなの？

	DeskTop筐体	ラック筐体	
型名	Dell Dimension	Dell PowerEdge-Rack	
周波数	866MHz/128MB	866MHz →	700MHz ←
高さ	大きい	2U	1U ←
価格	20	42.4	54.4（万円）
価格比	1	2.1	2.72

高い!

- 信頼性を上げるには、筐体熱対策に尽きる

  格好良さ（サイズ）を求めると高速MPUは使えない

  でも、……日本は狭い…

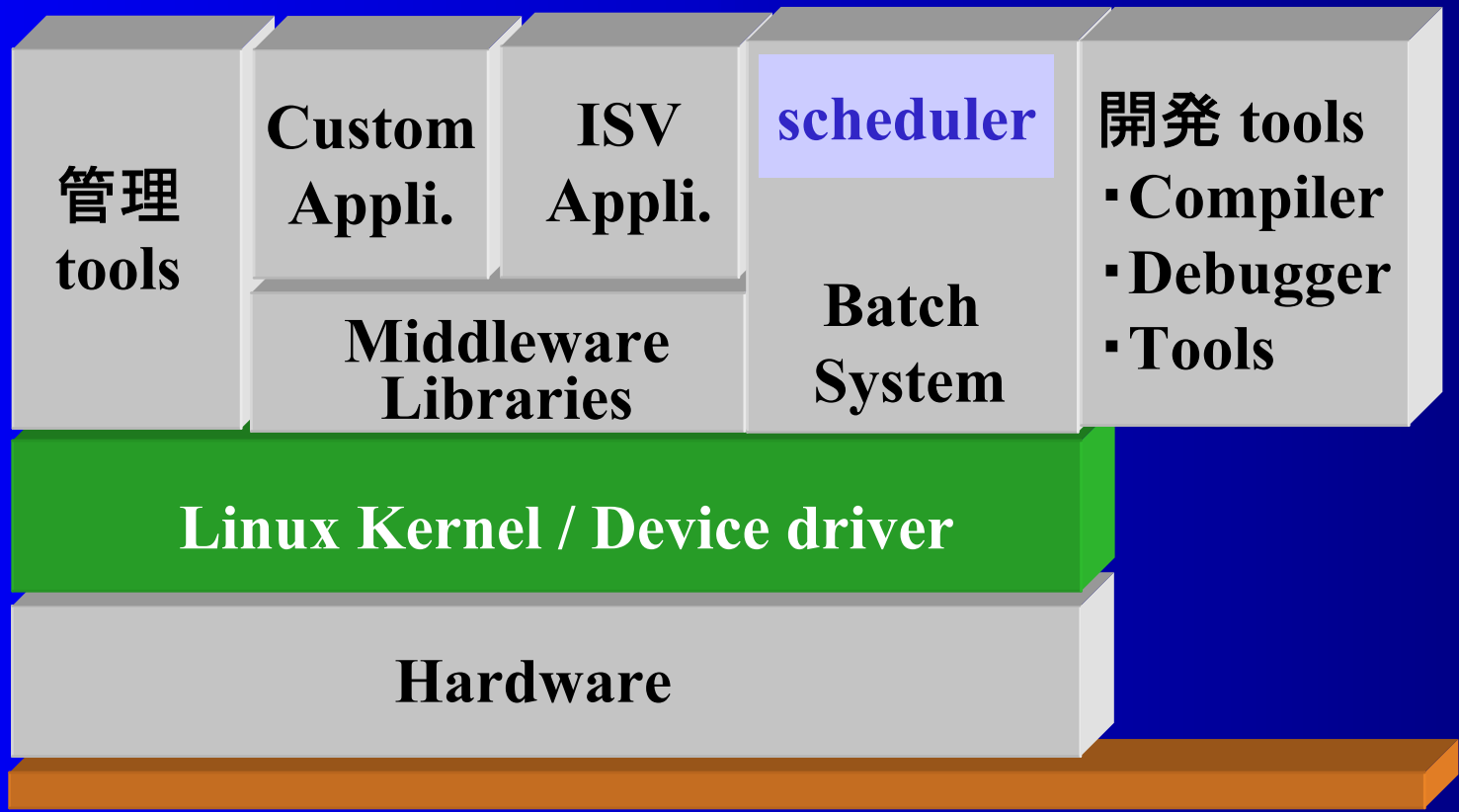
障害時の対処の考え方

- クラスタは、疎結合のPC独立システム
- １台故障時には、その切り離しが可能
- バッチ処理の設定等の構成を変更するだけ
- Disk 障害時には、システムイメージコピー

  → 経験豊富なテクニカルサービスプロバイダの選択必須
- ハードウェアで耐障害性を（on Linux)
  - RAID装置
  - Backup装置
  - NAS装置

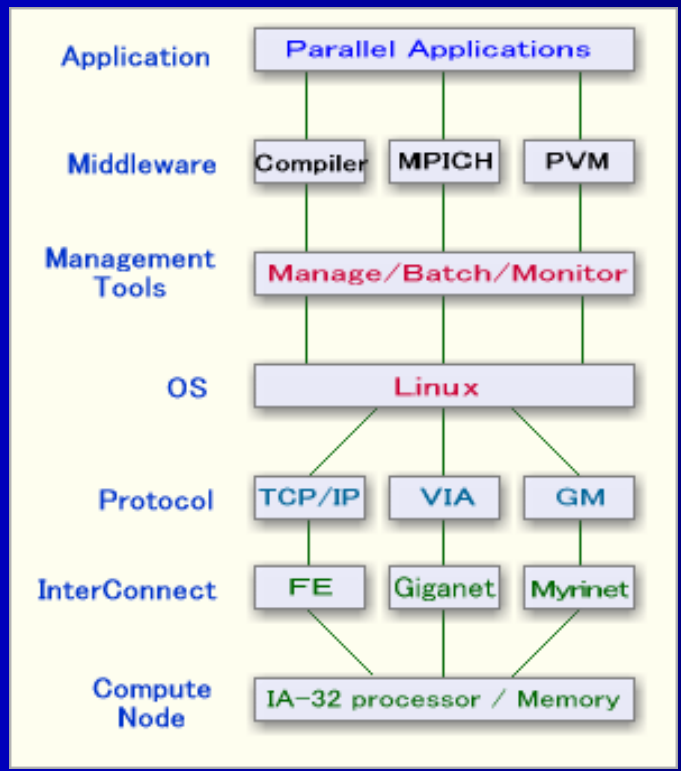
  高いメーカ品だけでなく、最適なコストのものあり

HPC PC Cluster「基本ソフトウェア」



基本ソフトウェア

- Linux 2.2.x, 2.4.x
- TCP/IP
- NFS v2 / v3
- NIS
- rsh / ssh
- Power control
- 各種drivers

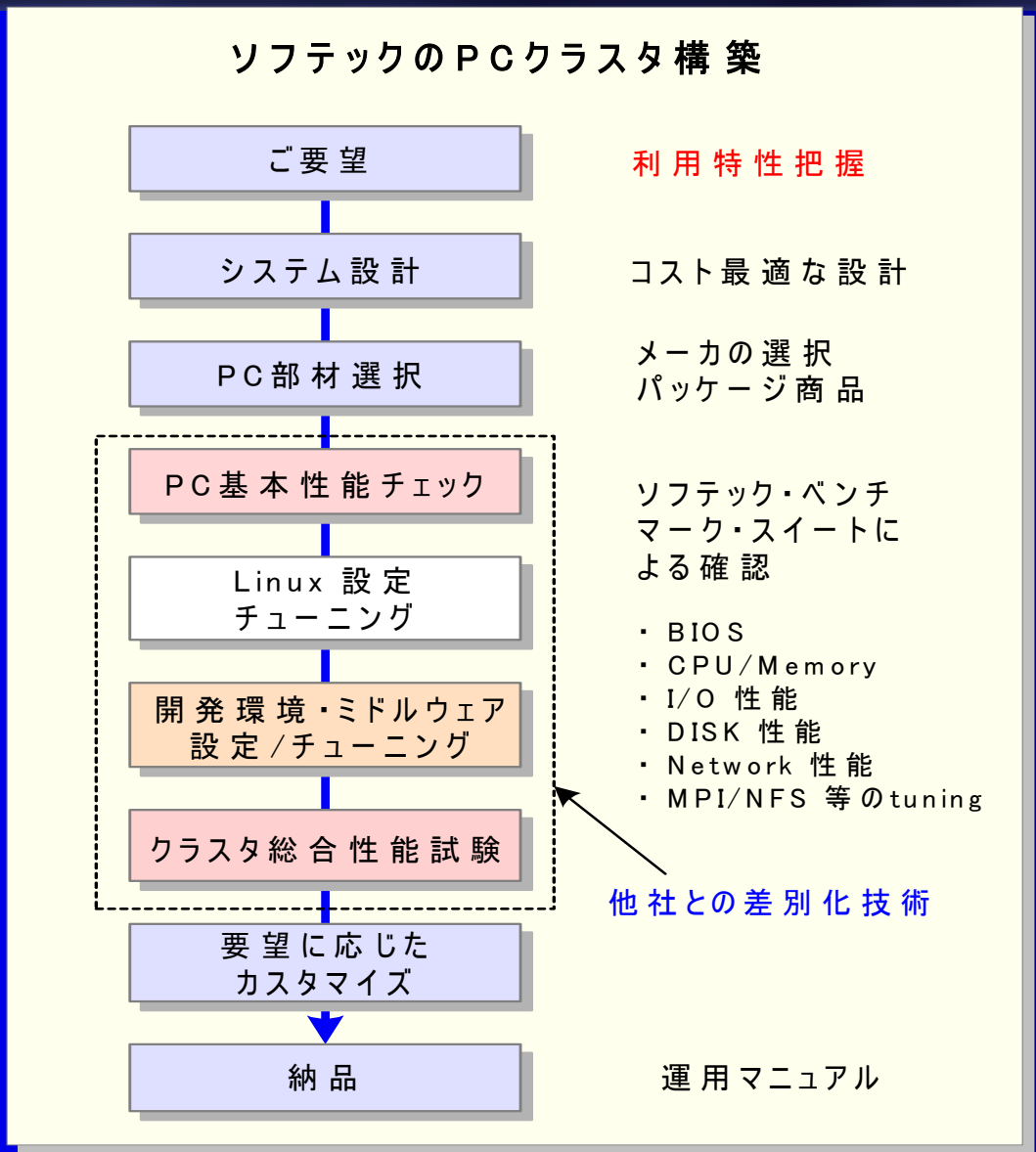


機能性はHigh-end UNIX server と同じ

HPC Linux Distribution Package ?

- HPC Linux 用のパッケージが多くなってきた
  - 商用パッケージは高い（ノード数に依存）
  - 但し、自動的にソフトウェアがインストール可能
  - しかし、中身は全てFree で入手できるもの
  - 買う？買わない？
- ただ、ソフトウェアを入れただけでは、十分な性能を引き出せない!!
- パッケージを買ってもテクニカルサポートが必要
  - Package Fee ＋ Support Fee のコスト
- 最新のS/Wを検証して、技術が分かるサービスプロバイダが必要。（HPC Package Feeはいらない）

弊社のHPC Clusterの構築



システムチューニング

- システムの状況を知る
  - */proc/partitions*, Disk partition 状況
  - */proc/cpuinfo*, CPU 特性
  - */proc/pci*, PCI devices
  - */proc/interrupts*, IRQs 情報
  - */proc/dma*, DMA channels 情報
  - */proc/ioports*, I/O port address ranges
- システムチューニング
  - Memory/Swap, IDE disk, TCP/IP,File I/O

IDE / SCSI Disk 性能の評価

IDE

```
root@photon0 root]# hdparm -Tt /dev/hda6
/dev/hda6:
 Timing buffer-cache reads:   128 MB in  1.09 seconds =117.43 MB/sec
 Timing buffered disk reads:  64 MB in  3.39 seconds = 18.88 MB/sec
[root@photon0 root]# hdparm -Tt /dev/hdb
/dev/hdb:
 Timing buffer-cache reads:   128 MB in  1.08 seconds =118.52 MB/sec
 Timing buffered disk reads:  64 MB in  3.17 seconds = 20.19 MB/sec
```

SCSI

```
root@photon7 /root]# hdparm -Tt /dev/sda5
/dev/sda5:
 Timing buffer-cache reads:   128 MB in  0.68 seconds =188.24 MB/sec
 Timing buffered disk reads:  64 MB in  2.29 seconds = 27.95 MB/sec
```

ファイルシステム

- NFS v2
  - Max. file size < 2GB
- NFS v3
  - NFS v2 + ジャーナリング機能（保守性向上）
  - NFS v2 の上位互換性あり（相互変換可能）
  - Max. file size < 2GB（large file patchあり）
- ReiserFS
  - ジャーナルファイルシステム
  - Max. file size < 4GB , B+ツリー構造
  - 小さなファイルを複数扱う処理速い
  - NFS, Databese, Bioinfomatics系のアプリに向く

NFS v3 性能--bonnie++ Benchmark

Server : PentiumIII 800MHz – **Dual processor**
: HDD: ST340016A;UDMA(33);2MB-cache
: **buffered disk reads: 64 MB ;19.22 MB/sec**
Client : Pentium 4 1.8GHz
OS : Linux 2.4.7 with **Fast Ethernet**

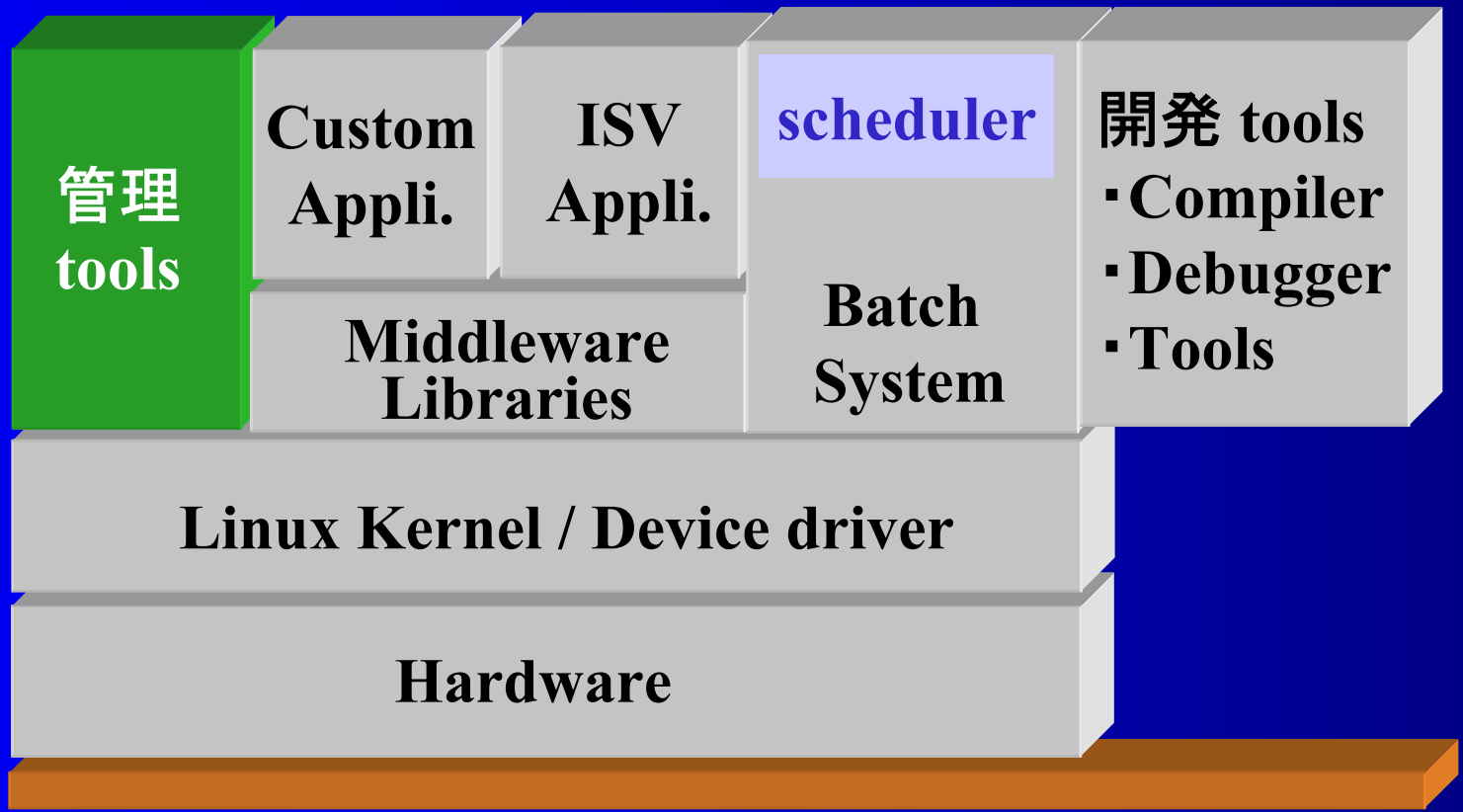
・NFS サーバのデーモン数 : 8 プロセス起動
・NFSクライアントの rsize、wsize : 8,192 Byte
・Server File system : **ext3**

	------Sequential Output----------						---Sequential Input---				--Random--	
	-Per Char-		--Block---		-Rewrite--		-Per Char-		--Block---		--Seeks---	
	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	/sec	%CPU
512MB	10619	98.2	10603	6.9	2940	2.5	6908	68.7	8602	2.0	1781.4	8.9
1024MB	11309	57.5	11335	3.7	11412	4.2	21978	100.0	1691710	100.0	7079.7	10.6
1536MB	11358	58.3	11149	3.7	2644	1.6	7967	40.8	8947	1.1	1396.2	1.4

システムの自動ブート・シャットダウン

- システムBIOS の設定
- ネットワーク・ブート（Wake-on-LAN)可能
  - 対応可能なNIC 必要
- リモートパワーコントロール可能
  - BIOS の設定必要
- cron により Power On/Off スケジュール可能

HPC PC Cluster「管理ツール」



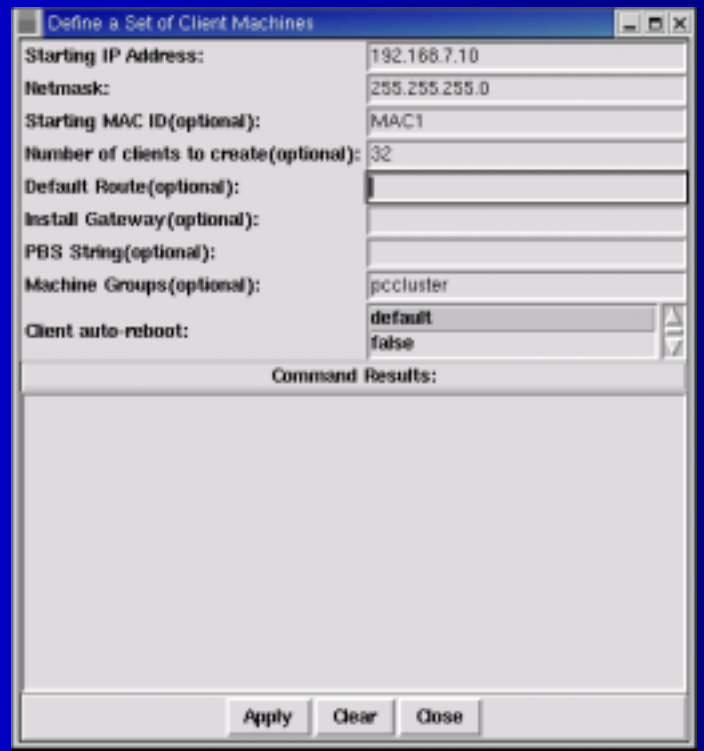
LUI (Linux installation for Cluster)

- **LUI (The Linux Utility for cluster Installation)**
  - PXEブート対応NICを使用、フロッピーや CD等の bootメディアを一切使うこと無く、ネットワーク経由でLinuxをインストール可能。
  - インストールしたいアプリケーション（RPMパッケージ）を細かく指定することが可能
  - クラスタを構成する各 PCの IPアドレス、ホスト名等を自動設定
  - PCの HDD内容をまるごとコピー（クローニング）可能
  - Linux OS以外にも、事前設定で、各種ユーザ／商用アプリケーションも自動的にインストール可能
  - GUIおよびコマンドラインで操作可能
  - コマンドラインレベルでスクリプトを作成すれば、定期的な PCのバックアップツールとしても使用可能

http://oss.software.ibm.com/developerworks/projects/lui

LUI の GUI

- GUI 経由で様々な設定可能
- 但し、様々な知識を必要とする



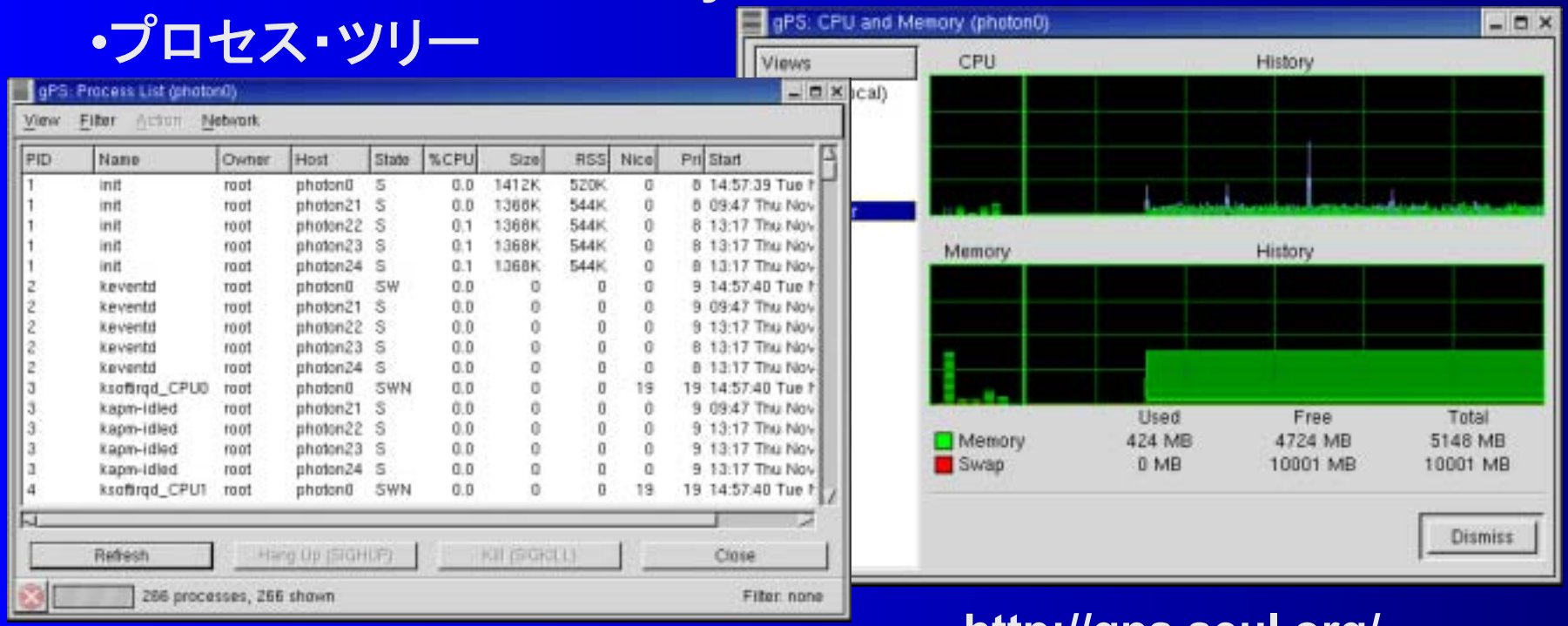
LUI の効果

- Network 経由のインストレーションの自動化
  - 30台のLinux & Other S/Wのインストール：20分
  - 各種 config file を Master で設定しておく
  - インストール後、Client node は即利用可能
  - Red Hat 付属の Kick Start 等では、同様におこなうためにFloppy, CDの入れ替え必要

- Disk 障害時のノードの復旧を迅速にできる
  （これが、最も重要な効果である）
  復旧用の disk imageを保存しておくこと

gPS (graphical Process Statistics)

- 全ノードのプロセス・リスト／プロセス詳細情報
- CPU/Memory usage history
- ユーザ毎のCPU/Memory使用状況
- プロセス・ツリー



http://gps.seul.org/

bWatch（リモート監視）

- 各ノードのシステム利用状況の監視
  - CPU / Memory / # of Users / # of process
  - Static な状況監視 （システム負荷軽い）

http://www.sci.usq.edu.au/staff/jacek/bWatch/

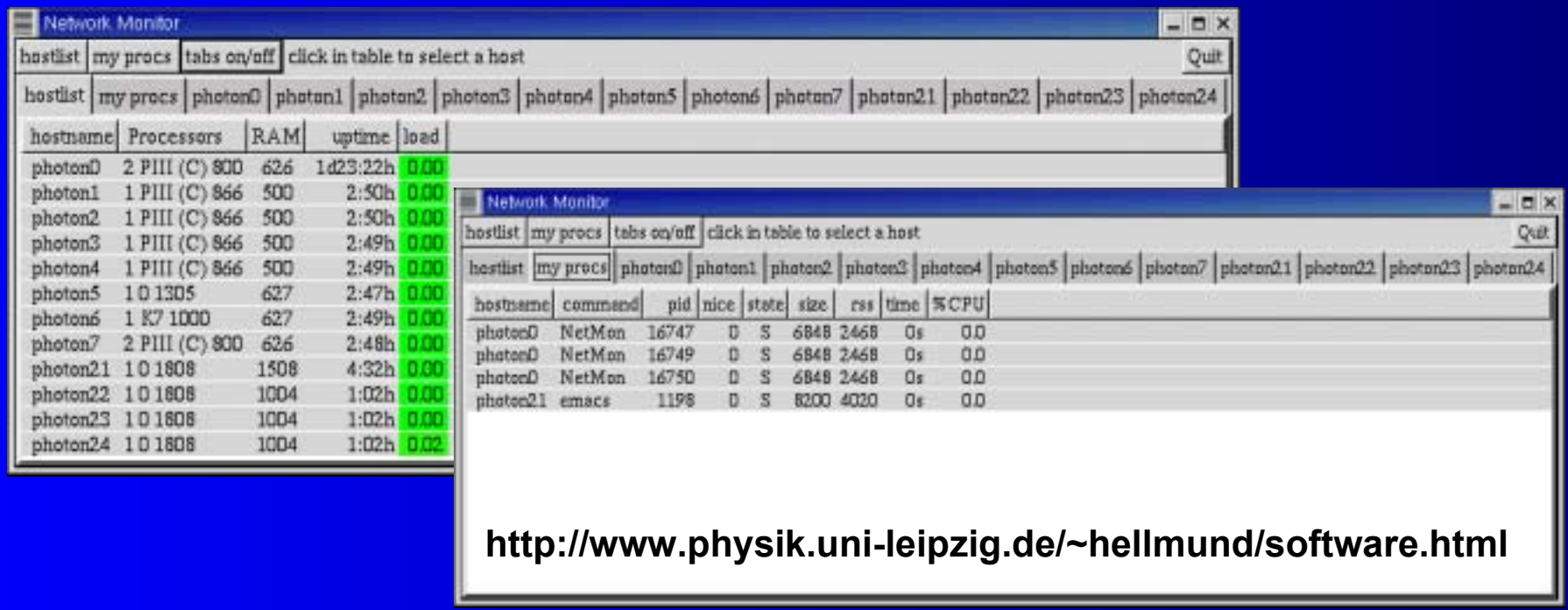
bWatch 1.0.3

Host Name	Num Users	Time	1 min Load	5 min Load	15 min Load	Num procs.	Total Mem	Free Mem	Shared Mem	Buffers	Cache	Total Swap	Free Swap
photon0	7 users	4:16	0.01	0.00	0.00	92	626 Mb	187740 Kb	0 Mb	114 Mb	91 Mb	2000 Mb	2000 Mb
photon1	0 users	4:16	0.00	0.00	0.00	48	500 Mb	480108 Kb	0 Mb	3 Mb	12 Mb	2000 Mb	2000 Mb
photon2	1 user,	4:16	0.00	0.00	0.00	49	500 Mb	476124 Kb	0 Mb	3 Mb	15 Mb	2000 Mb	2000 Mb
photon3	0 users	4:16	0.00	0.00	0.00	45	500 Mb	482448 Kb	0 Mb	2 Mb	11 Mb	2039 Mb	2039 Mb
photon4	0 users	4:16	0.00	0.00	0.00	45	500 Mb	482384 Kb	0 Mb	2 Mb	11 Mb	2000 Mb	2000 Mb
photon5	0 users	4:16	0.00	0.00	0.00	46	627 Mb	607740 Kb	0 Mb	2 Mb	11 Mb	2000 Mb	2000 Mb
photon6	0 users	4:16	0.00	0.00	0.00	45	627 Mb	608196 Kb	0 Mb	2 Mb	11 Mb	2000 Mb	2000 Mb
photon7	0 users	4:16	0.00	0.00	0.00	43	626 Mb	603652 Kb	0 Mb	5 Mb	11 Mb	2000 Mb	2000 Mb
photon21	1 user,	4:16	0.00	0.00	0.00	50	1508 Mb	1433232 Kb	0 Mb	44 Mb	19 Mb	2000 Mb	2000 Mb
photon22	0 users	4:16	0.00	0.00	0.00	46	1004 Mb	995024 Kb	0 Mb	2 Mb	11 Mb	2000 Mb	2000 Mb
photon23	0 users	4:16	0.00	0.01	0.00	47	1004 Mb	864204 Kb	0 Mb	2 Mb	37 Mb	2000 Mb	2000 Mb
photon24	0 users	4:16	0.00	0.01	0.00	47	1004 Mb	962832 Kb	0 Mb	2 Mb	12 Mb	2000 Mb	2000 Mb

Refresh Exit

NetMon (プロセスモニター)

- network-wide process monitor
- “Top” のような振る舞い (Daemon必要)
- リモート・プロセスをKillできる



http://www.physik.uni-leipzig.de/~hellmund/software.html

Pconsole (parallel console)

ttyを制御してリモートホストへコマンドを直接投入

システム管理上、全てのホストの同一変更の必要時に有効

コマンドウィンドウ

Basic Cluster Scripts(BCS)コマンド

- palive - create a list of up (alive) nodes
- puseradd - parallel user add
- puserdel - parallel user delete
- pusersync - synchronize the password files
- pup - check to see if nodes are up
- pload - show the load for each node
- pswap - show the swap usage for each node
- ploadu - show the largest load for a user on each node
- preboot - reboots cluster nodes
- pshutdown - shutdown cluster nodes
- pcp - execute a copy in parallel
- pexec - execute a command in parallel

http://www.plogic.com/bcs/

Monitoring hardware performance

- UNIX でお馴染みのユーザコマンド
  - top
  - vmstat

```
procs                  memory    swap       io     system         cpu
 r  b  w   swpd   free   buff   cache  si  so    bi    bo   in   cs  us  sy  id
 0  0  0   6664   7272 196836 199008   0   0     0     0  105   17   0   1  99
```

  - iostat

Device:	tps	Blk_read/s	Blk_wrtn/s	Blk_read	Blk_wrtn
dev3-0	1.20	3.10	21.49	2012578	13961644
dev3-1	0.50	20.95	40.21	13609728	26116056

  - netstat

プロセスアカウンティング

- sa コマンド
- プロセス・アカウンティング (UNIX の pacct)

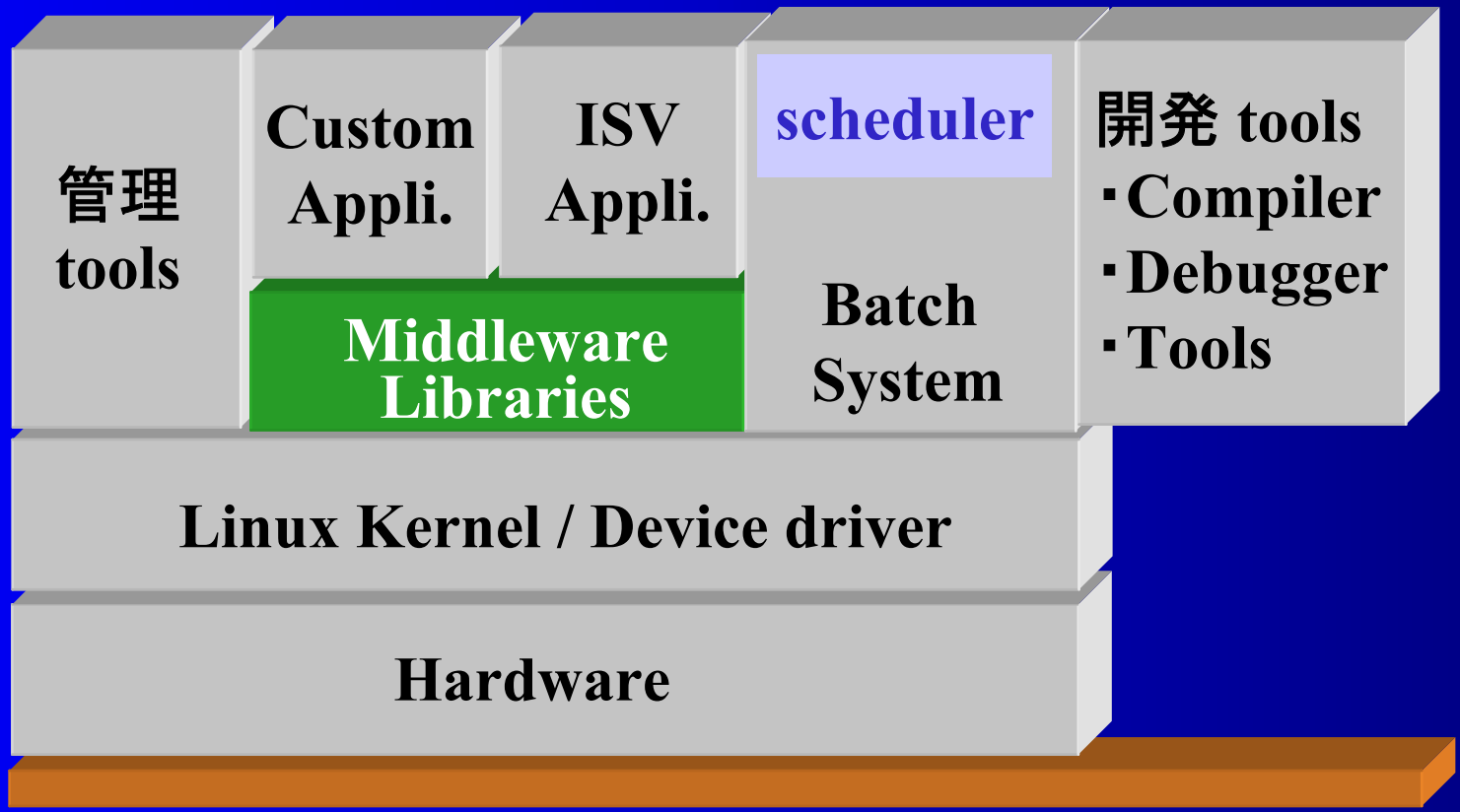
ID	real time	cpu time	I/O	Kore	command
3696	0.26re	0.18cp	0avio	349k	cat
160	0.14re	0.14cp	0avio	531k	strings
964	33.59re	0.13cp	0avio	431k	make
5096	12.37re	0.12cp	0avio	516k	sh*
143	0.47re	0.10cp	0avio	350k	sadc

System Activity Report (sar)

- High-end UNIX & Supercomputer上で動作
- Linux 2.4 で標準サポート
- CPU / Memory / Disk / Network / File system
- Paging / swap 他各種のシステム性能のレポート
- これらの情報のLogging

03:35:43 AM	CPU	%user	%nice	%system	%idle
03:35:44 AM	all	0.00	0.00	1.50	98.50
03:35:45 AM	all	0.00	0.00	0.50	99.50

HPC PC Cluster「ミドルウェア」



High Performance Computingの中身

- 一つの処理をより高速に!
  - **High performance Computing**
  - 並列処理、並列プログラミング
  - 開発ツール、ライブラリとその機能性が重要
- 多数の処理をより高速に!
  - **High throughput computing**
  - 並行・分散処理を多数のノードで！
  - バッチシステム、Grid、専用ミドルウエアが重要

並列処理のMiddleware

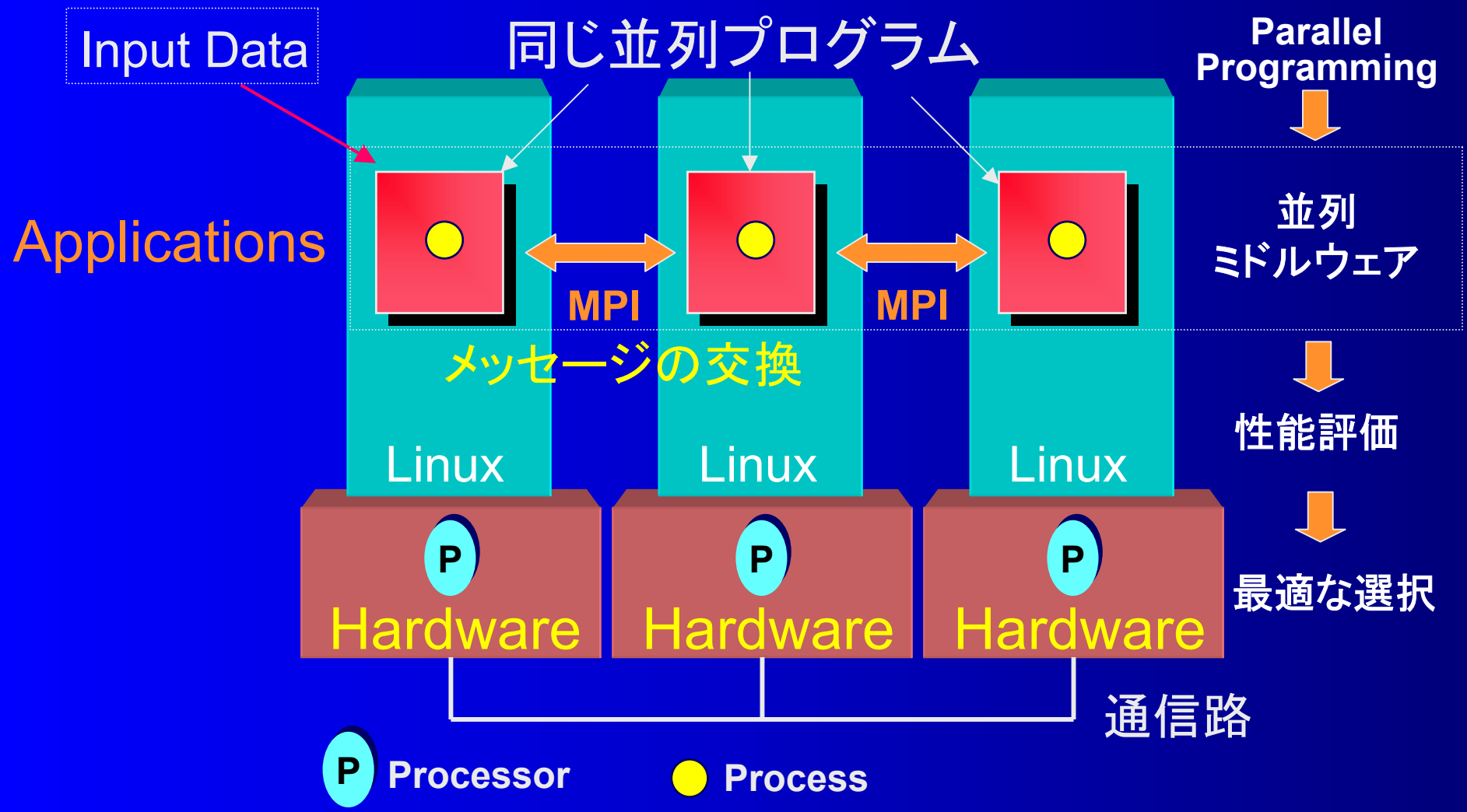
- High Performance Computing
  - 並列計算時の Message Passing Interface(MPI)
  - mpich (http://www-unix.mcs.anl.gov/mpi/mpich/)
  - LAM/MPI (http://www.lam-mpi.org/)

  -------- Low Latency communication S/W ----
  - MPI/Pro (http://www.mpi-softtech.com/)
  - MVICH (http://www.nersc.gov/research/FTG/mvich/)
  - MPI-GM (http://www.myri.com)

分散処理のMiddleware,subsystem

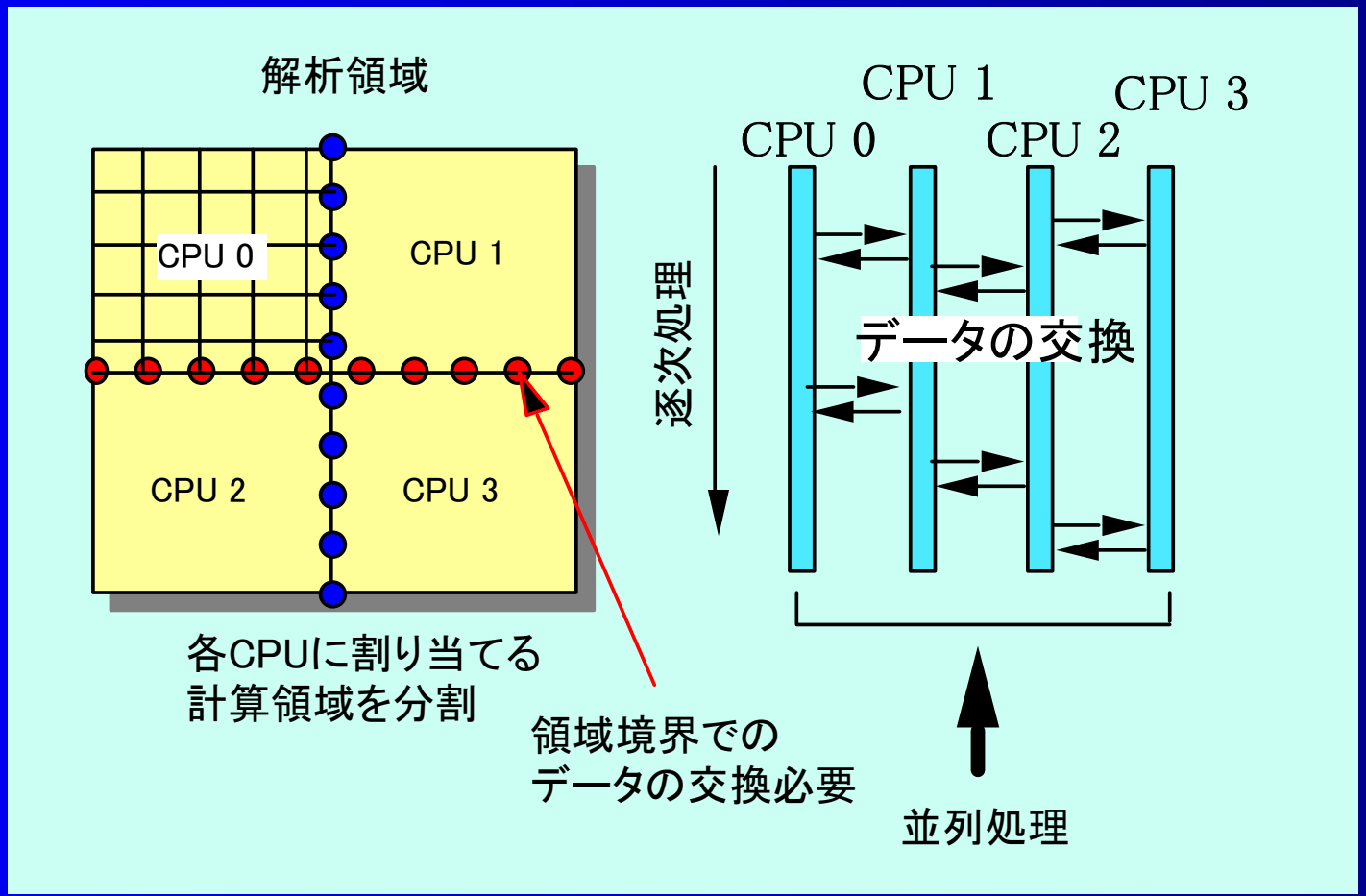
- High throughput Computing
- 多数のジョブを制御し、スケジューリングする機能
  - Parametric study 専用のツール ： EnFuzion
  - MOSIX (明日の講演） 自動負荷分散処理
  - Batch system
  - Grid Engine tool

High Performance Computing（並列）

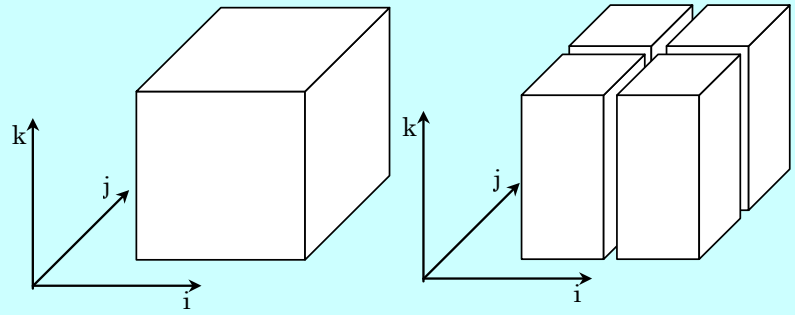


MPIを使用した並列計算

領域分割法

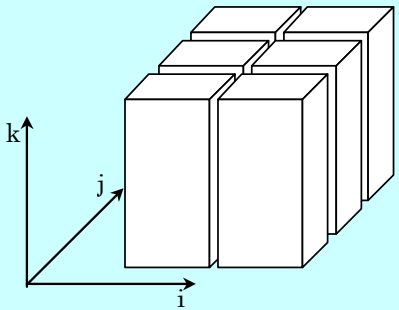


MPI並列化/領域の分割指針の一例



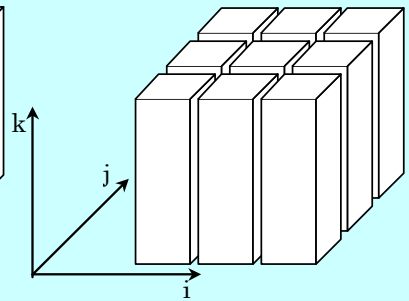
全体領域

i,j 方向とも 2 分割



i,方向 2 分割

j 方向 3 分割



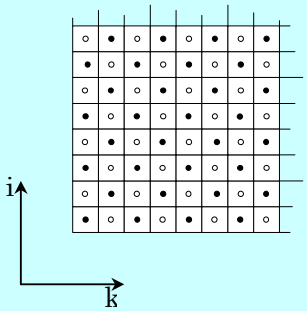
i,方向 3 分割

j 方向 3 分割

通信量を減らすためには、どの方向？

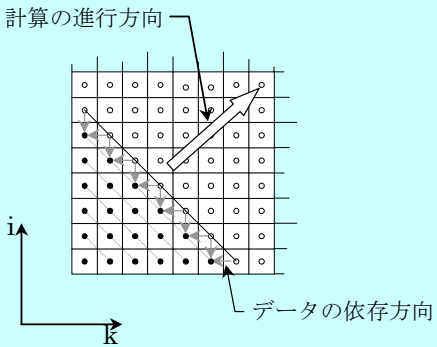
RedBlack法、Hyperplaneによる並列化

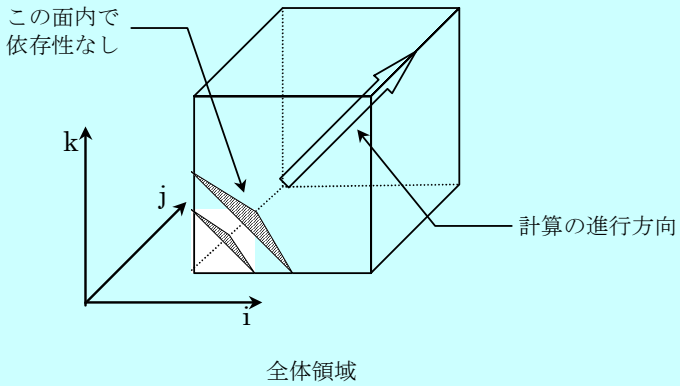
RedBlack



red-black 法：‘●’で表されるセルは一つ前のタイムステップの‘○’のセルのデータを用いて他の‘●’のセルのデータに依存することなく独立に解ける．同様に‘○’のセルも他の‘○’のデータに依存することなく独立に解ける．

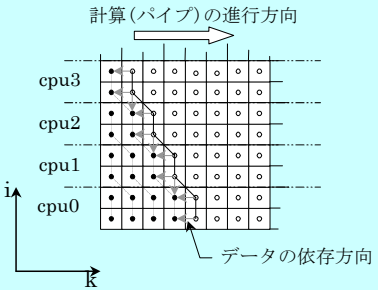
Hyperplane

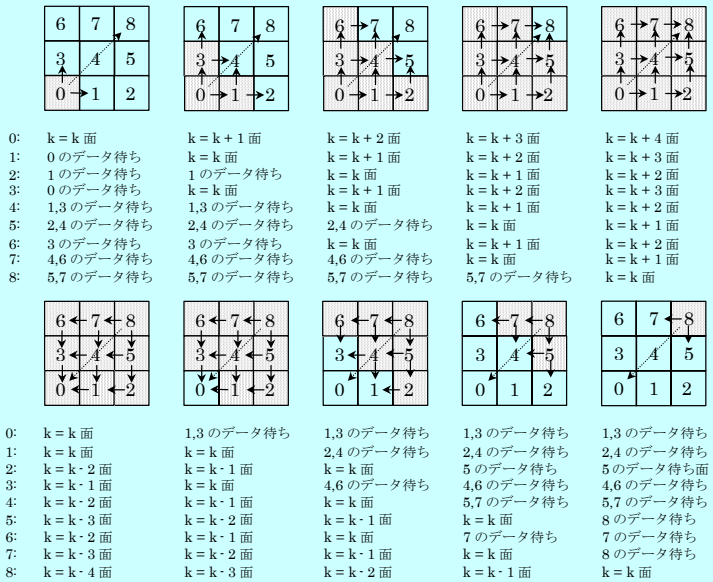




全体領域

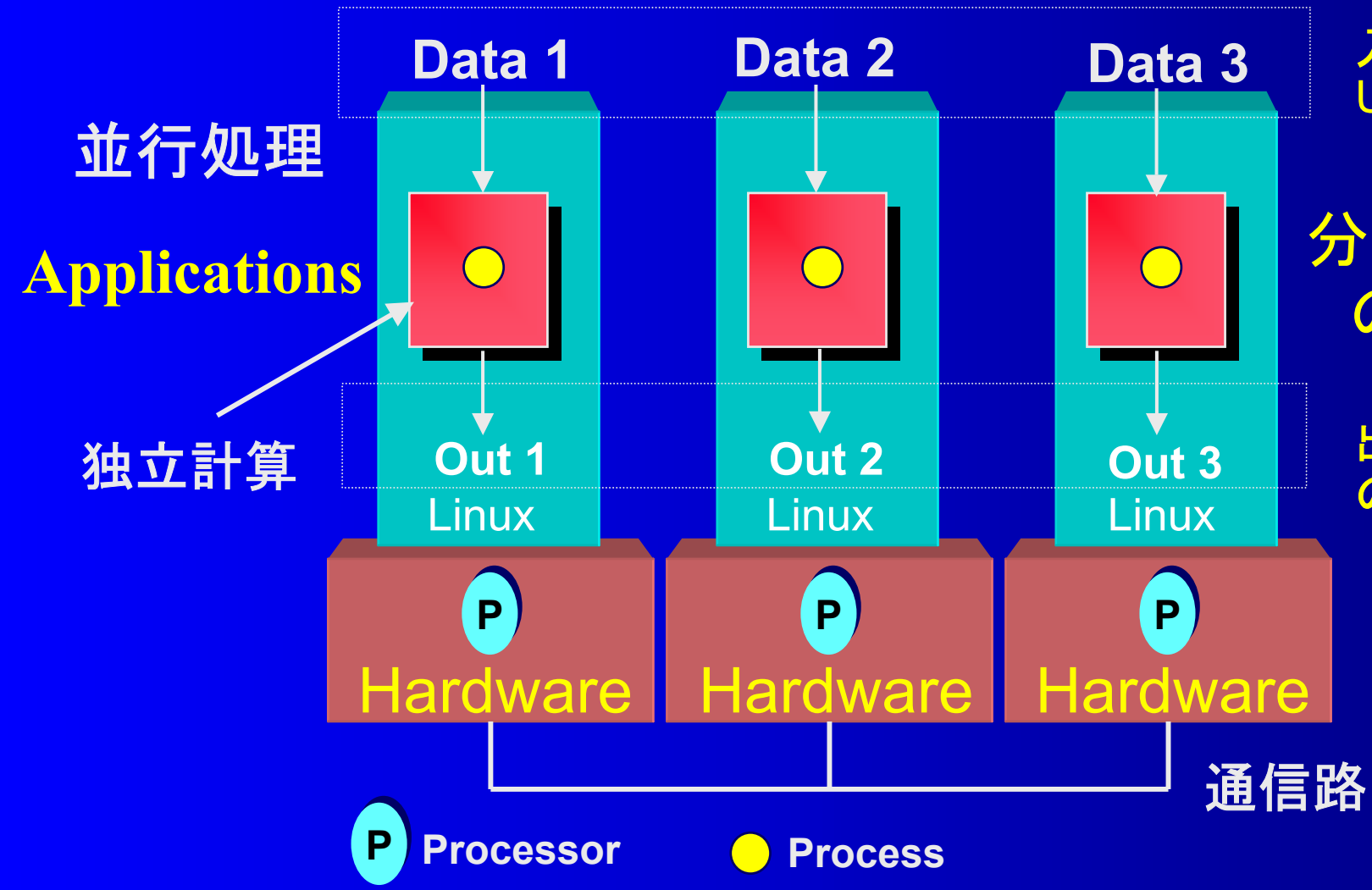
並列処理のパイプライン化



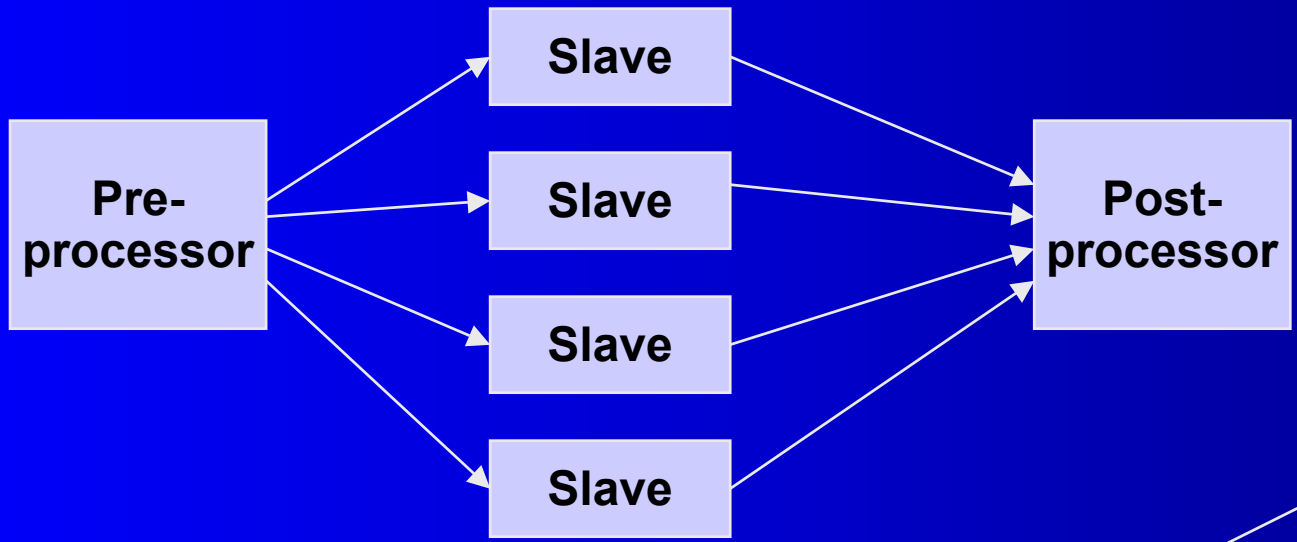


通信処理の最適化のために、処理のパイプライン化を設計

High throughput Computing（分散処理）



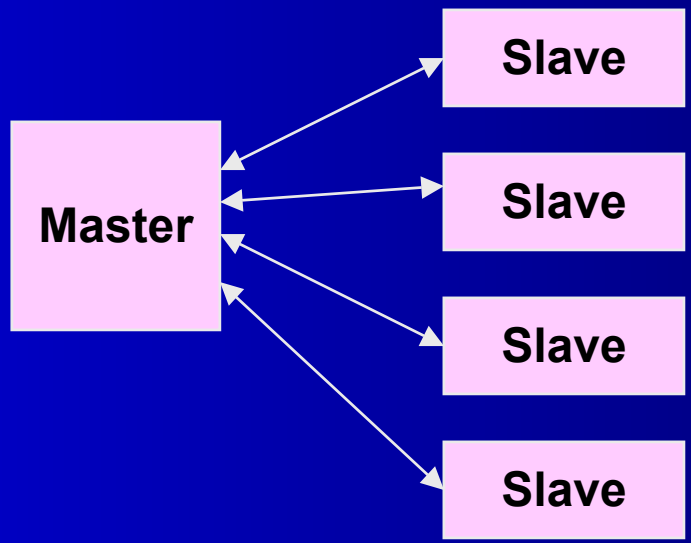
Master-Slave Parallel Computing



Script + Batch
or
Toolを利用

Supports parametric execution

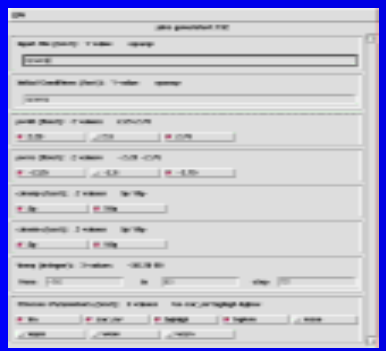
- Execute programs
- Varying parameters
- Simple scatter/gather

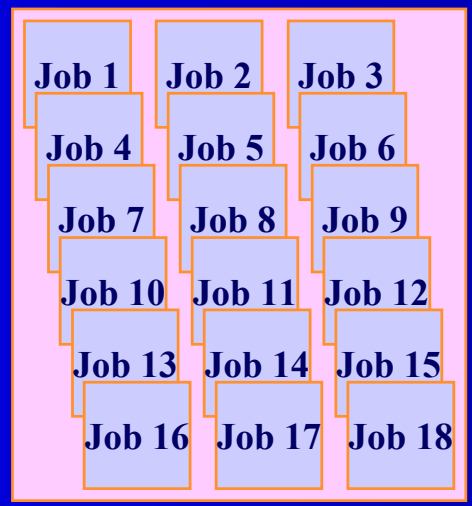


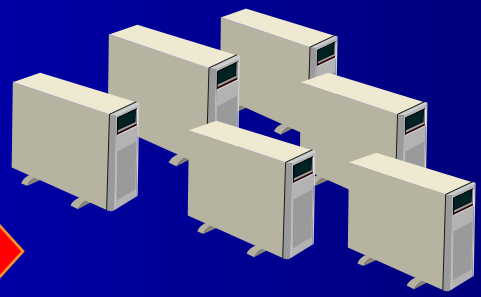
分散(並行)処理ソフトウェア EnFuzion

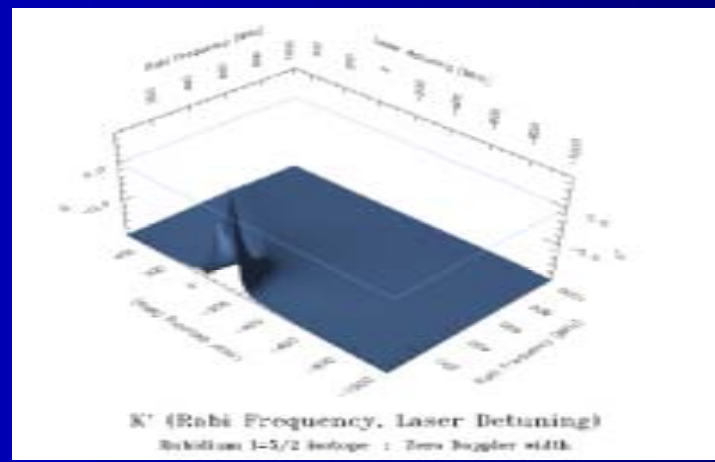
TurboLinux / *SofTek*

EnFuzion



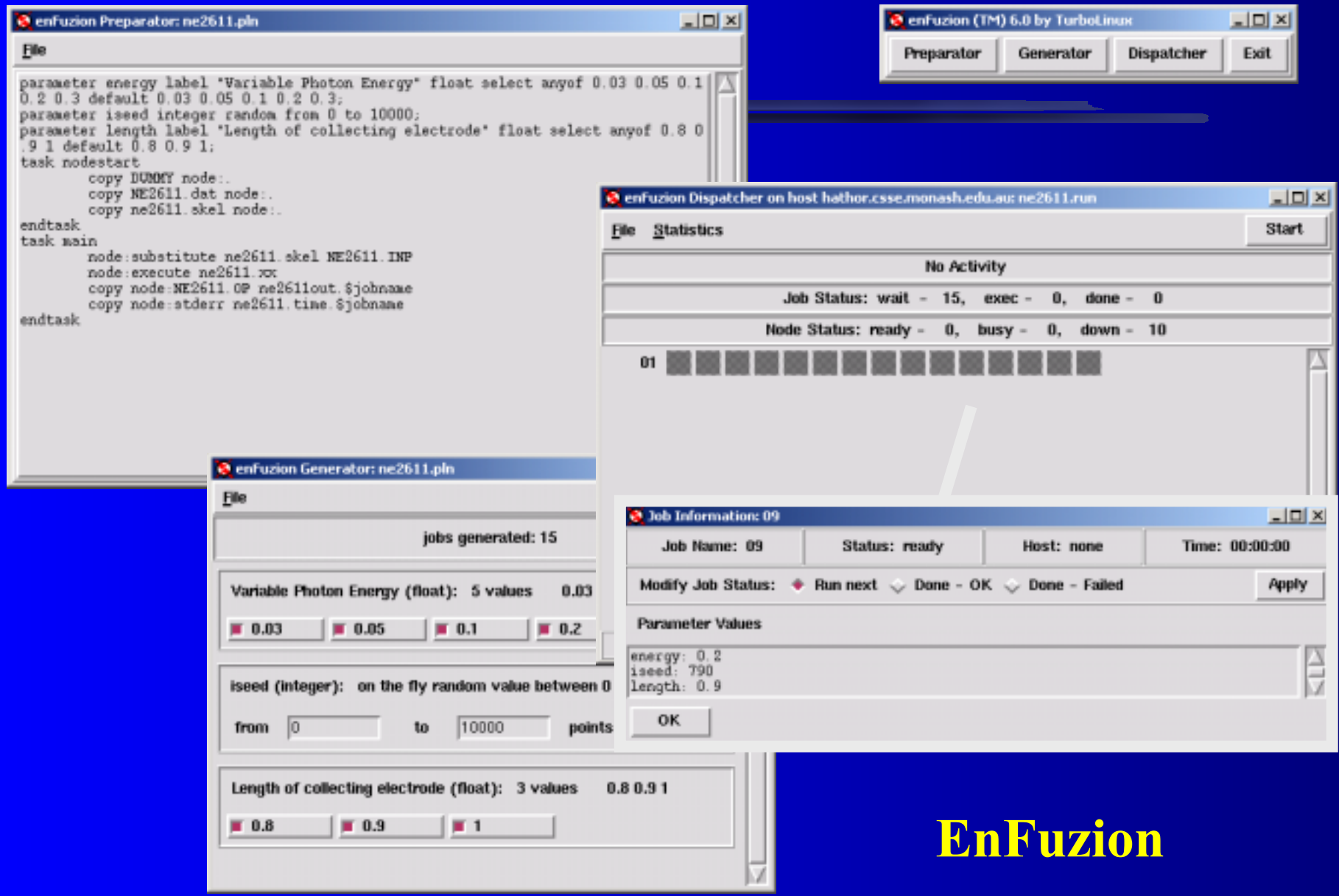






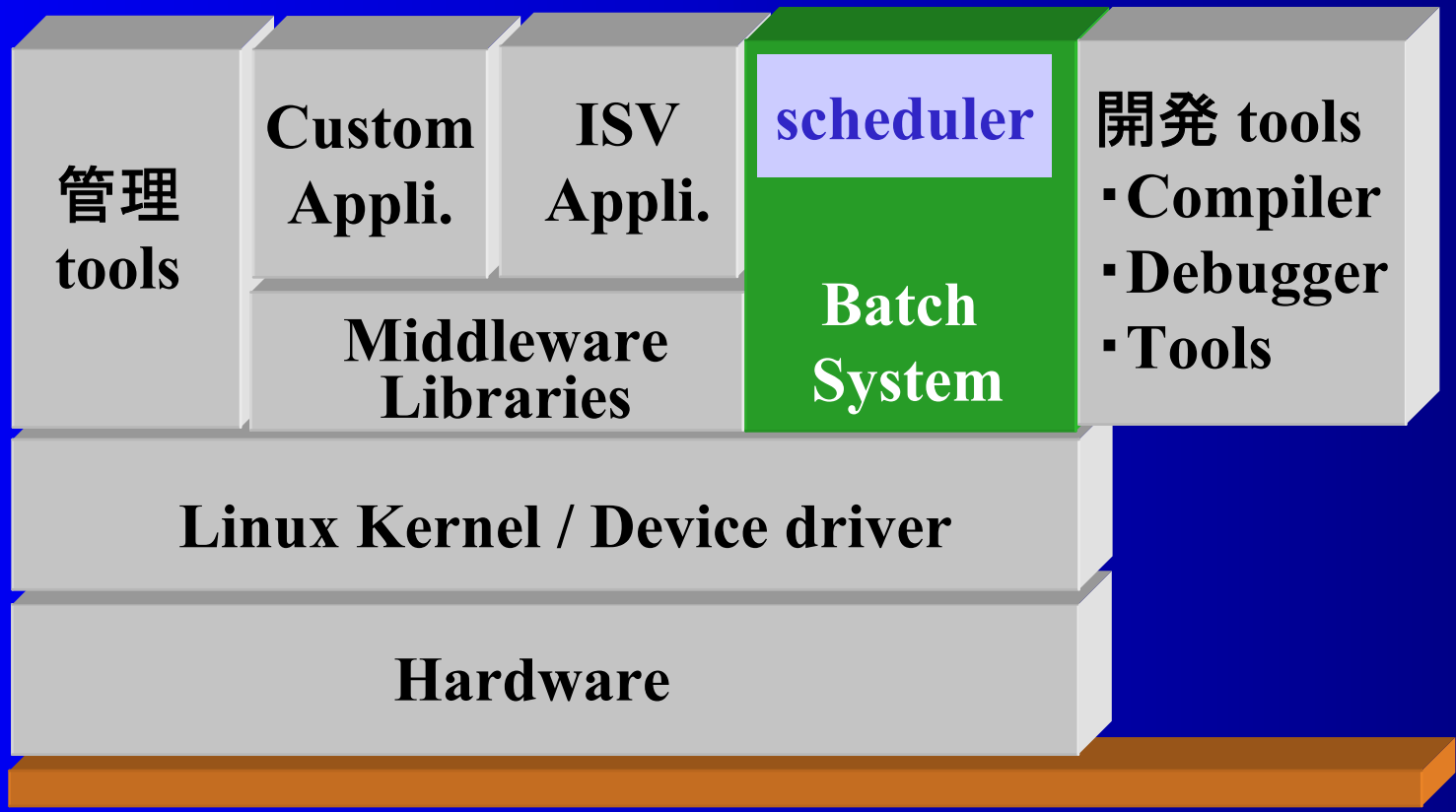
パラメーターの記述

- パラメータ自動生成
- 手続きのスクリプト化
- 自動ジョブ投入・集積
- ジョブ監視機構
- フェールオーバー機能



EnFuzion

HPC PC Cluster「バッチシステム」



バッチシステムとは

- スパコン等の主要コンポーネント
- 多数のジョブをジョブクラス、優先度に応じてスケジューリングし、ジョブ終了後にユーザ・ディレクトリに戻すシステム
- 商用パッケージ（多機能だが、ほとんどは必要ない）
  - LSF　(高い!!)
  - PBSpro
- Open Source
  - openPBS (http://www-unix.mcs.anl.gov/openpbs/)
  - SunGrid Engine (http://gridengine.sunsource.net/)

OpenPBS バッチシステム (admin)

- Configurations
- qmgr コマンドで設定
  - ノード設定ファイル
  - ジョブクラス属性設定
  - スケジューリング法設定ファイル
  - Prime time / non Prime time 設定ファイル
- Node properties設定
  - ノードと partition name との連結
  - 並列ジョブ用、シングルジョブ用 partition
  - ノードへのジョブ多重度の設定
- ジョブ・アカウンティング

OpenPBS バッチシステム (admin)

- Scheduling Method
  - By Queue, By Priority
  - Strict FIFO
  - FairShare-scheduling
- PBS+Maui scheduler (多数ユーザの場合有効)
  - http://supercluster.org/projects/maui/
  - 高度なスケジューリング機能
  - ジョブ Priority 計算
  - ジョブ統計情報集積機能

もはや、スパコン等のバッチ機能を凌ぐ!!

バッチジョブスクリプト例

```
#!/bin/bash
#PBS -N tinker
#PBS -l nodes=8
#PBS -j eo
cd $PBS_O_WORKDIR       ! job submitしたdirectory
                  ! ファイルのステージング
for n in 1 2 3 4 5 6 7 8
do
rcp dynamic.x             photon$n:/scratch/kato/dynamic.x
rcp fort10.org            photon$n:/scratch/kato/fort.10
rcp 2fsp.xyz              photon$n:/scratch/kato/2fsp.xyz
rcp 2fsp.xyz_2            photon$n:/scratch/kato/2fsp.xyz_2
rcp 2fsp.key              photon$n:/scratch/kato/2fsp.key
rcp ../params/amber.prm                 photon$n:/scratch/kato/amber.prm
done
cd /scratch/kato/           !   実行directoryへ移る
#
echo This jobs runs on the following processors:
echo `cat $PBS_NODEFILE`

# Define number of processors
NPROCS=`wc -l < $PBS_NODEFILE`   ! batch systemから適用可能なノードファイルをもらう
time mpirun -v -machinefile $PBS_NODEFILE -np $NPROCS ./dynamic.x
```

Job submit / Queue Classの状況

ジョブサブミット:

qsub –q small job.script

```
[kato@photon1 work]$ qstat -q
server: photon1
```

Queue	Memory	CPU Time	Walltime	Node	Run	Que	Lm	State
small	--	00:20:00	--	1	4	4	4	E R
long	--	12:00:00	--	1	1	0	7	E R
mpi4	--	24:00:00	--	4	1	0	2	E R
mpi8	--	24:00:00	--	8	0	1	1	E R
default	--	--	--	--	0	0	8	E R
					6	5		

Job Status (qstat -a)

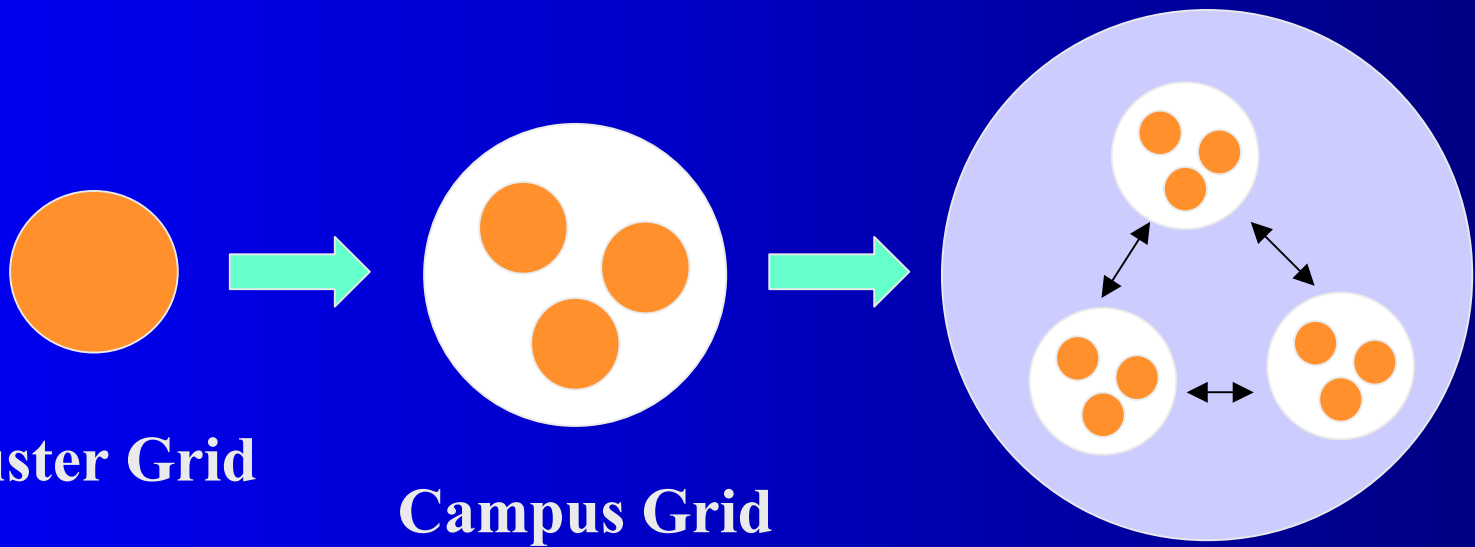
```
[root@photon1 /root]# qstat -a
photon1:                                                  Req'd   Req'd  Elaps
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
282.photon1     kato     small    nas         952     1   --    --    00:30 R 00:00
283.photon1     kato     small    nas        2191     1   --    --    00:30 R 00:00
284.photon1     kato     small    nas        2095     1   --    --    00:30 R 00:00
285.photon1     kato     small    nas        2080     1   --    --    00:30 R 00:00
286.photon1     kato     small    nas        3103     1   --    --    00:30 R 00:00
287.photon1     kato     small    nas        4578     1   --    --    00:30 R 00:00
288.photon1     kato     small    nas        1491     1   --    --    00:30 R 00:00
289.photon1     kato     small    nas        1519     1   --    --    00:30 R 00:00
290.photon1     kato     small    nas        4606     1   --    --    00:30 R 00:00
291.photon1     kato     small    nas        3142     1   --    --    00:30 R 00:00
292.photon1     kato     small    nas          --     1   --    --    00:30 Q   --
293.photon1     kato     small    nas          --     1   --    --    00:30 Q   --
294.photon1     kato     small    nas          --     1   --    --    00:30 Q   --
```

個別ジョブのステータス(qstat -f)

```
kato@photon1 work]$ qstat -f
Job Id: 235.photon1
    Job_Name = comm3
    Job_Owner = kato@photon1
    resources_used.cput = 00:02:19
    resources_used.mem = 18476kb
    resources_used.vmem = 28528kb
    resources_used.walltime = 00:02:32
    job_state = R
    queue = mpi8
    server = photon1
    Checkpoint = u
    ctime = Fri Oct 19 16:32:00 2001
    Error_Path = photon1:/home/kato/PBS/work/o
    exec_host = photon6/0+photon4/0+photon3/0+photon5/0+photon1/0+photon2/0+pho
       ton7/1+photon7/0
    mtime = Fri Oct 19 16:32:08 2001
    Output_Path = photon1:/home/kato/PBS/work/comm3.o235
```

Sun GridEngine

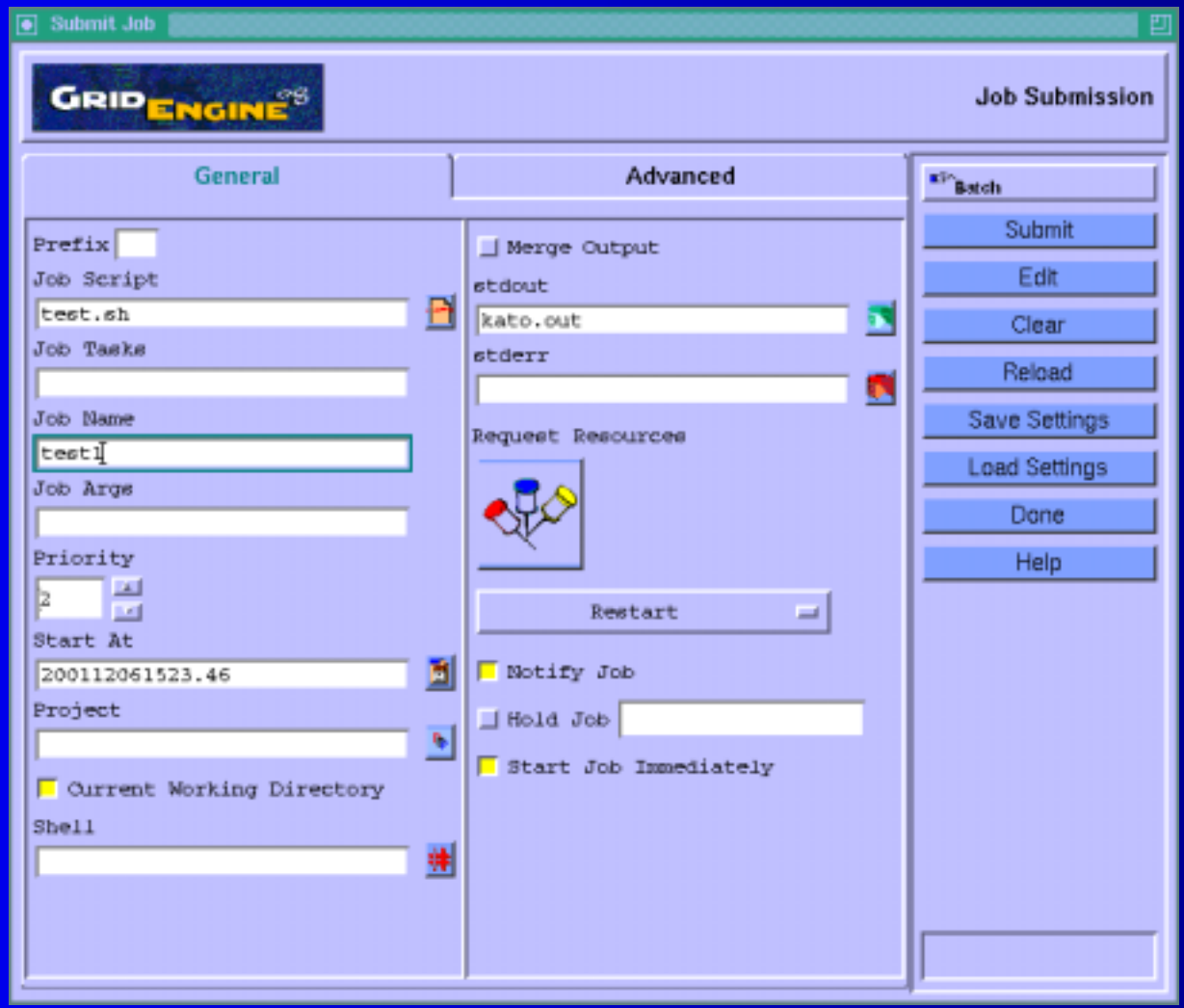
- バッチシステムのひとつ
- Gridとは、全てのリソースを隈無く使い切る用途
- Parametric Study + many many jobs



Sun GridEngineの特徴

- Daemonの フォールトトレラント機能
- User GUI の充実
- User job Checkpointing の機能
- ユーザコマンドは、他のものとほとんど同じ
  - qsub
  - qstat
  - Qrsh (Interactive)
  - Qacct (get accounting data)

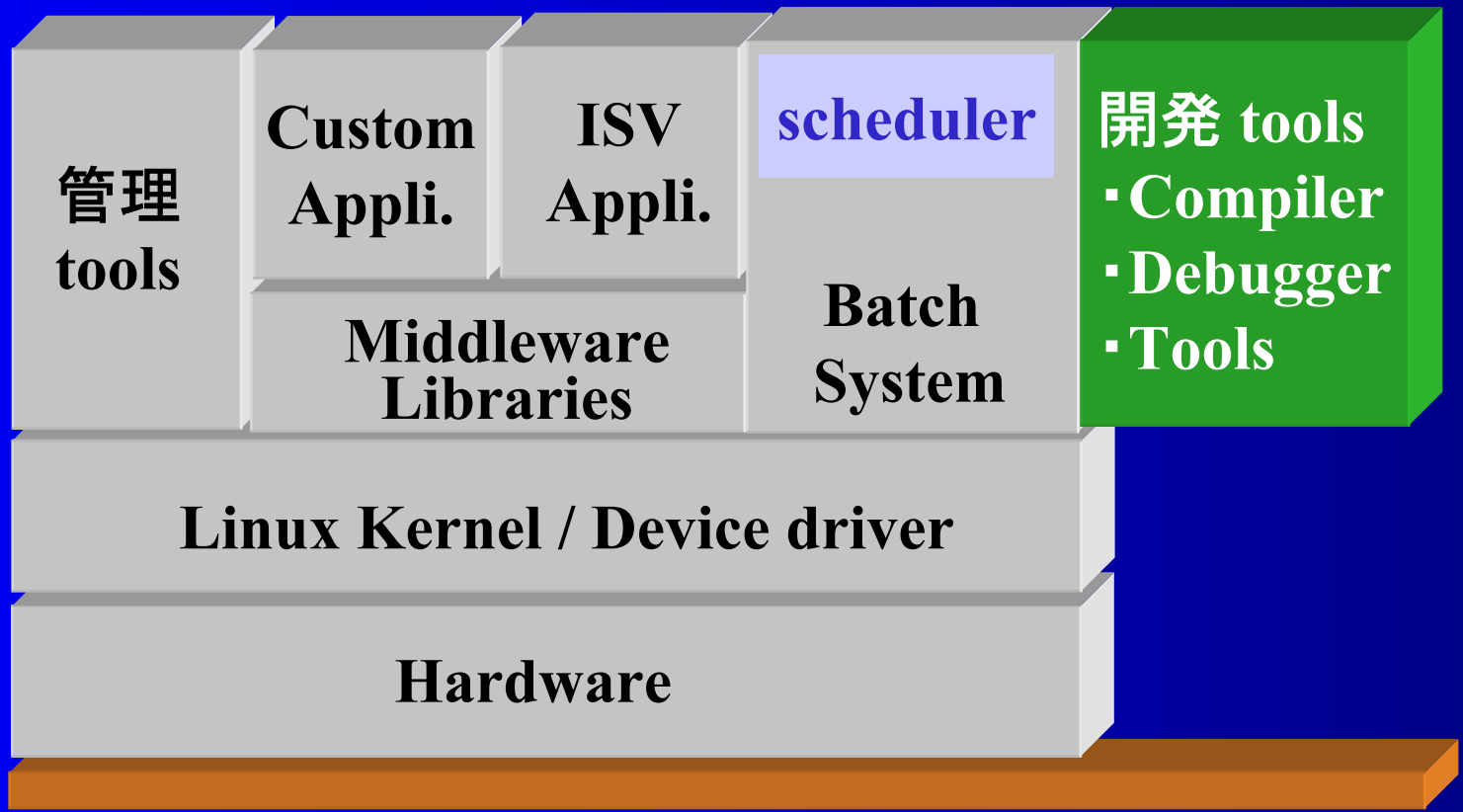
User GUI - qmon



Checkpointing / Restart 機能

- バッチシステムとの連携
- Process Migration,Crash recovery
- Load balancing , Roll Back
- EPCKPT
  - Eduardo Pinheiro Checkpoint Project
  - http://www.cs.rutgers.edu/~edpin/epckpt/

HPC PC Cluster「開発ツール」



クラスタシステムでのプログラム開発

- 1CPU / SMP並列高速性を追求する
  - コンパイラの機能性、性能（PGI compiler)
  - 性能評価ユーティリティ(Rabbit)
- 並列高速性を追求する
  - コンパイラの機能性、性能（PGI compiler/Profiler)
  - 開発ツール（CASEツール、相互参照、Call Tree)
  - 並列デバッガ（Etnus TotalView)
  - 数学ライブラリ
  - 性能評価ツール（Pallas VAMPIR/VAMPIRtrace)

HPC開発ツール Linux vs. スパコン

Linux Cluster 上では、free or 低価格で全てが揃っている

必須ツール	Linux Cluster	High-end Server Supercomputer
最適化コンパイラ	○ PGI, Intel 他	○ 各vendor
性能プロファイラ(GUI)	○ PGI pgprof	○ 各vendor
クロスリファレンス	○ GXCHK 他	○ CASE tools
H/W 性能モニタ	○ rabbit	△ 一部のみ
並列デバッガ(GUI)	○ TotalView	○ TotalView
MPI性能解析ツール	○ VAMPIR	○ VAMPIR
性能監視モニター	○ Linux tools	○ 各vendor

PGI compiler for IA-32

- Fortran / C / C++ / HPF パッケージ
- SMP 用の自動並列化機能搭載、業界標準 OpenMP 対応のコンパイラ
- Pentium III / 4 の SIMD SSE / SSE2 に対応
- メモリプリフェッチ機能、多種の最適化オプション
- キャッシュ最適化、ベクトル（配列）の最適化を自動化（ -fast -Mvect ）
- 古いアプリケーションの互換性を維持するために IBM/DEC/Cray の Fortran 拡張を実装
- Byte-swapping I/O をサポートし、他の RISC/UNIX とのエンディアン互換性
- Profiler /Debugger　付属

なぜ、GNU complerを使用しないか

- プログラム高速性を引き出すため
- GNU compiler では、
  - No Pipelining
  - No Optimization for Floating Point Division
  - No Vectorization
  - No Square Root Instruction
  - No Loop Interchange Capability
  - No Data Prefetching Capability （もう直）
  - No Report or Detailed Listing
  - Information for Performance

PGI compiler vs. GNU compiler

PGI compiler の最適化機能は、スパコンのコンパイラ機能と同等レベル

言語	アプリ	Linux	GNU (g77)	PGI (pgf77)	Ratio
Fortran (1)	分子動力学	2.4.2	1296秒	899秒	1.44
C (2)	コンボリューション	2.2.X	67.7秒	58.6秒	1.15

(1) Pentium4 1.8GHz　RDRAM 1.5GB
(2) Pentium3　866MHz SDRAM 256MB

pgf77/pgf90(Fortran) + data prefetch

- Pentium データプリフェッチ機能の活用
- pgf90 -fast -Mvect=prefetch src.f
- NAS Parallel Benchmark 2 serial

Linux 2.4.7 Pentium 4 1.8GHz ,1.5GB

NAS serial	Prefetch なし	Prefetch あり
FT (FFT) 256x256x128	221.1 MFLOPS	243.4 MFLOPS (x1.10)
LU (SOR) 64x 64x 64	310.4 MFLOPS	333.0 MFLOPS (x1.07)

Pgcc + OpenMPによるSMP並列化

pgcc -c -mp conv.c　(-mp:並列化オプション)

```
#include      <omp.h>
void    convo(p1, ja, jb, x1, x2, fco)
float   p1, ja, jb, fco[N][N];
int     *x1, *x2;
{
#pragma omp parallel private(ix,iy,jj,r,ftmp)
                     shared(p1,ja,jb,x1,x2,fi)
#pragma omp for schedule(guided,20)
      for(iy = 0; iy < N; iy++){
            ftmp = p1 + iy * jb + ja * x1[iy];
            .....(コンボリューション処理).....
      }
}
```

並列化のためのpragma

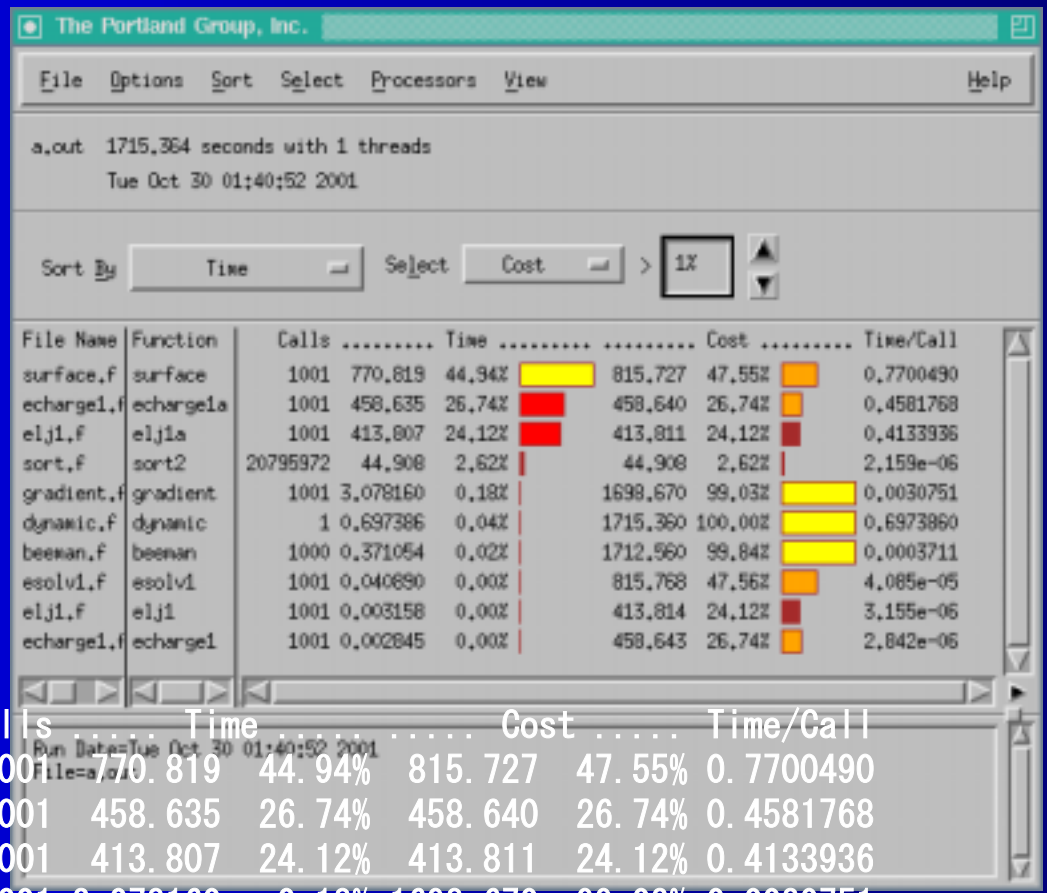
コンボリューションのSMP並列の例

Linux 2.4.2	Pentium3 (Dual CPU)	Pentium3	Pentium4 (1cpu)
Clock (MHz)	800	800	1300
Memory	PC100	PC100	PC800
Data Size	小	大（x3)	小
PGI as is (秒)	63	—	24.5
PGI +gcc-assembler	29.0 (x2.17)	168 (noSSE) 144 (SSE)	12.5 (x1.96)
1thread	29.8	—	—
2thread(OpenMP)	17.9 (x1.66)	—	—

Profiler – pgprof (1cpu) の例

コンパイルオプション
・関数レベル
pgf90 –Mprof=func
・ラインレベル
pgf90 –Mprof=lines

性能プロファイラ
pgprof pgprof.out

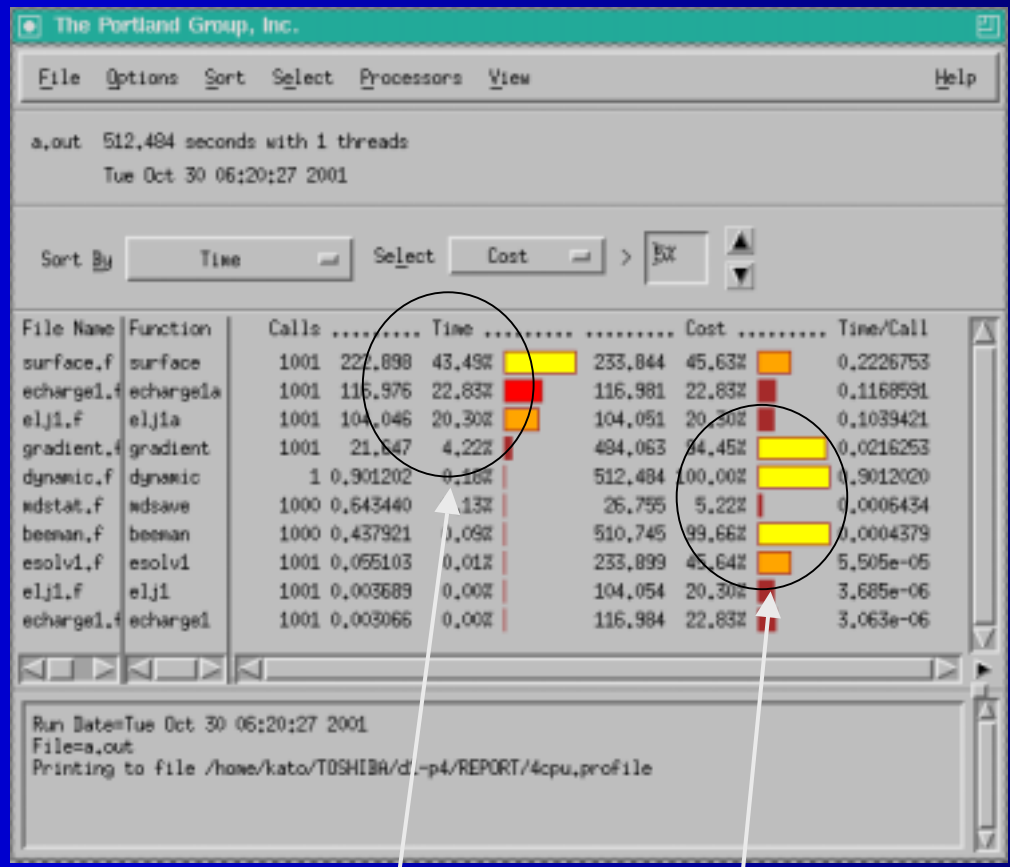


File Name	Function	Calls	Time		Cost		Time/Call
surface.f	surface	1001	770.819	44.94%	815.727	47.55%	0.7700490
echarge1.f	echarge1a	1001	458.635	26.74%	458.640	26.74%	0.4581768
elj1.f	elj1a	1001	413.807	24.12%	413.811	24.12%	0.4133936
gradient.f	gradient	1001	3.078160	0.18%	1698.670	99.03%	0.0030751
dynamic.f	dynamic	1	0.697386	0.04%	1715.360	100.00%	0.6973860
beeman.f	beeman	1000	0.371054	0.02%	1712.560	99.84%	0.0003711

Profiler – pgprof (4cpu) の例

並列化におけるルーチン毎の性能向上率の把握

Routine	1cpu	4cpu
surface	770.819	222.898
echarge1a	458.635	116.976
elj1a	413.807	104.046



Pentium3(866MHz) vs.Pentium4(1.8GHz)

プロセッサの違いよる性能向上の比率(クロック比:1 : 2.07)

P	File Name	Function	..... Time		..... Cost		Time/Call
P3	echarge1.f	echarge1a	1078.320	35.31%	1078.320	35.31%	1.0772428
P4	echarge1.f	echarge1a	458.635	26.74%	458.640	26.74%	0.4581768

x 2.35

P3	surface.f	surface	996.890	32.64%	1053.940	34.51%	0.9958941
P4	surface.f	surface	770.819	44.94%	815.727	47.55%	0.7700490

x 1.29

遅い？=> 分岐命令多し

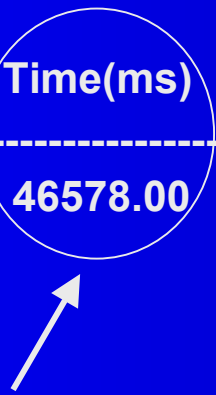
P3	elj1.f	elj1a	876.658	28.71%	876.665	28.71%	0.8757822
P4	elj1.f	elj1a	413.807	24.12%	413.811	24.12%	0.4133936

x 2.12

Profiler – pgprof (Line mode)

- Statementレベルの Hot spotを探す
- pgf90 -Mprof=lines …. ; pgprof

Visits	Time(ms)	Line#	
300455355	46578.00	156	do i = 1, n
		157	if (i .eq. ir) goto 30
		158	rplus = rr + r(i)
		159	tx = x(i) - xr
		160	if (abs(tx) .ge. rplus) goto 30
		161	ty = y(i) - yr
		162	if (abs(ty) .ge. rplus) goto 30
		163	tz = z(i) - zr
		164	if (abs(tz) .ge. rplus) goto 30
			…….



大コスト部分
コード特性を
評価する

Cross Reference (GXCHK free version)

*plus*FORT Version 6 (free version for Linux)

http://www.polyhedron.com/pf/plusfort.html

```
0001: MAIN
0002: |--AM1
0003: | |--MOLDAT
0004: | | |--GMETRY
0005: | | | '--GEOUT
0006: | | | '--XYZINT
0007: | | | |--BANGLE
0008: | | | |--DIHED
0009: | | | | '--DANG
0010: | | | '-XYZGEO
0011: | | | |--BANGLE
0012: | | | '---DIHED > 0008
0013: | | |--READA
0014: | | | '--DIGIT
0015: | | '--VECPRT
0016: | |--READA > 0013
0017: | '--UPDATE
0018: |-CMNAM <- unresolved
0019: |-COMPFG
```

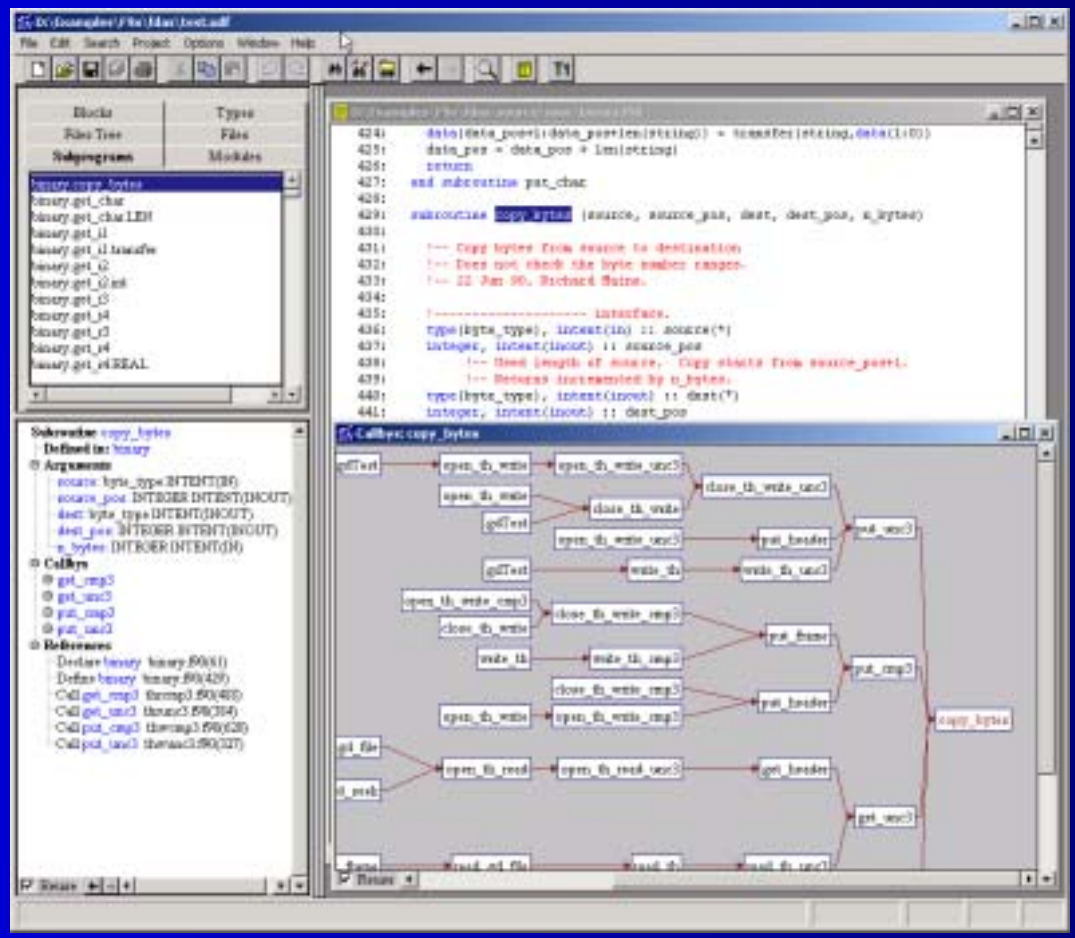
```
ANEXT ..... changed by SEARCH
                      used by POWSQ SEARCH
AQM  .....  changed by AA0001 CALPAR
                      used by MOLDAT
AQPM3 ..... changed by AA0001
                      used by MOLDAT

COMMON variable name used in other contexts  ALPHA
 ------------------------------------------------------------------------
 | ALPHA    R*8  d.arg  variable    pos   1 in ASET  in
                                      overlap.f,0000302,0000314
 | ALPHA    R*8  static variable    DEFINED in DSTMAT   in
                                      embed.f,0000956,0001343
 | ALPHA    R*8  d.arg  variable    pos   1 in EWALDCOF in
                                      kewald.f,0000176,0000214
```

Reverse engineering, cross reference tool

- ソース解析
- クロスリファレンス
- Call Tree
- 商用ツール（低価格）

“understand”
for Fortran, C, C++



http://www.scitools.com/fortran_tools.html

実効性能・H/Wカウンタ特性を見る

- Rabbit（**Open Source)**
  A Performance Counters Library
  for Intel/AMD Processors and Linux
  - **http://www.scl.ameslab.gov/Projects/Rabbit/**
- Pentiumプロセッサが備えるH/W performance Counter の値をサンプリングするユーティリティ
  - 実効性能(**MFLOPS**)を知りたい
  - キャッシュヒット率が知りたい
- 性能最適化の度合いを知る上で重要な機能
- Pentium 3で動作

Rabbit の使用法

- 使用法

  rabbit –g {0 - 9} a.out (wrapperとして機能)

<-g 0の場合>

Event	Events	Events/sec
0x43 67 data_mem_refs	19267722998	267718057.88
0x10 16 fp_comp_ops_exe	8997782344	125031419.51

実効性能　　：　125MFLOPS

メモリ参照数　：　267Mevent/s

Computational Intensity ：125/267= 0.47

On Pentium3 866MHz

rabbit –g 9 a.out (all couters report)

イベント		単位	サンプル数	
番号	イベント名			
0x10 16	fp_comp_ops_exe	FLOP/sec	125,031,419	(a)
	fp_comp_ops_exe	FLOP-Event	8,997,782,344	(b)
0x43 67	data_mem_r	Event/sec	267,718,057	©
	data_mem_r	Event	19,267,722,998	(d)
0x45 69	dcu_lines_in	Event	392,299,537	(e)
0x46 70	dcu_m_lines_in	Event	76,293,902	
0x47 71	dcu_m_lines_out	Event	87,213,638	(f)
0x28 40	l2_ifetch	Event	11,023,714	
0x29 41	l2_ld	Event	399,529,490	
0x2a 42	l2_st	Event	10,251,997	
0x24 36	l2_lin	Event	71,699,451	(g)
0x26 38	l2_lines_out	Event	71,699,961	
0x25 37	l2_m_lines_inm	Event	53,919,242	
0x27 39	l2_m_lines_outm	Event	53,880,727	(h)
0x79 121	cpu_clk_unhalted	Cycle	62,440,168,177	(i)
0xa2 162	resource_stalls	Cycle	40,562,402,489	(j)
0xc4 196	br_inst_retired	Event/sec	40,855,706	(k)
0xc5 197	br_miss_pred_retired	Event/sec	1,059,120	(l)
0x79 121	cpu_clk_unhalted	Cycle	62,440,168,177	(m)
0xc0 192	inst_retired	Event	30,609,661,800	(n)

次頁

カウンタ指標から各種性能指標計算

ある分子動力学コードの特性

実効性能	125	MFLOPS	(a)
平均データ転送率	2,142	MB/sec	(c)x8byte
L1キャッシュヒット率	90%		(d-(e+f)*4)/d)
L2キャッシュヒット率	97%	高い	(d-(g+h)*4)/d)
理論計算密度	0.47		(a/d)
実効計算密度	17.91		(b/d*L2-HITratio)
リソーススストール比	0.39		(j/i)
分岐予測ミス率	2.59%	ミス多い	(l/k)
IPC(Inst. per cycle)	0.49		(n/m)

	Pentium3 868MHz	Pentium4 1.8GHz
Elapsed time	3017秒	1710秒
MFLOPS	125(実測)	220(計算)

並列プログラム開発環境での問題

- 見たいものが見えない
  - 並列動作時の問題箇所の特定が難しい
  - MPI並列開発では、演算・通信の状況が見えない

並列プログラミング環境の導入必須

業界標準のデバッガ&ツール

マルチプロセスデバッガ
: **TotalView**

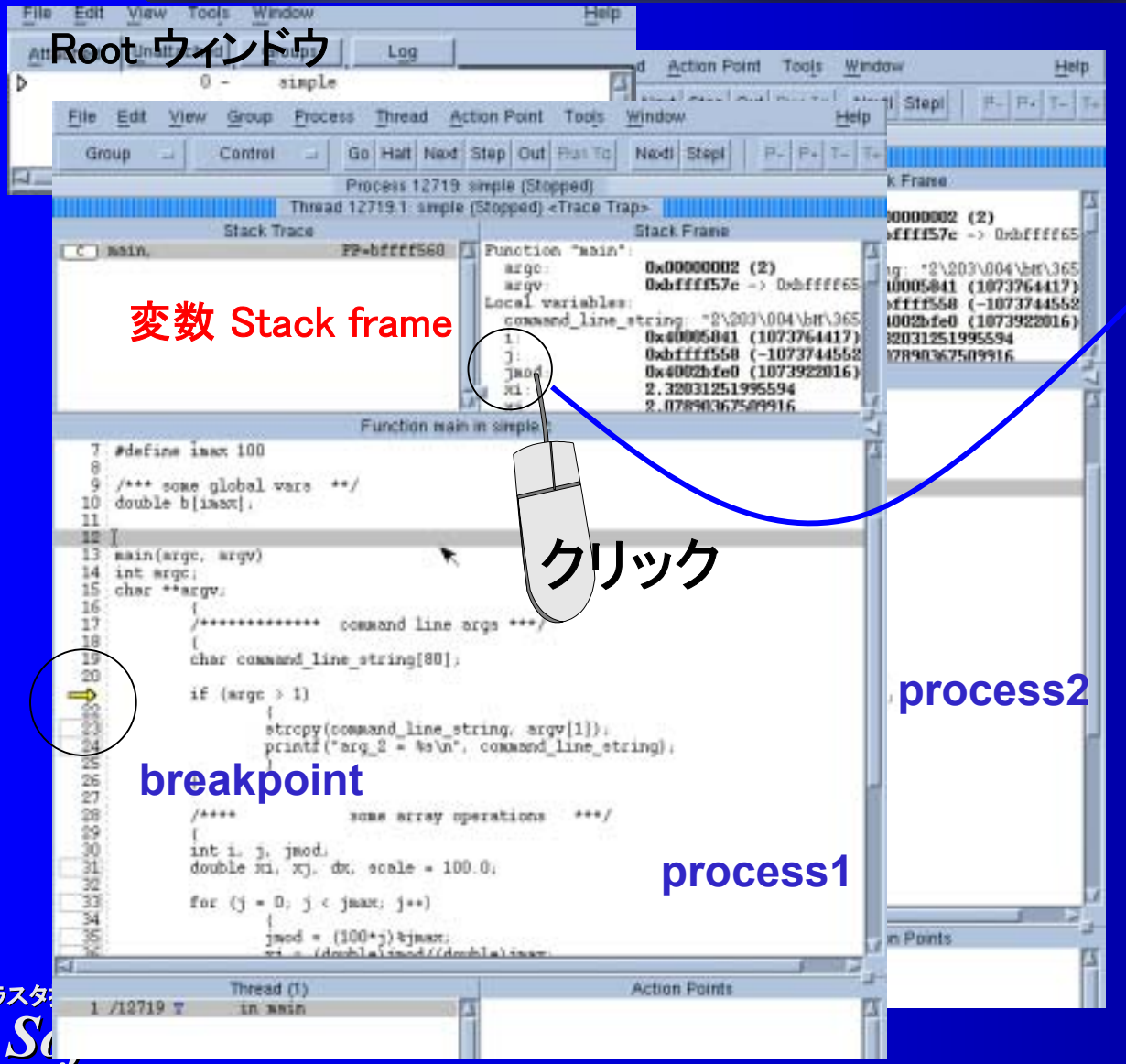
MPI プログラムの性能解析
: **VAMPIR**

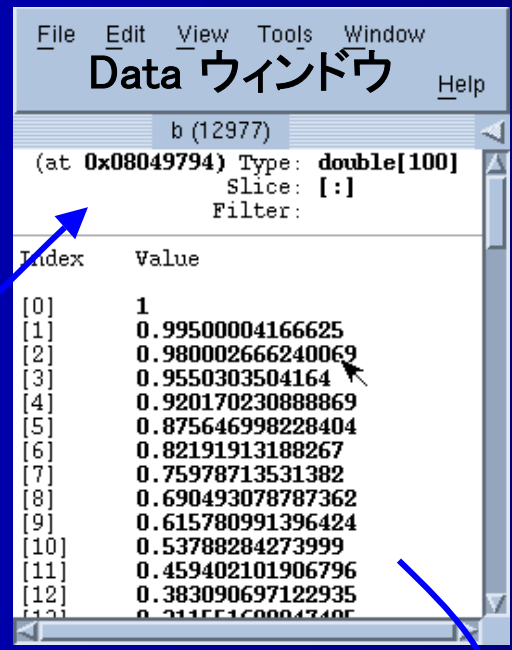
並列デバッガ Etnus TotalView

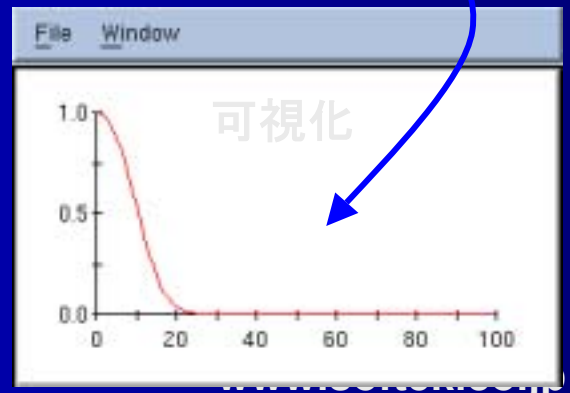
- 業界標準並列デバッガ （スパコンからLinux Clusterまで）
- 操作性に優れたGUIにより効率よく使用できる
  - マウスのボタン操作で主要なコマンドを実行可能
- 多様な分散、並列プログラミングモデルをサポート
- 巨大で、複雑なプログラムもデバッグ可能
- 様々な言語、プラットフォーム、アーキテクチャで利用可能
  - Multi-thread programming
  - MPI programming
  - OpenMP programming

Totalview の操作

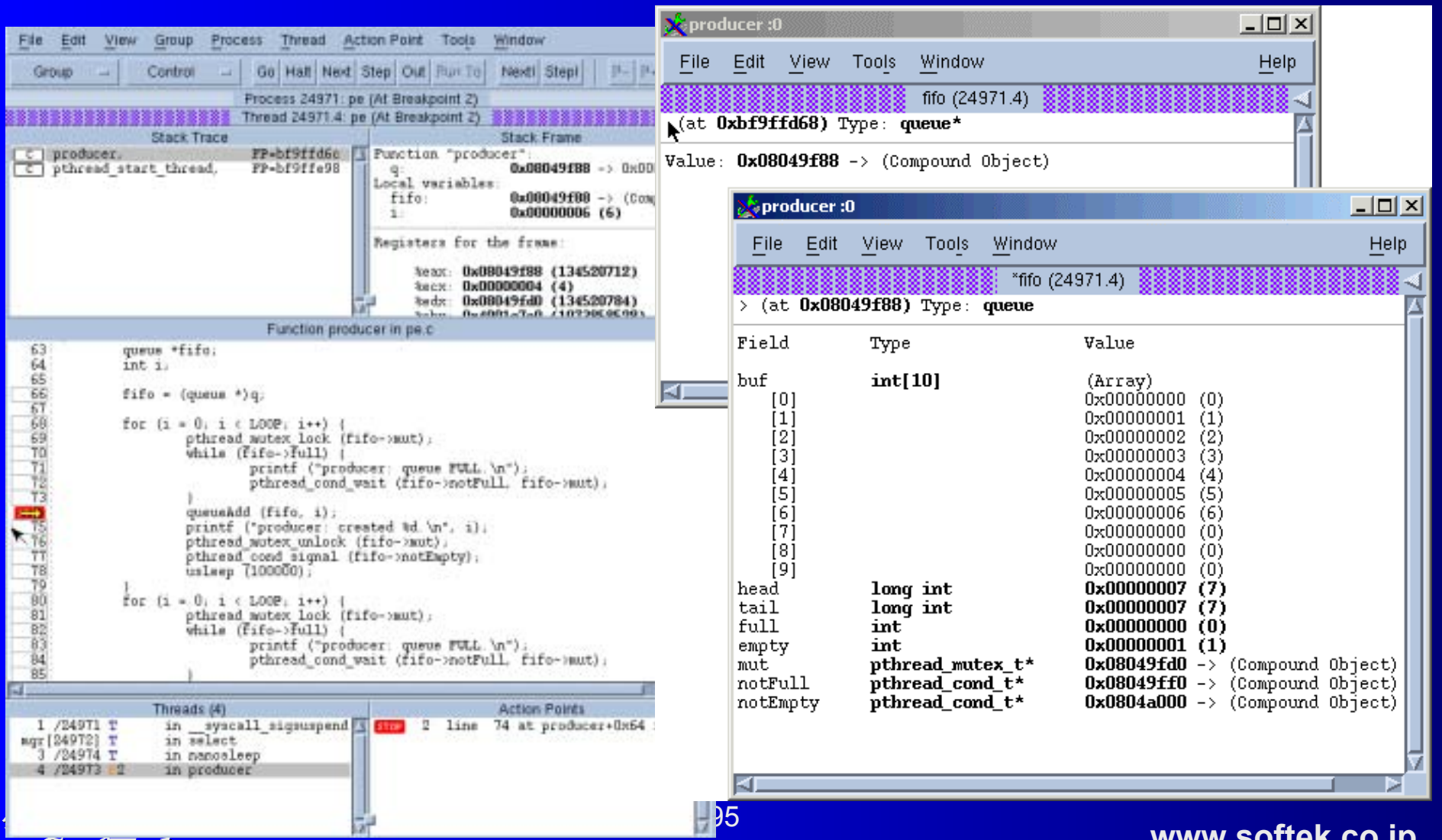
ダイブによりプロセスの
内部を表示可能







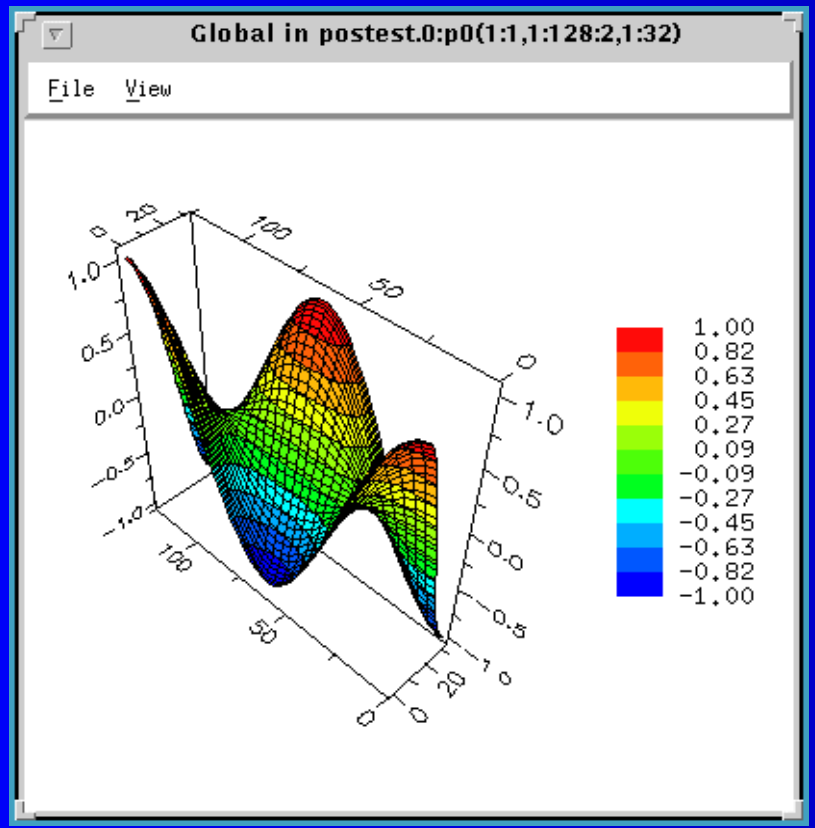
Multi-Threaded コード



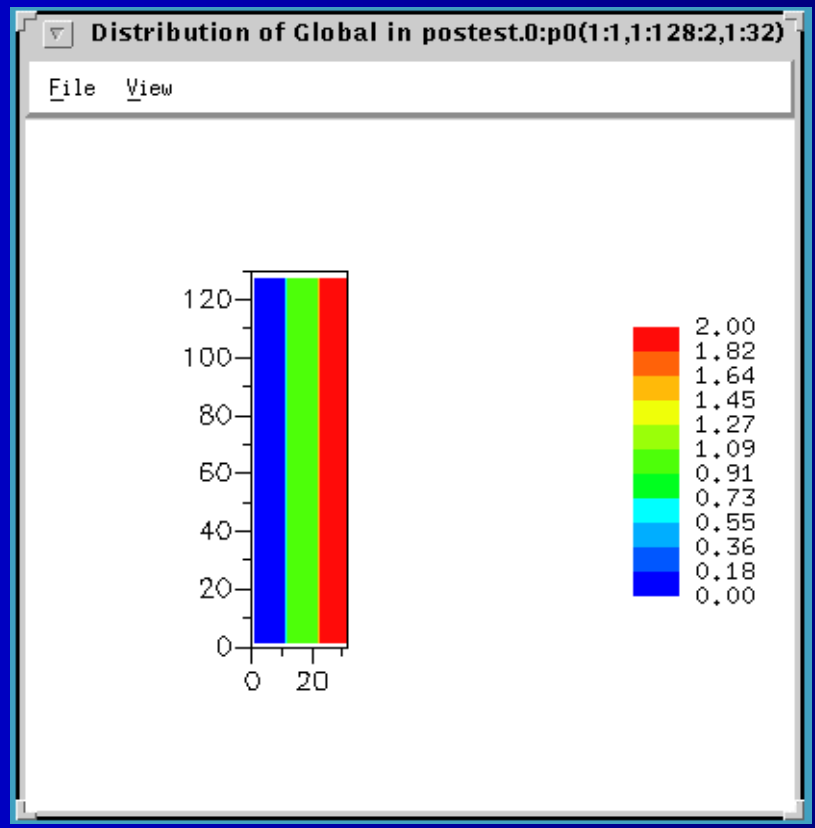
SofTek

TotalView データの可視化

分散配列の可視化



Visualize array distribution

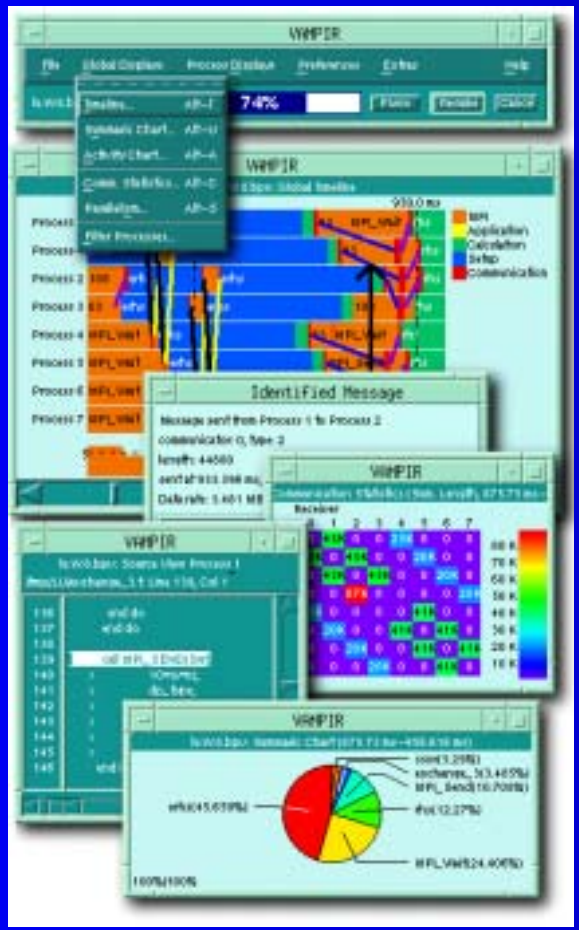


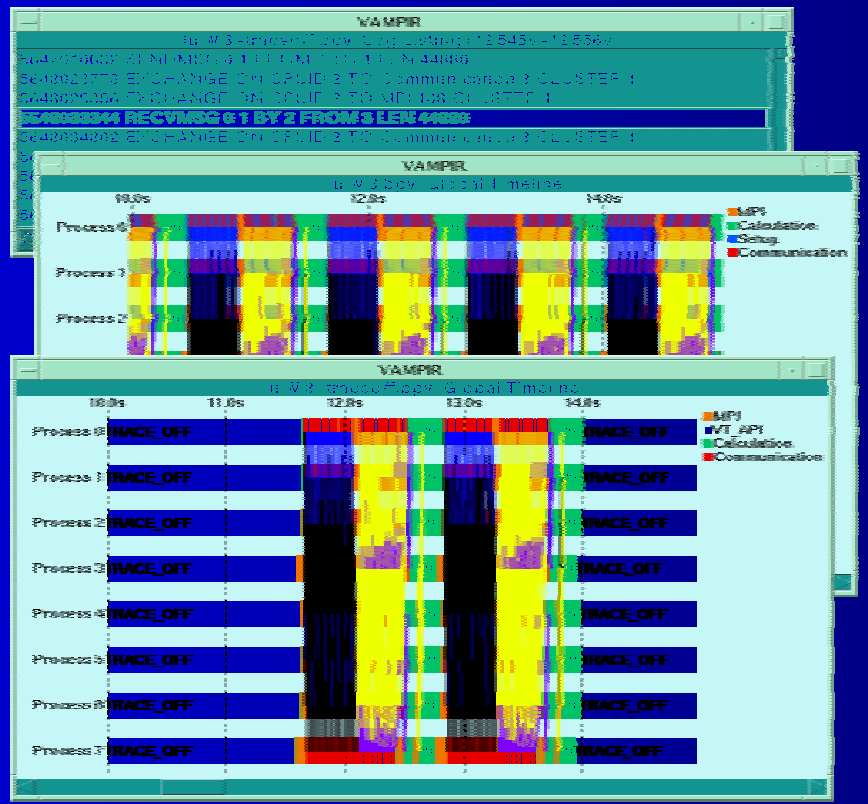
VAMPIRの特徴

- MPI (及びアプリケーションイベント)のオフラインイントレース解析
- VAMPIRtrace ツールによるトレース生成
- 扱いやすいユーザインタフェース
- スケーラブル（時間とプロセッサ空間）
- 秀逸なズーミング、フィルタリング機能
- 高性能グラフィックス
- MPIとアプリケーションイベントの表示、解析:
  - **MPI** ルーチン
  - **1対1** 、集団通信

VAMPIR GUI (1)

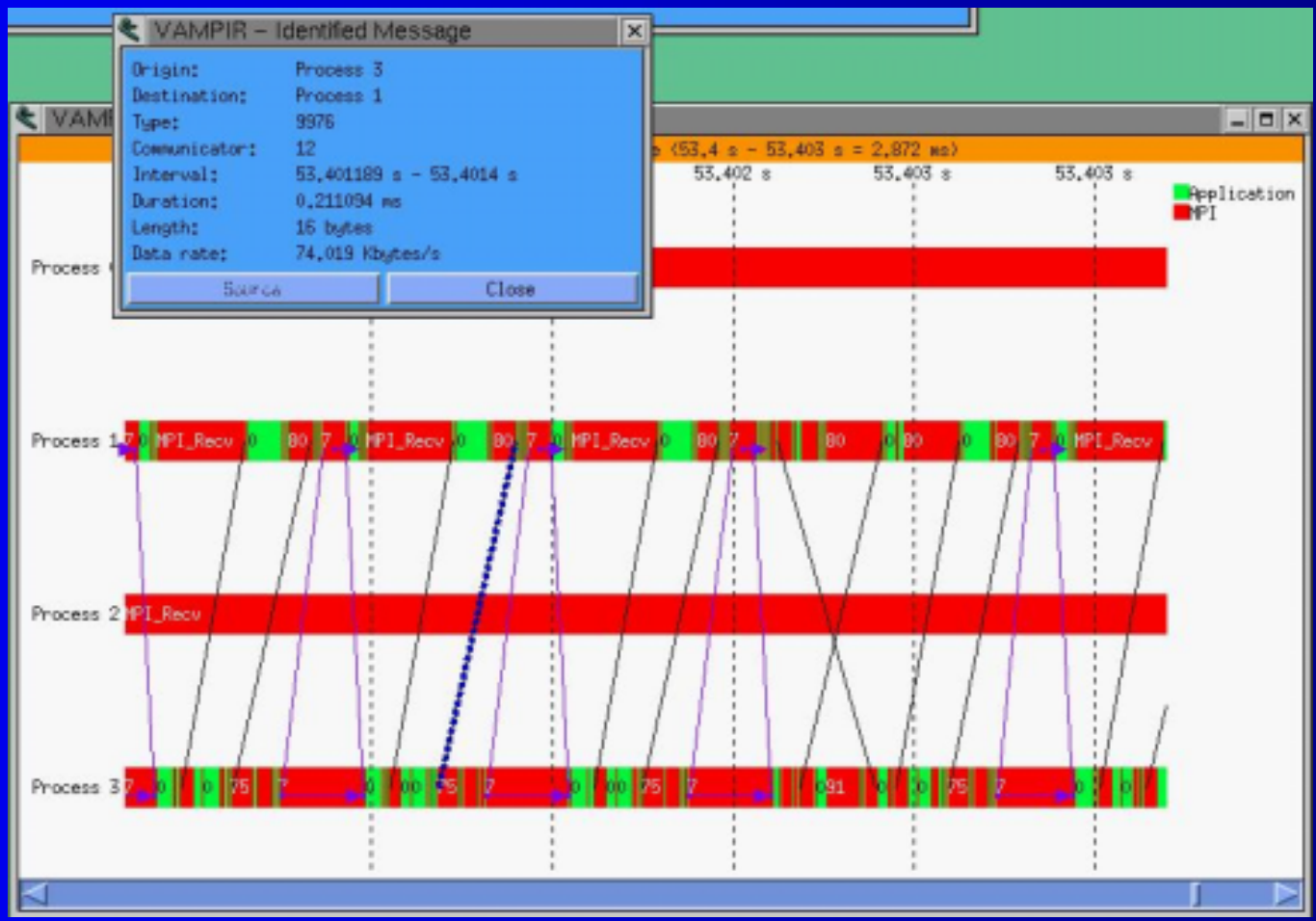
MPIの動作挙動が見える





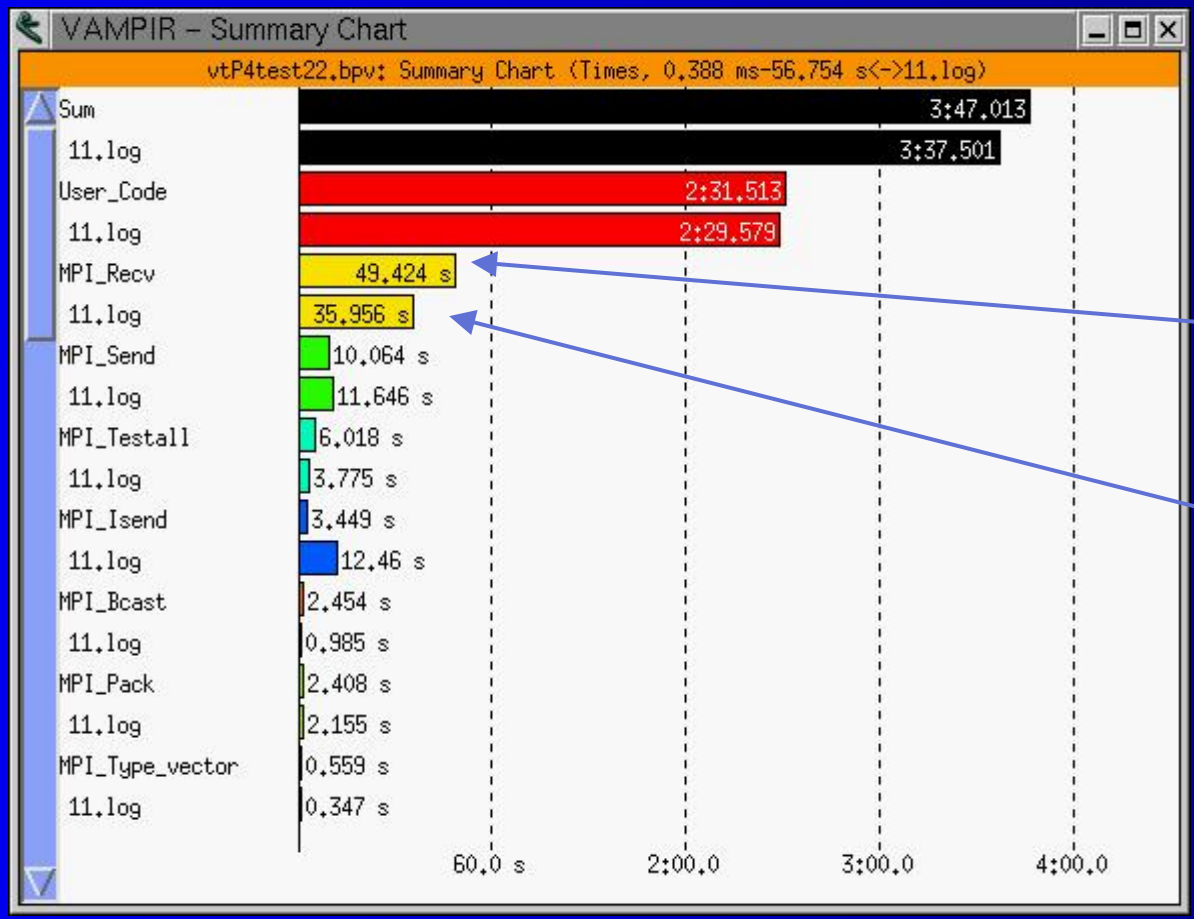
VAMPIR GUI (2)

MPICH(p4)による通信状況のスナップショット



VAMPIR GUI (3)

通信のサマリチャート(二つの結果の同定例)



MPICH

MPI/GAMMA

3D-FFT計算における通信特性

TCP/IP / Fast Ethernet

GM / Myrinet2000

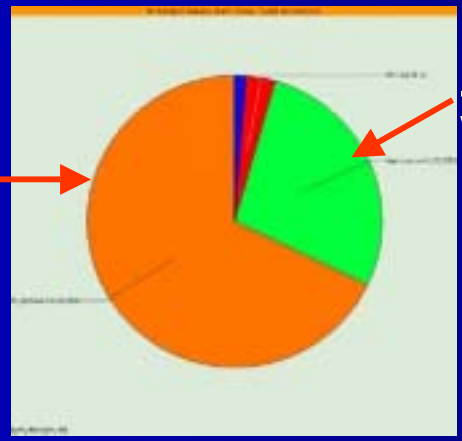
全計算時間100%

演算時間

30%

70%

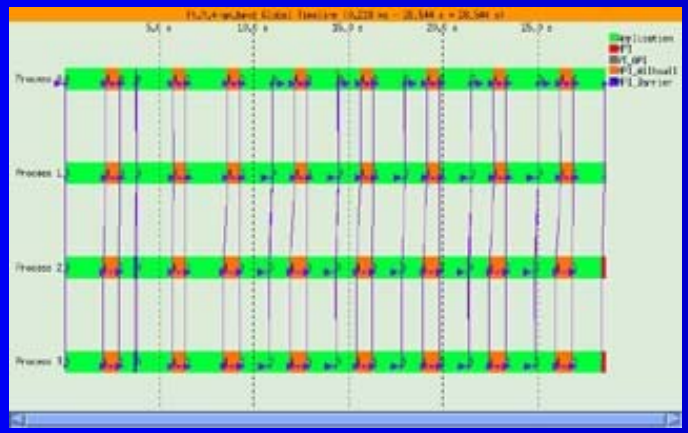
通信時間



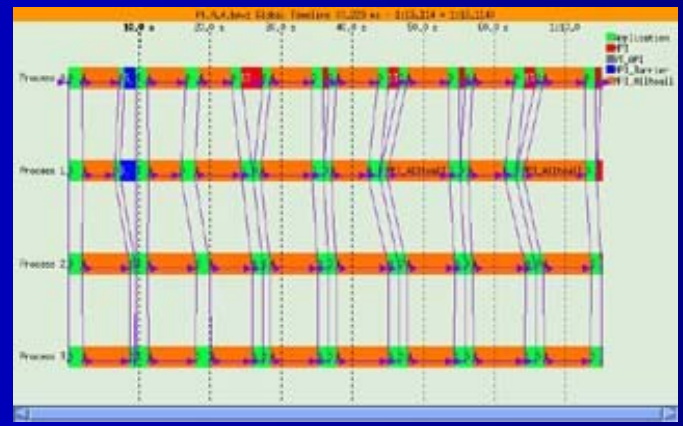
通信時間

時間トレース

通信パターンの違い（4 並列の場合）



4.67MB/sec



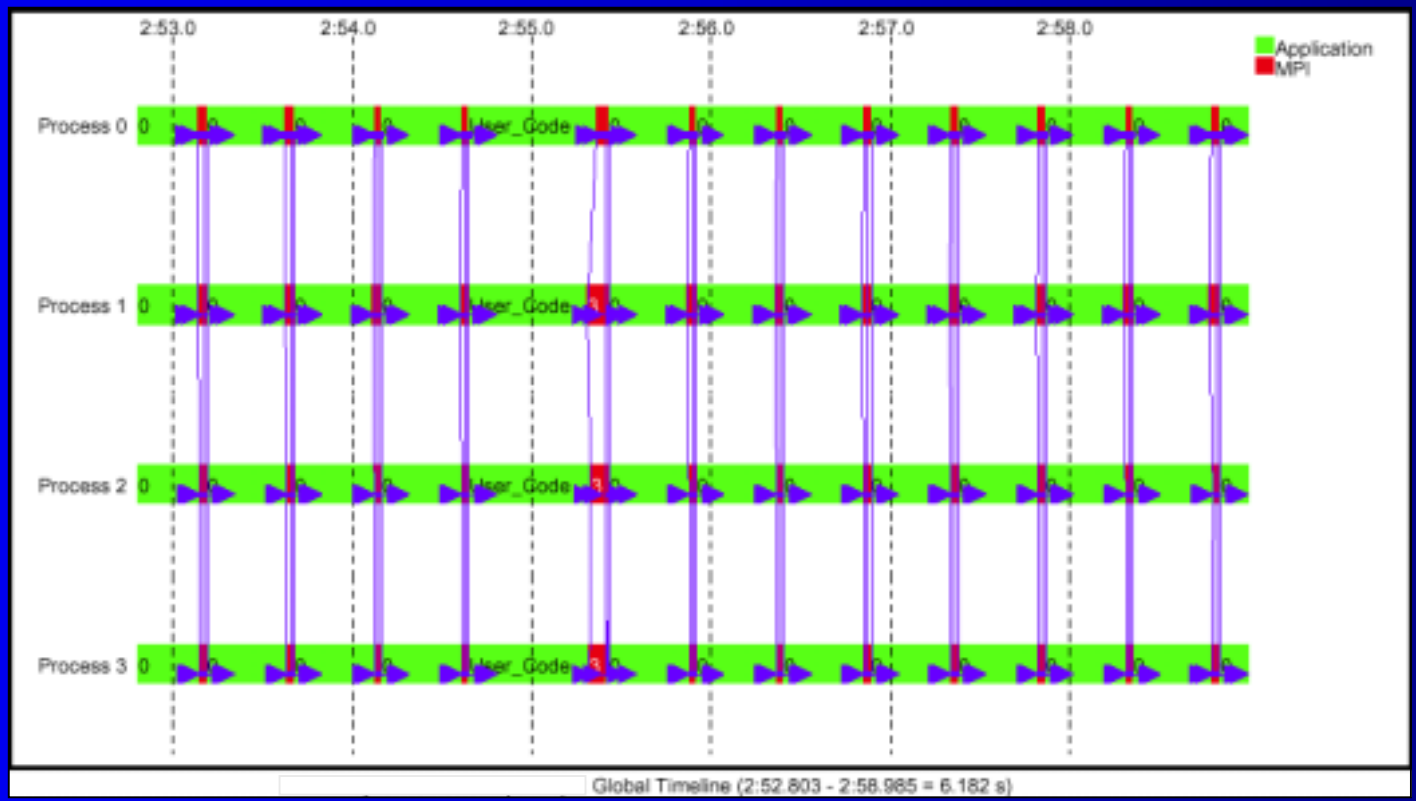
43.1MB/sec

VAMPIRによる性能評価予測（1）

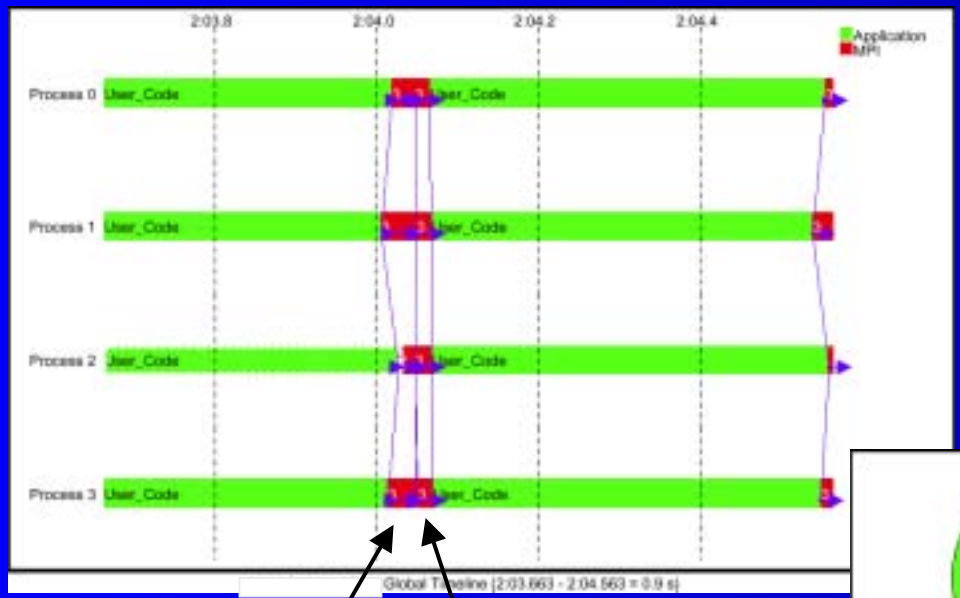
題材：Chemistry 系 MPI program (Fast Ethernet)

通信のパターンをマクロで押さえる

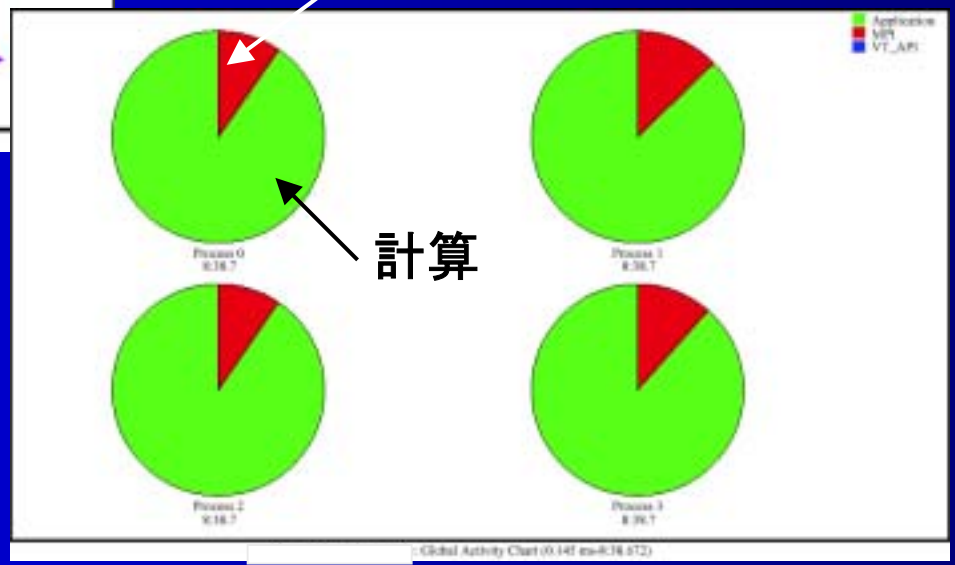
CPU



VAMPIRによる性能評価予測(2)



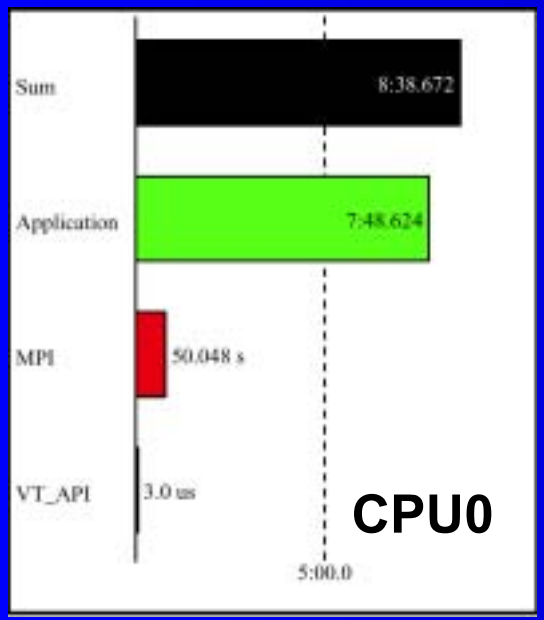
通信

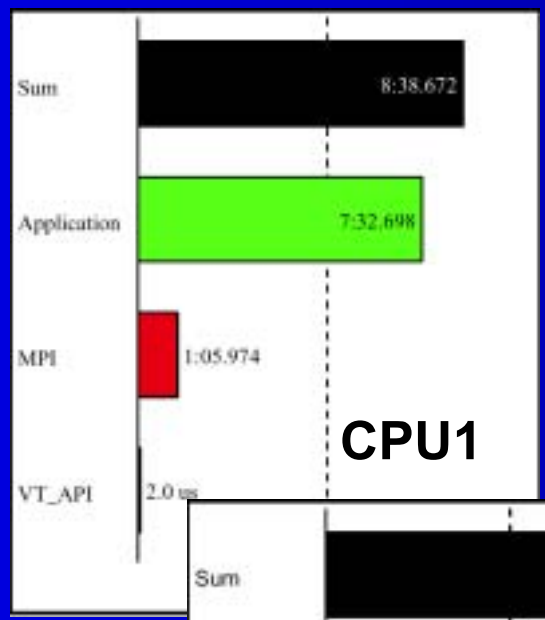


計算

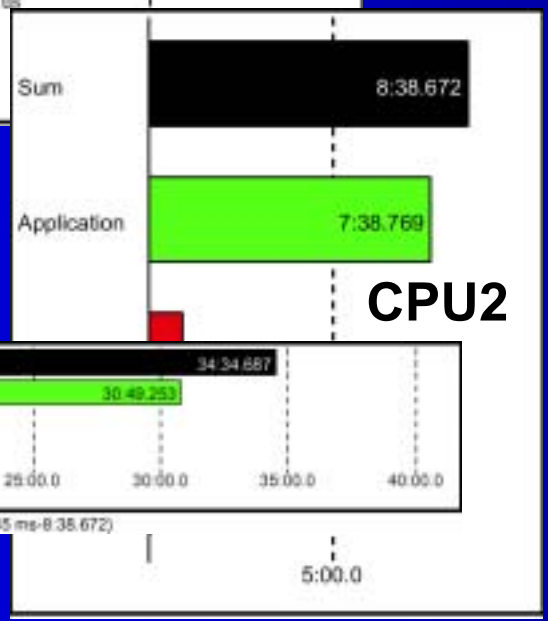
二つの通信が行われている

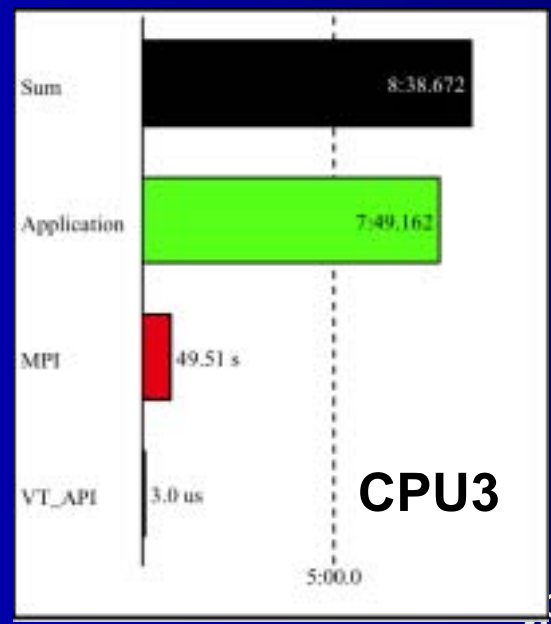
VAMPIRによる性能評価予測(3)





各プロセッサの計算粒度と
通信時間のバランス





全プロセッサ・サマリ

VAMPIRによる性能評価予測(4)

・通信の種類・パターンを押さえる(1:1通信 or 集合通信)
・最大・平均・最小メッセージ通信量
・最大・平均・最小通信速度
・個々の通信モデルの定量評価(時間・メッセージ量)

集合通信の最大転送レート

VAMPIRによる性能評価予測(5)

通信媒体: Fast Ethernet使用時
（Vampirによる実測時間による抽出）

- アプリケーション実行ステップ数　：　1001ステップ
- ステップ当たり実行時間　：　0.43～0.44秒
- 通信関数(step 当たり)　：　MPI_Allreduce 2 回
- 通信総回数、総転送量　：　2002回、108MB x 2 /CPU
- 通信サイズ　：　1回目:63.5KB
　：　2回目:47.68KB
- 通信時間　：　1回目:21～31msec
（純粋な通信時間＋待ち時間）　：　2回目:18～19msec

通信媒体の違いによる性能差

Fast Ethernet (100Mbps) vs. Myrinet (1.25Gbps)

Fast Ethernet

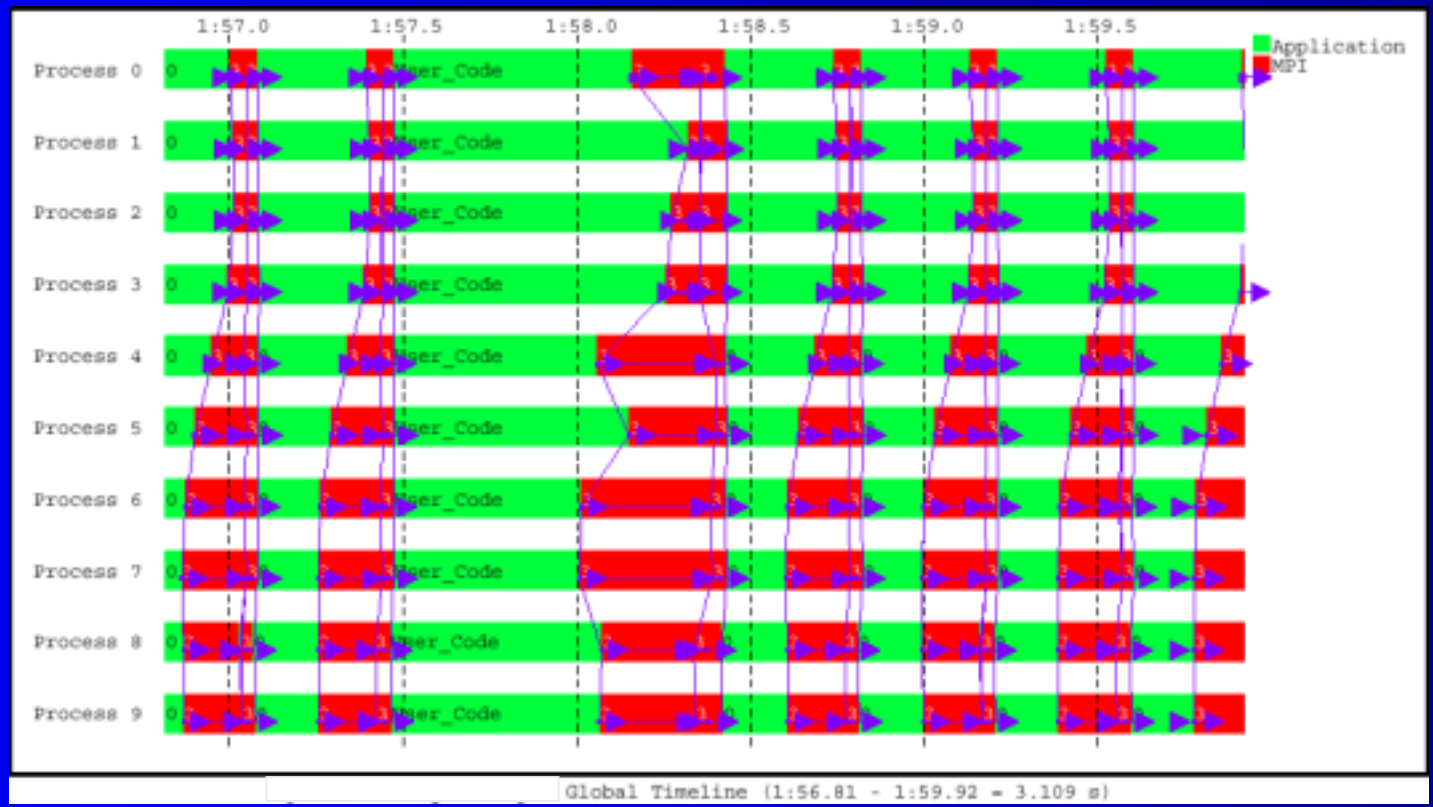
通信時間の違い

Myrinet

2cpu並列の場合

VAMPIRによる性能評価予測(6)

プロセッサ性能の異なる10台によるトレース図



並列負荷分散が悪いと上記と同じような様相を示す

VAMPIRによる性能評価予測(7)

媒体の違いによる MPI_AllReduce の性能実験

BYTE	2cpu-FE	4cpu-FE	8cpu-FE	2cpu-GM	M2000-2	M2000-4
0	0	0	0	0	0	0
4	161	342	538	34	27	53
8	161	341	496	36	28	55
16	172	342	500	36	28	55
32	163	343	499	36	29	57
64	166	343	700	38	30	59
128	174	356	601	48	39	76
256	216	436	688	61	49	96
512	301	602	1165	74	62	119
1024	473	941	1464	100	88	168
2048	709	1423	2243	153	141	267
4096	1072	2166	3309	228	216	413
8192	1799	3586	5437	371	368	720
16384	3234	6506	9788	657	571	1144
32768	6204	12310	18464	1095	967	1948
65536	12423	24488	36119	1986	1849	3597
131072	25795	49905	74648	11034	7280	11340
262144	51494	99858	148228	16217	14611	24817

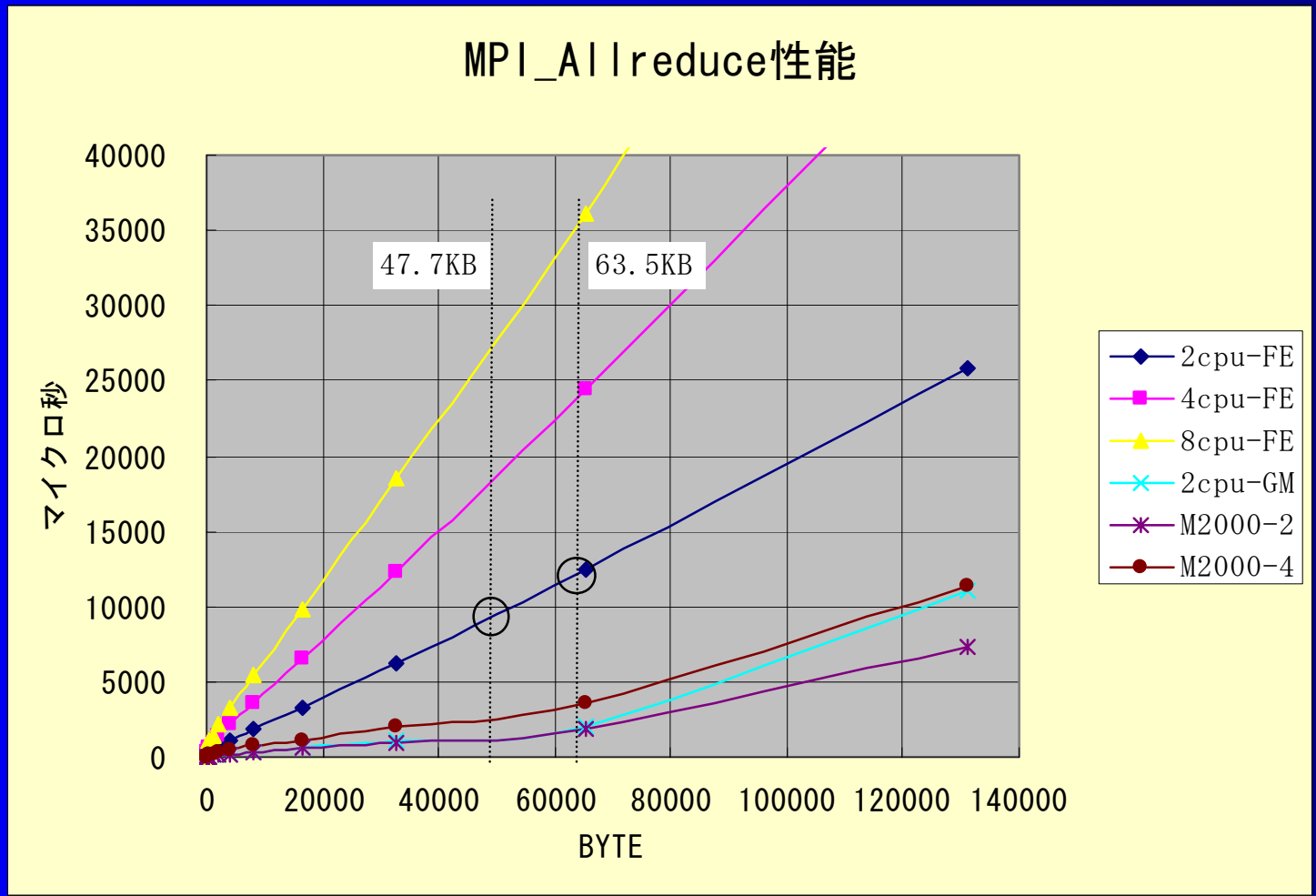
(Byte)

単位:時間(usec)

使用している通信パターンでのベンチマーク比較

FE : Fast Ethernet
GM : Myrinet(1.25Gbps)
M2000 : Myrinet2000(2Gbps)

VAMPIRによる性能評価予測(8)



VAMPIRによる性能評価予測(9)

性能予測式の作成

$$T(p) = k \times (T(1) - Tscalar) / p + Tcom + Toverhead + Tscalar$$

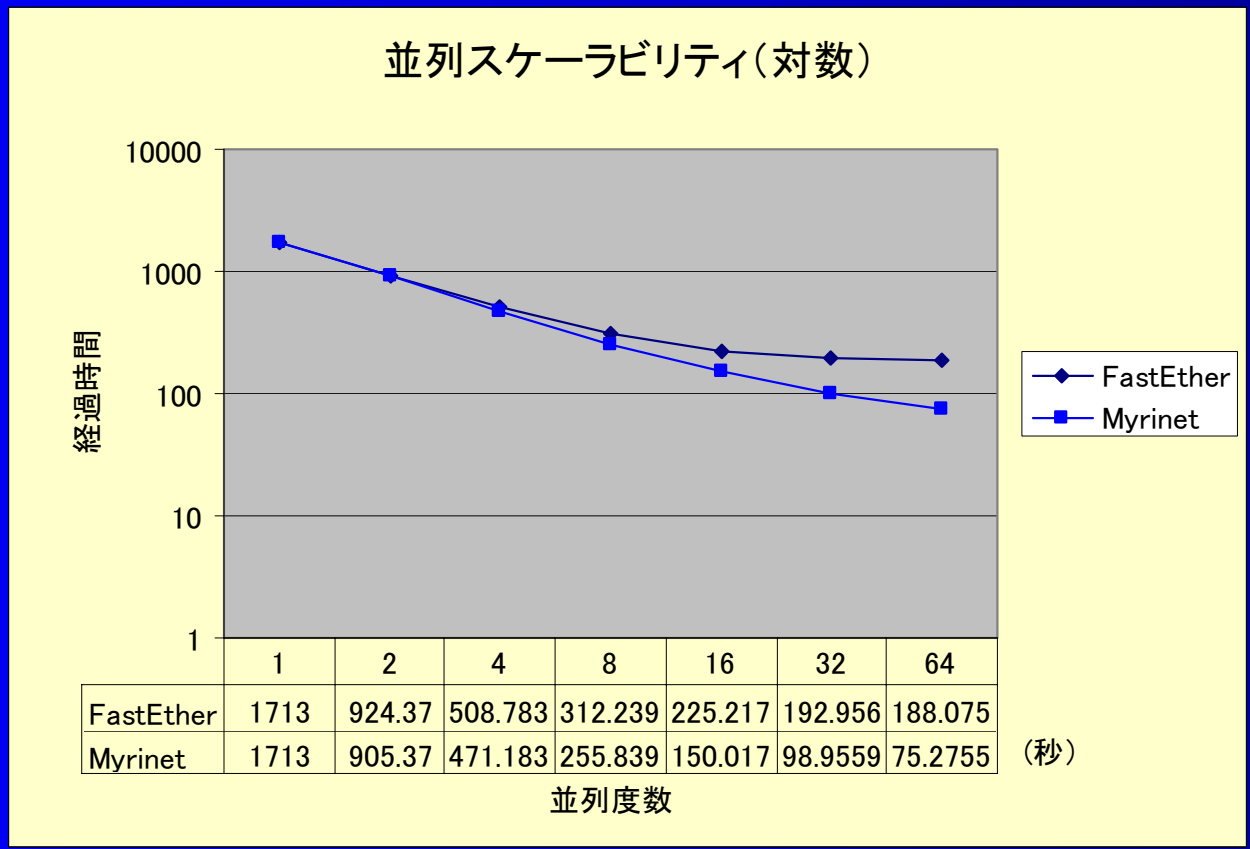
T(p) は並列性能時間 ； p はプロセッサ数
kは、並列に伴うextra workを含めるための係数（本計算では1と近似）
Tcomは、通信時間（reduction通信近似）
Toverheadは、負荷分散不均衡による待ち時間
Tscalarは、I/Oを含むスカラ計算部分の時間（T(1)の1.5%とする）

$$T(p) = 1.68735/p + 0.0225 \times \log2(p) + 0.065/p + 0.025695$$

Fast Ethernet 係数 : 0.0225
Myrinet 係数 : 0.0037

VAMPIRによる性能評価予測(10)

並列 Scalability の予測　(Fast Ethernet vs. Myrinet)

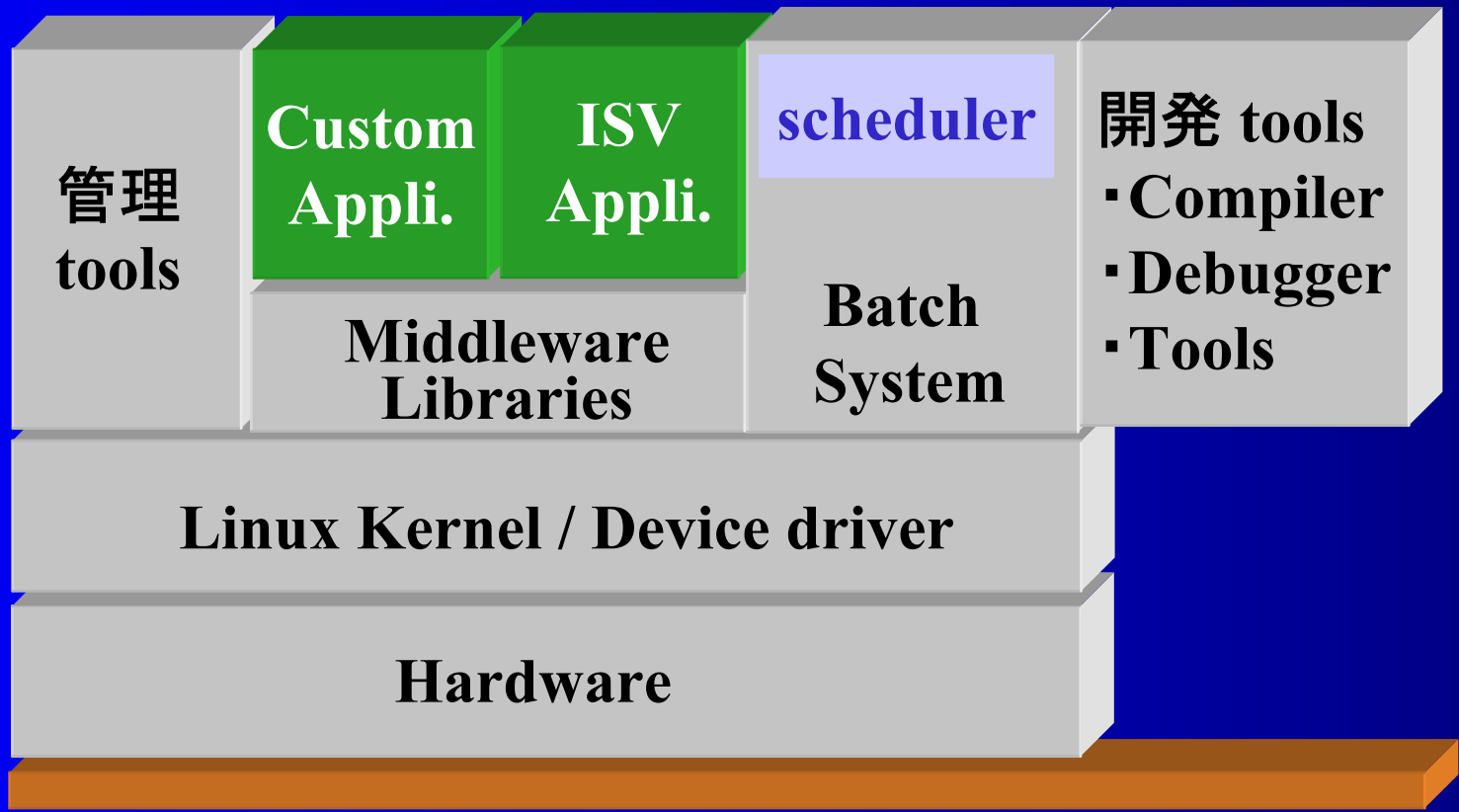


	1	2	4	8	16	32	64
FastEther	1713	924.37	508.783	312.239	225.217	192.956	188.075
Myrinet	1713	905.37	471.183	255.839	150.017	98.9559	75.2755

Myrinetを
買うか否か？

期待性能要件
を決めることが
大事

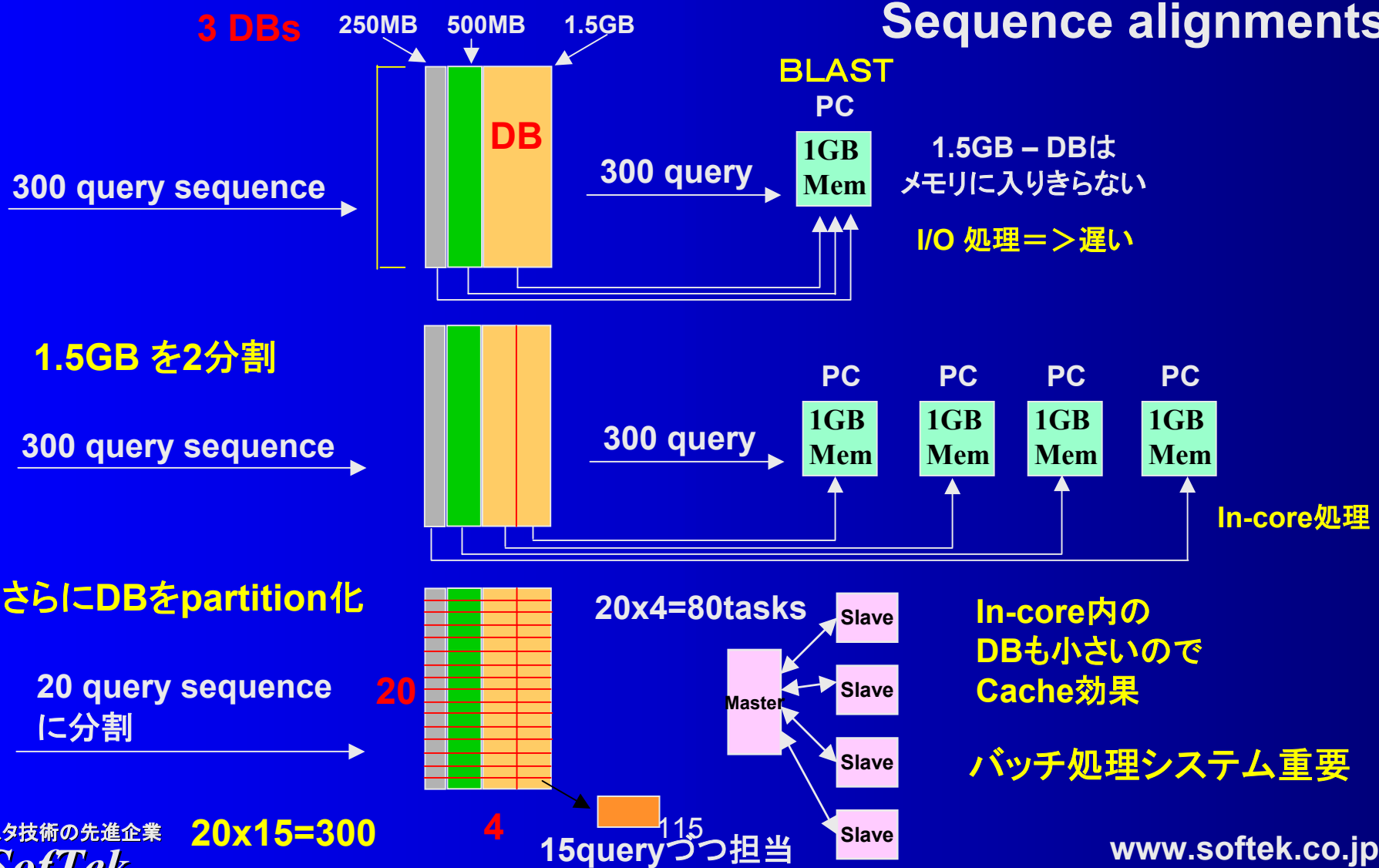
応用事例



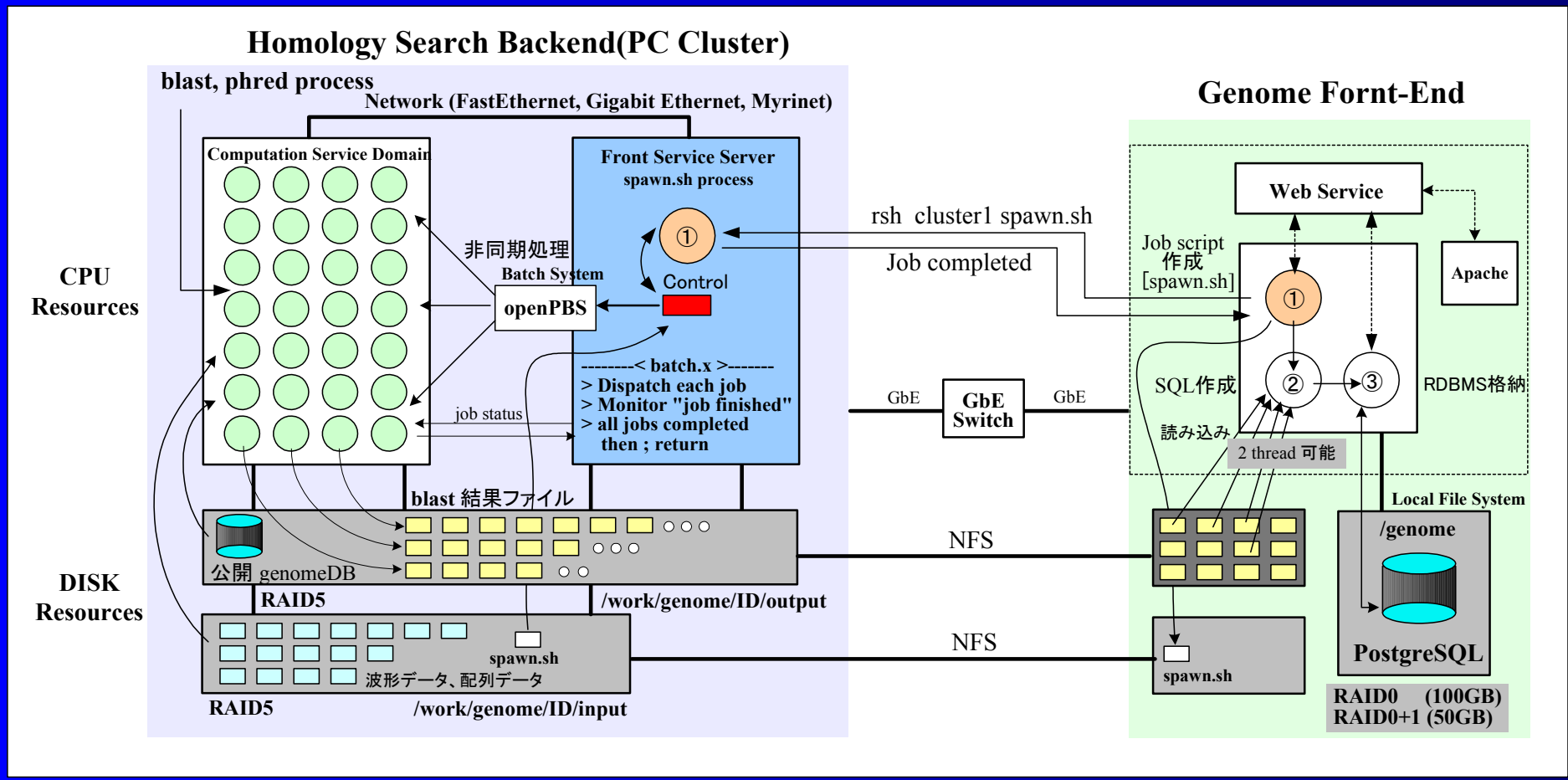
High Throughput Computing

- 多数のジョブの実行とその管理
- アイドリングリソースを有効に
- 企業のIT投資へのコスト管理が重要に
- Parametric Study
- バッチシステムの有効性
  - Bioinfomatics
  - MCAE
  - Electric Device開発
  - 金融デリバティブ計算
  - Scientic Research
  - その他、多数

Bioinfomatics BLASTとPCクラスタ



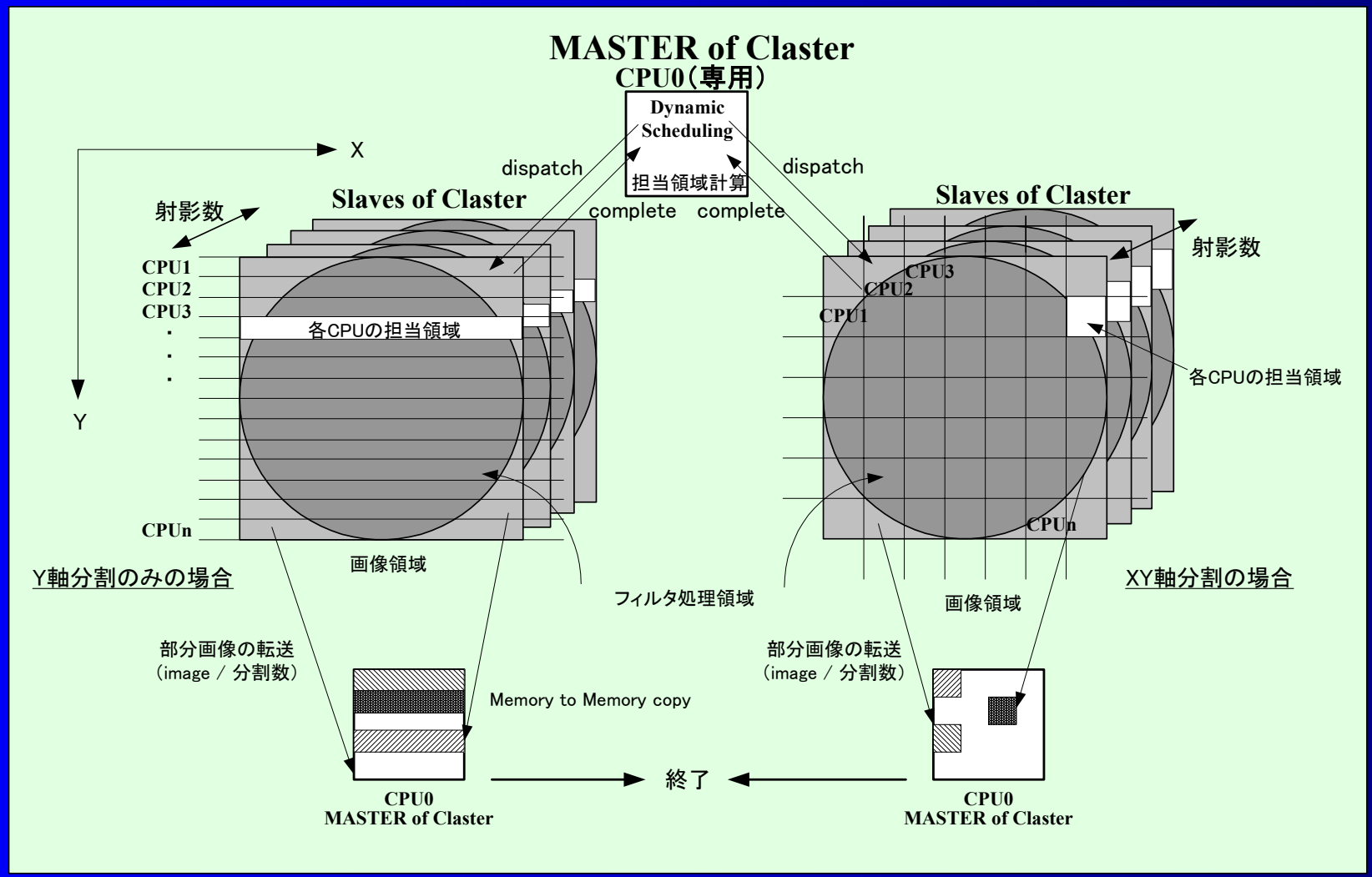
Bioinfomatics (Homology search)



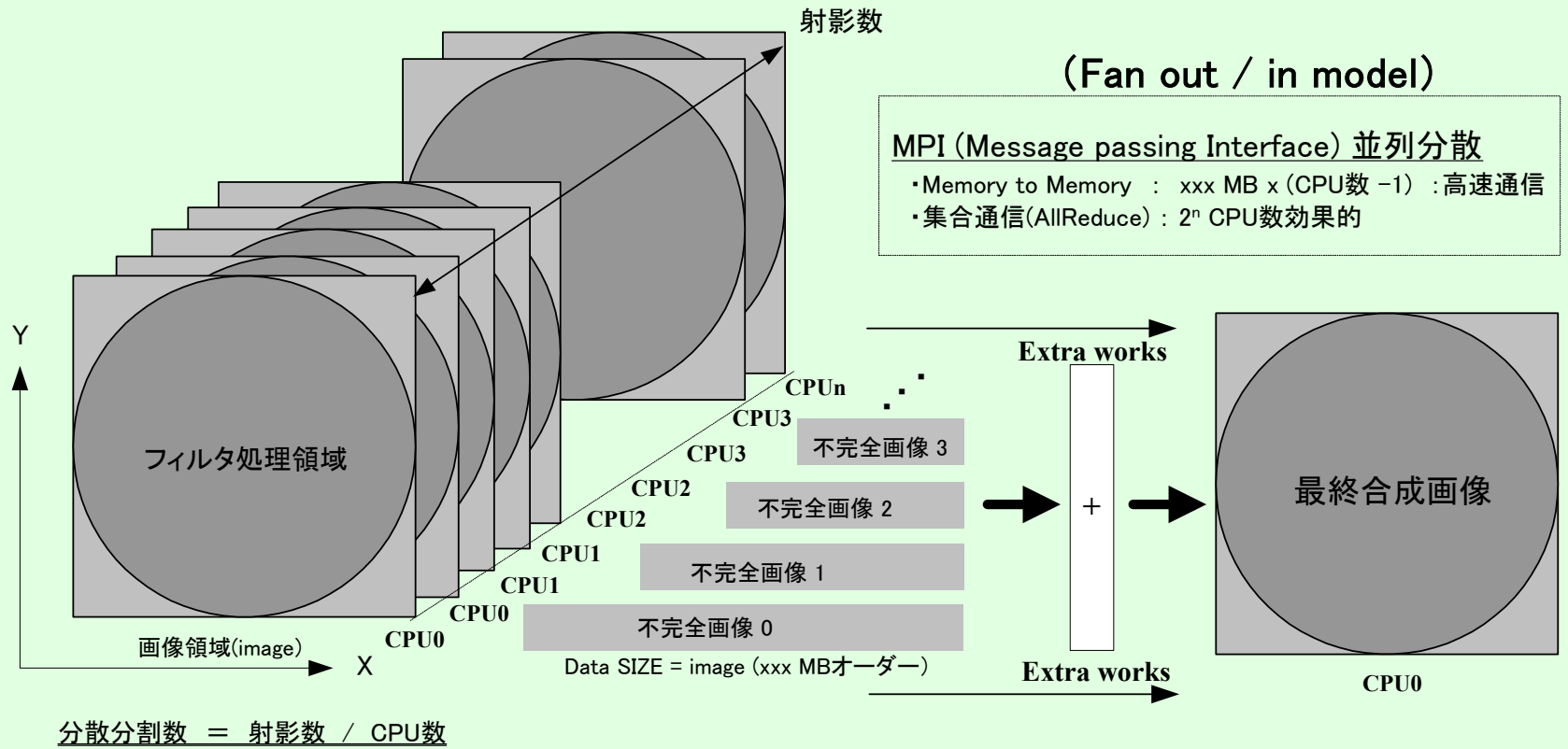
High Performance Computing

- 益々、シミュレーションの重要性が問われてきた
- シミュレーションの大型化
- エンジニアリングの現場でもこの波が!
- 1処理の高速化
- 高速性が必要
- プログラムの並列化が必須＝＞MPIによる
  - MCAE (構造解析,流体解析....)
  - 大型画像処理
  - 物理化学(MD、第一原理、、、、)
  - 気候、気象モデリング

領域分割(Master-Slave Model)



MPIによる並列分散モデル

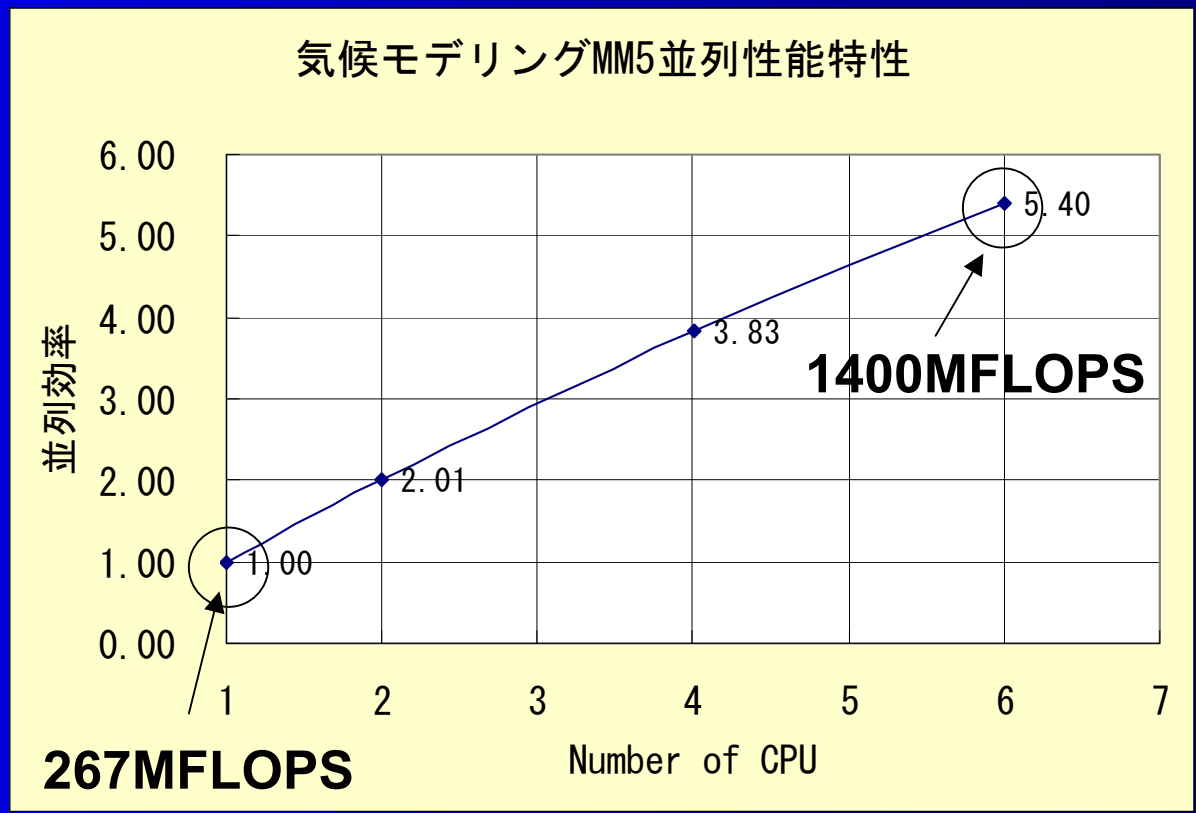


現行システム11,000秒（1CPU）=>220秒（Dual-Pen3:10台）へ

気候モデリングMM5並列性能

Pentium4 1.8GHz
128MB-Rumbus
Fast Ethernet

Fast EthernetでもこのScalability



領域分割による
通信量を最適化
したモデル

Pentium4が4台
程度でGFLOPS
の世界へ!!!

ISVアプリケーションの並列化状況

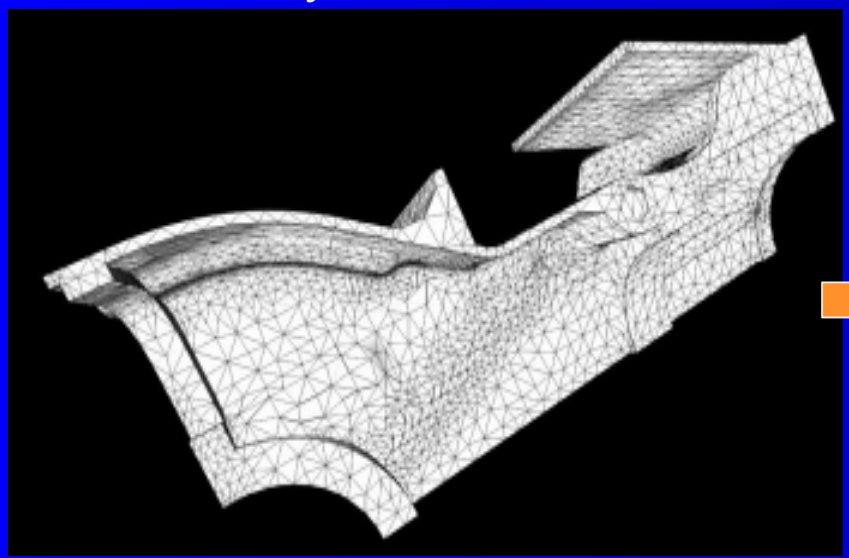
- CAE分野でのISVの並列化対応(IA-32用)
  - 線形・非線形構造解析系
    - ANSYS （DDS 領域分割）
    - MSC.MARC （DDS 領域分割）
  - 陽解法非線形構造過渡解析
    - ANSYS/LS-DYNA
  - 流体解析（CFD）系
    - STAR-CD(STAR-HPC)
    - FLUENT
    - CFX

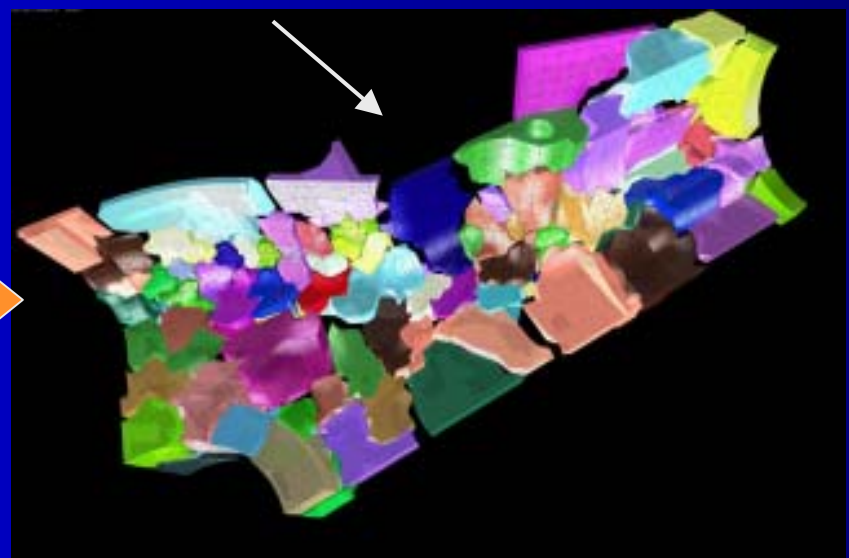
  他、現在、多数ポーティング中

ANSYS v5.7 のDDS (Distributed Domain Solver)

- 領域分割法による並列化
- 細かな Subdomain に分割計算 (直接法)
- 通信はMPI/Proパッケージを使用
  - TCP/Ethernet
  - VIA
  - Myrinet

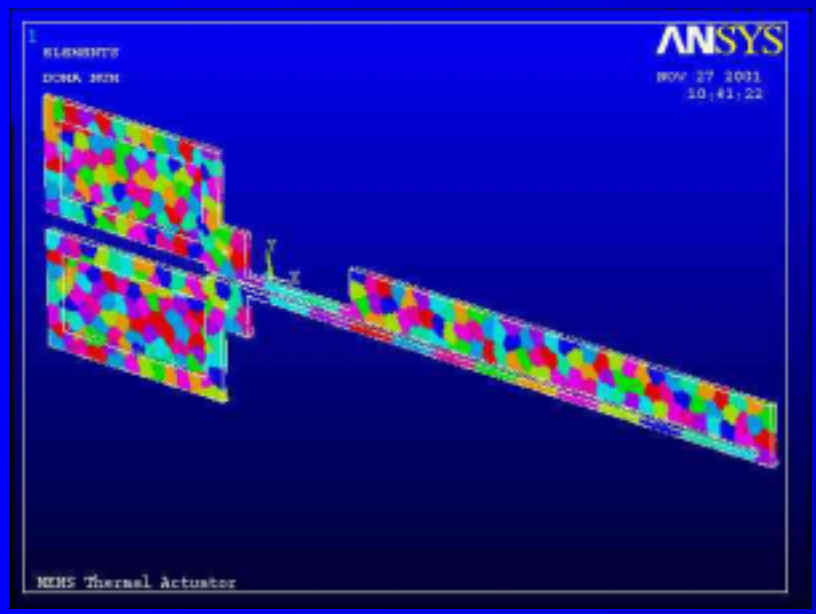
subdomain

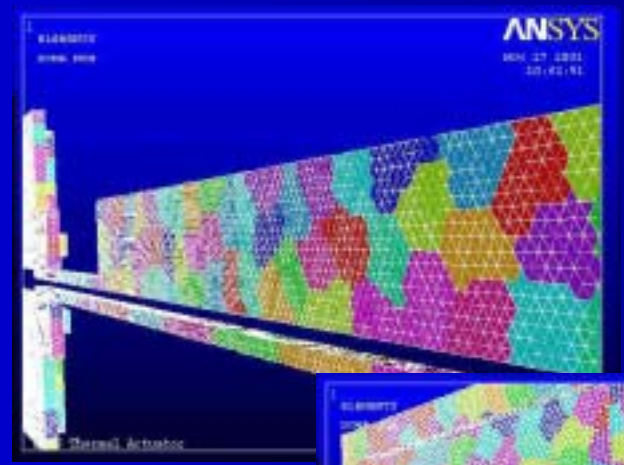


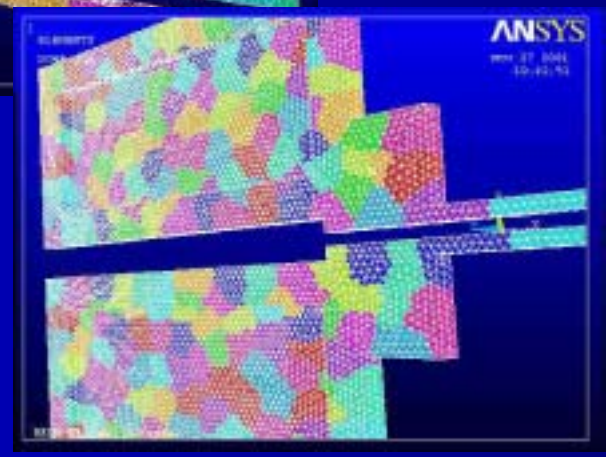


イメージ提供 : ANSYS,Inc.

ANSYSの並列計算例





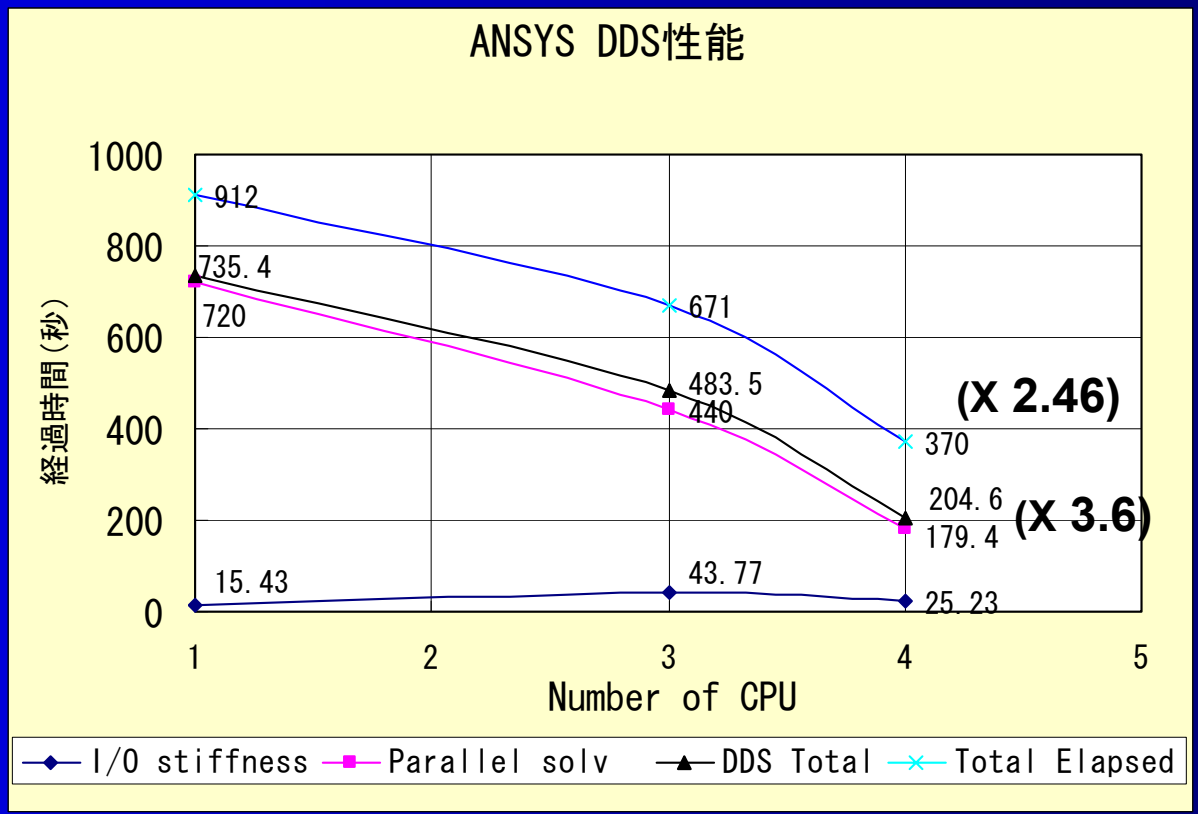


自由度数	605388
要素数	117696
節点数	201796
MPI	MPI/Pro1.6.3(TCP版)
解析内容	静的線形構造解析
要素タイプ	SOLID92
サブドメイン数	316

データ提供：サイバネットシステム株式会社

ANSYS DDS の性能効果

<Hardware>
Pentium3 1.33GHZ
Memory 0.5GB
NIC Gigabit
<Software>
OS Linux 2.4.2
ANSYS 5.7.1

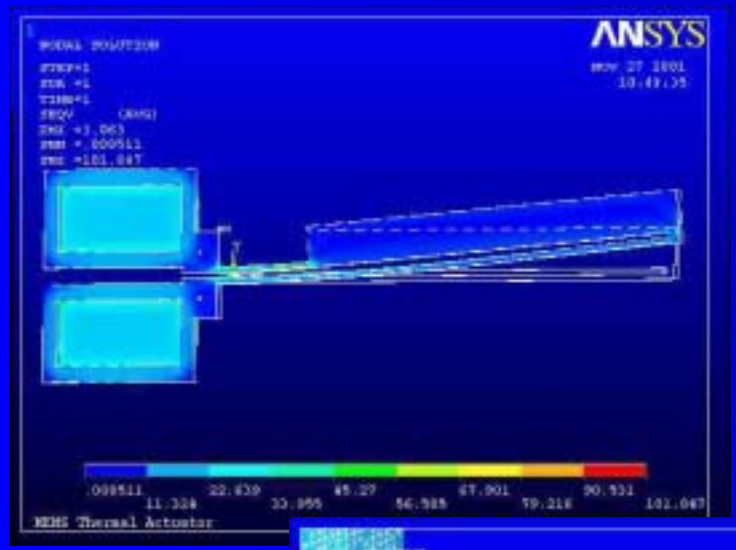


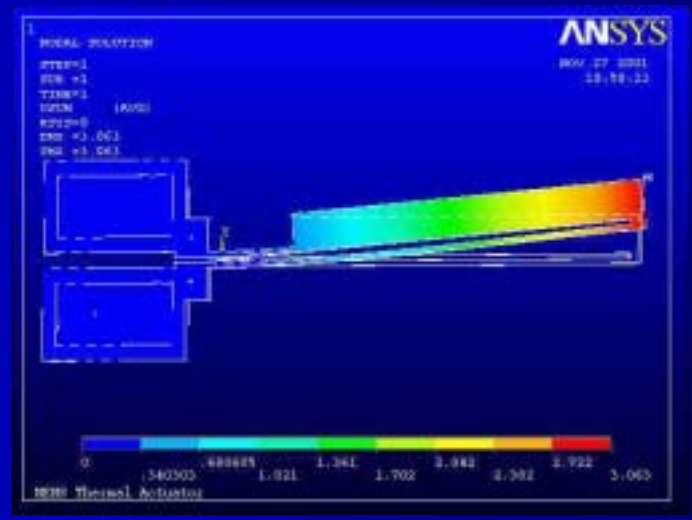
自由度数	605388
要素数	117696
節点数	201796
MPI	MPI/Pro1.6.3(TCP版)
解析内容	静的線形構造解析
要素タイプ	SOLID92

通信バンド幅が必要！

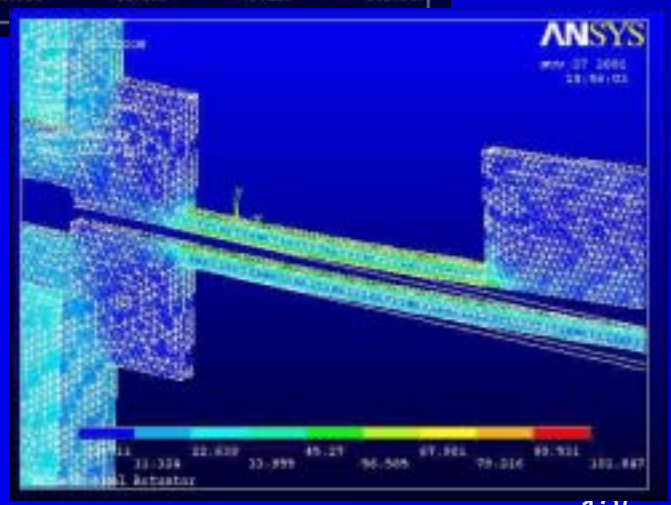
データ提供:サイバネットシステム株式会社

線形構造解析結果





変形図



相当応力図

データ提供：サイバネットシステム株式会社

クラスタシステムは実用期に!

- 目的は何？（並列、分散、H.A.、）
- 使用アプリケーションの特性は？（演算、並列性、...)
- 要求する性能要件を定義
- 最適なH/Wコンポーネントを選択
  - 価格性能比（高い計算機部材が必要か？）
  - 信頼性はどの程度、設置性との関連
- 最適なソフトウェアの選択
  - ハードウェア購入より、安く性能を向上させることが可能
- システムの構築
  - 目に見えない問題の解決＝＞Service Providerへ

結果的にエンジニアリング費含めても安い環境構築が可能

Cluster Information Directory

Powered by *SofTek*

http://www.softek.co.jp/CID/

PCクラスタに関するH/W、S/W技術情報サイト

- PCクラスター般情報
- 性能情報
- ベンチマーク
- ハードウェア情報
- ソフトウェア情報
- Linux OS情報
- クラスタ構築・応用ノウハウ

Cluster Solutions Service by SofTek

・最適なソリューションの提供

・最適なコストでのシステム構築

・クラスタ総合技術力でバックアップ

ソフテックは、

テクニカル・サービス・プロバイダ

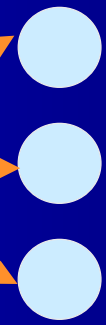
http://www.softek.co.jp/Cluster/

ソフテック -Technical Service Provider-

クラスタ関連
ミドルウェア製品

SofTek
Technical Service Provider

H/W Providers



IBM,Dell,SGI
NEC,日立,富士通
CTC他商社系
秋葉原ショップ系

お客様

最適システム選択
並列システム設計
並列システム構築
並列システムチューニング
並列アプリケーション開発
プロダクションシステム開発

・予算とニーズ
・最適なシステムの選択
　・対投資効果の見解
　・アプリケーション
　　特性との相性
・最適な価格でのH/W購入
・並列プロフェッショナル
　サービス

・ニーズに応じた選択
・コスト最小優先
・信頼性・保守優先
・見積合わせ可能
・入札システム導入

ソフテックのバックグラウンド

・官公庁大型スパコン・並列システムの最適設計
　積算・性能評価等の総合導入コンサルティング
・アプリケーションの並列化実装、性能評価実績
・並列ミドルウェアの共同研究開発

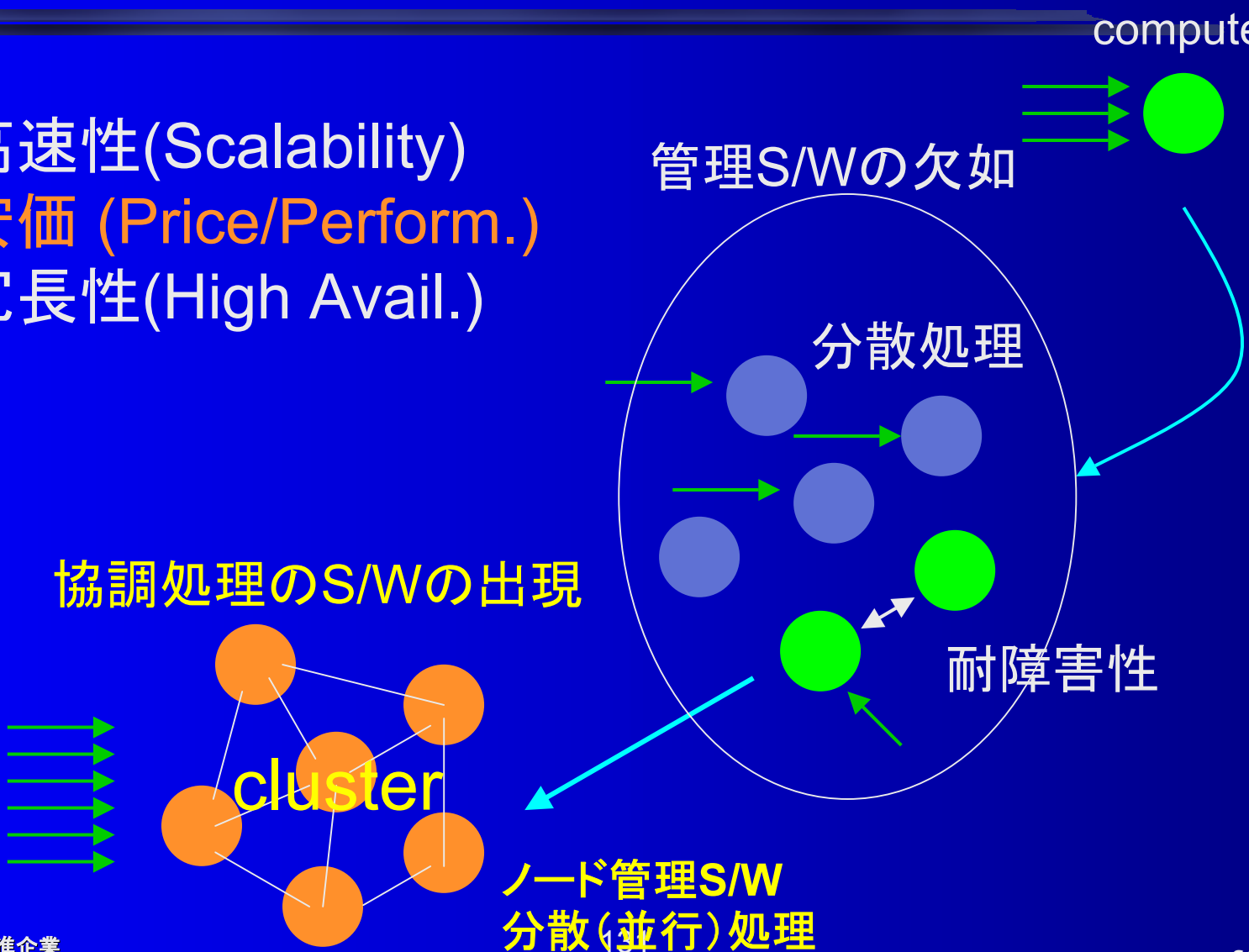
http://www.softek.co.jp/Cluster/

付録

- クラスタシステムの適用分野
- Pentium プロセッサの性能
- 並列効果

クラスタリングは自然の流れ

- 高速性(Scalability)
- 安価 (Price/Perform.)
- 冗長性(High Avail.)



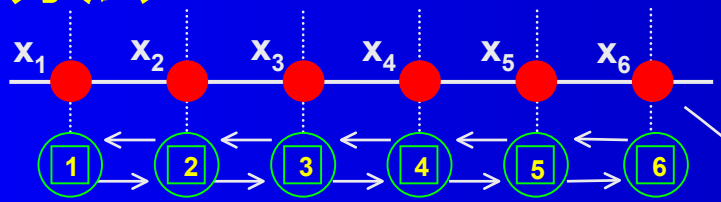
科学技術分野でのPCクラスタ適用

Scientific & Engineering Simulations

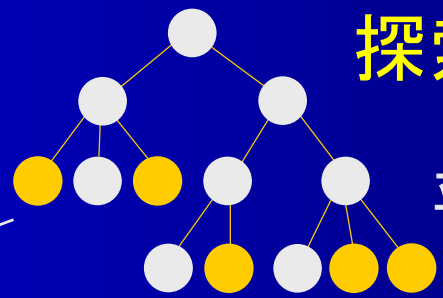
（他、多数）

高速性を求める

差分法

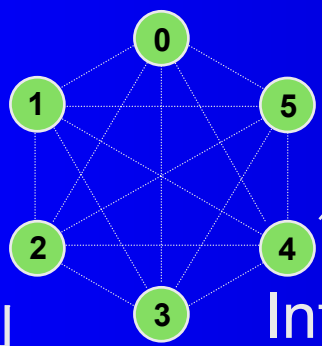


探索問題



並列・分散

N体問題

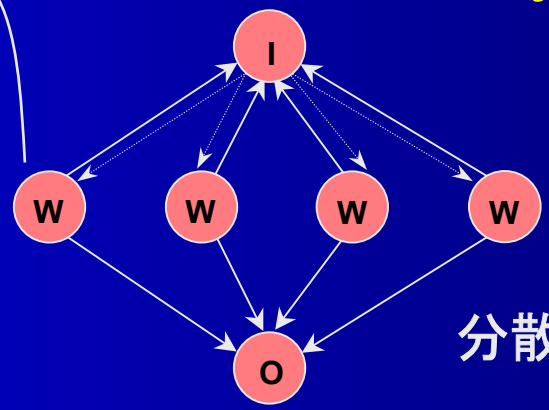


並列

Interactions

1CPUに相当

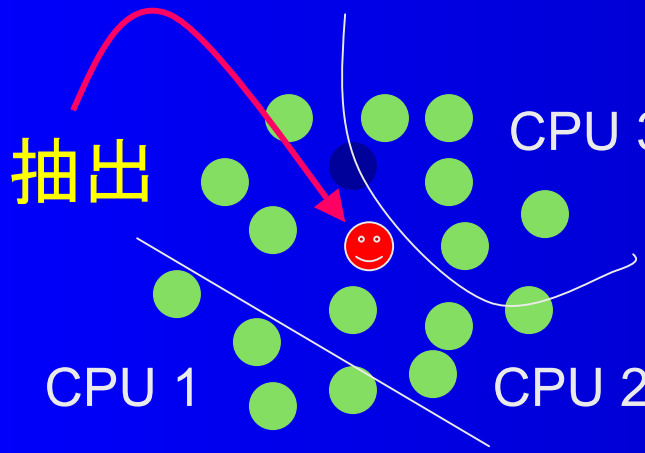
Parametric study

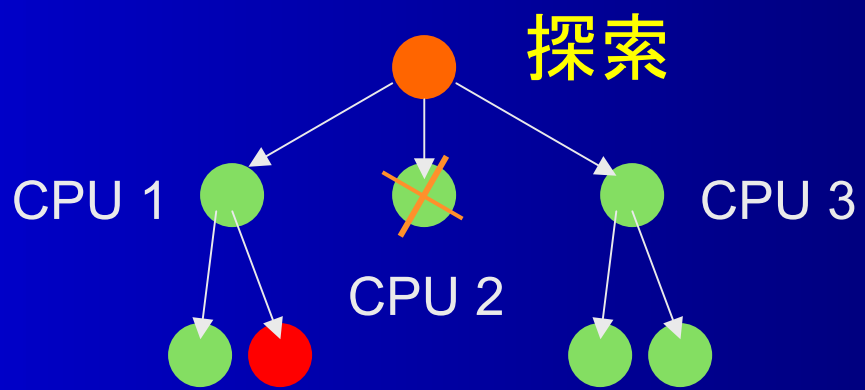


分散

ビジネス分野でのPCクラスタ適用(1)

- ビジネス分野での一例 (複雑なものから探し出す)
  - 経路最小最適化問題 (通信、交通、scheduling等)
  - データ探索(データマイニング)
  - 金融ポートフォリオ、リスク管理(最適化問題)





現在社会の最重要課題
のシミュレーション(Decision Support Systems)

ビジネス分野でのPCクラスタ適用(2)

- ビジネスアプリケーション(バックエンドシステム)
  - 金融デリバティブーバッチ処理計算(J.P.モルガン….)
  - E-コマース (Amazon.com, eBay.com ….)
  - データベース (Oracle on cluster,IBM-Microsoft-Intel Project)
  - Decision Support Systems
  - データマイニング
- インターネットアプリケーション
  - Web serving
  - Infowares (検索エンジンサイト、google.com)
- その他
  - 映画デジタルフィルムの作成
  - ネットワーククラッキング解析

Search Engine (www.google.com)

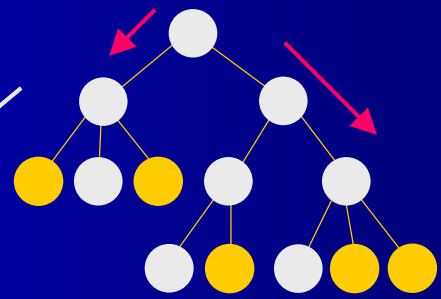
- **Google combines an easy-to-use interface with complex algorithms to determine the importance and relevancy of Web pages--a task that requires a high-performance back-end system.**

サーチエンジンと
そのバックエンド処理

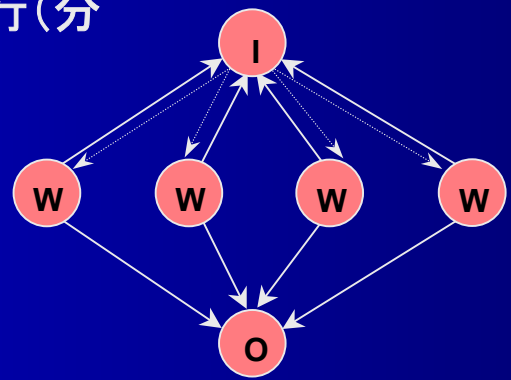
a cluster of more than
4,000 PCs

Linux Clusterの応用（大規模）

- http://www.genetic-programming.com/
  - 遺伝子複製的処理（ツリー階層）によるプログラム自動生成／合成手法
  - 高位レベルの問題から自動的にプログラムを作って実行
  - Beowulf-style 1,000 Pentium II 350 MHz processors
  - Connected with 100Mbps Ethernet



- “Titanic” filmの作成 (Digital Domain社 ‘97)
  - 3Dモデルの基本要素を作成後、ショットのフレームを並行（分散）処理で作成。floating-point-intensive部分
  - 160 433MHz DEC Alpha - Linux systems
  - Connected with 100Mbps Ethernet

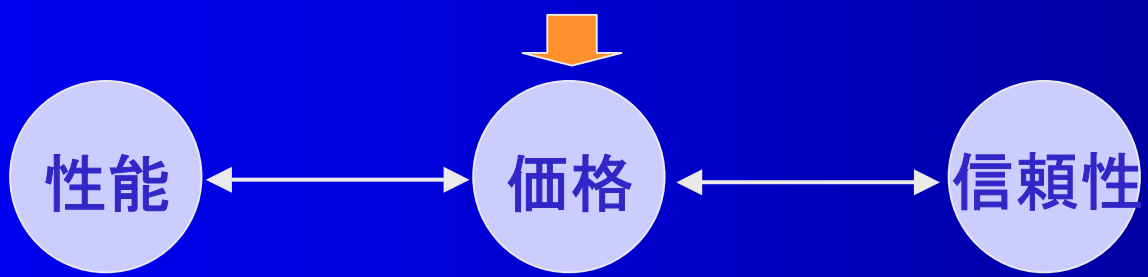


クラスタシステムをどのように使う？

- 動機付けを明確に
  - 高速性能　　：並列処理
  - 高スループット　：分散（並行）処理
  - 高信頼性　　：HA、ノード管理、フェールオーバ

生産性に対する、求める性能要件の定義

アプリケーションの特性・機能性



必要十分なH/Wコンポーネントを揃える

例えば

4倍の並列性能で
10倍のスループット

日本のユーザは
常にOver Spec.
を望む？

現在のプロセッサ性能を理解しよう

性能と価格　：　カタログ値では分からない

賢く購入するには、実アプリ性能で比較しよう

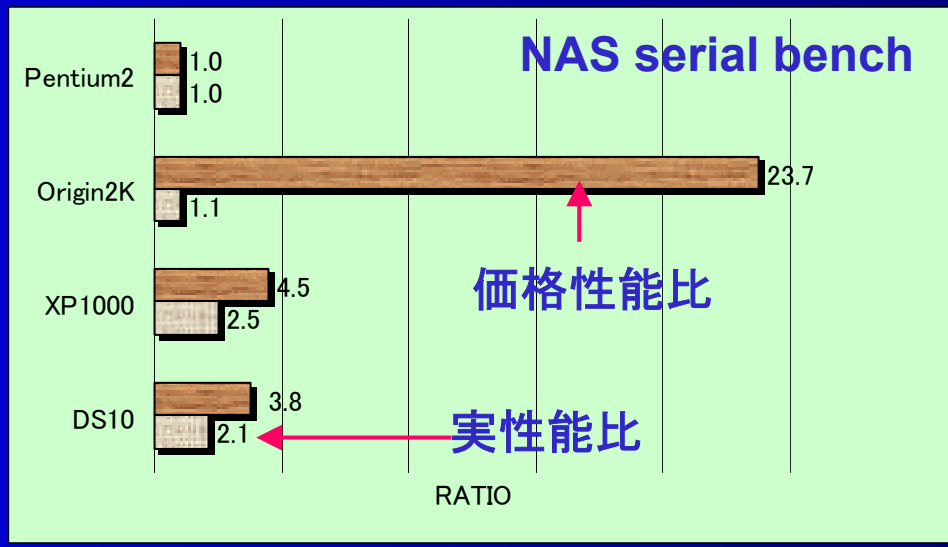
	Alpha-21264			Pentium			AMD
	DS10L	XP1000	DS20E	P3 500	P3 800EB	P4 1500	Athlon
クロック周波数 (MHz)	600	667	667	500	800	1500	1200
バスクロック(FSB:MHz)	100	333	333	100	133	400	266
2nd Cache Size	2MB	2MB	4MB	256KB	256KB	256KB	256KB
理論ピーク性能 (MFLOPS)	932	1334	1334	500	800	1500	1200
理論メモリバンド幅(GB/s)	1.3	2.6	5.2	0.8	1.06	3.2	2.1
実効メモリバンド幅(MB/s)	677	967	1346	406	544	1555	907
標準価格 (万円) (Mem:256MB)	?	?	?	10	13	25	23
実勢価格(含キャンペーン価格)	65	90	200	10	13	21	20

- **Pentium の性能は、他の商用 MPU 性能を超える？**
- 理論ピーク性能だけでは、価格性能比を語れない
- 実効メモリバンド幅：STREAMベンチマークより

価格性能比の一例（流体計算）

- CPU/Mem intensive
- キャッシュ外性能重要

- 実効性能比が分かれば MPU数と性能の比較 検討が可能



NAS Serial Benchmark	Alpha-21264		MIPS	Intel	AMD
	DS10	XP1000	Origin2K	Pentium2	Athlon
クロック周波数 (MHz)	466	667	195	400	500
キャッシュ内 計算主体(MFLOPS)	195	258	83.1	69.4	104.5
キャッシュ外 計算主体(MFLOPS)	142	166	69.9	66.3	-
キャッシュ内 計算主体(Ratio)	2.8	3.7	1.2	1.0	1.5
キャッシュ外 計算主体(Ratio)	2.1	2.5	1.1	1.0	-
実勢価格（概算：万円）	65	90	200	8	10
MFLOPS当たりの単価（円）	¥4,577	¥5,422	¥28,612	¥1,207	
Price/Perormance Ratio	3.8	4.5	23.7	1.0	

計算化学系アプリの実効性能（1）

GAUSSIAN98の実性能比較

GAUSSIAN 98 (Rel.A7) Benchmark Code	P-III 500 128 MB	Athlon K7-550 512 MB	RS6000 397 512 MB	Origin 2000 15.5 GB
bench1 : RHF OPT	9673	8645	6602	10042
bench2 : RHF FREQ	1707	1452	1097	1743
bench3 : MP2 OPT	13380	9998	7952	12546
bench4 : MP2 FREQ	13871	8195	5868	12213
bench5 : QCISD(T)	13947	5609	3741	7785
bench6 : CASSCF	2658	2260	3531	3222
bench7 : CIS	2371	2003	1766	2706
bench8 : TD-DFT	1042	831	652	953
調和平均値	3001.4	2388.2	1995.1	2971.1
Performance Ratio	**1.0**	**0.80**	**0.66**	**0.99**

低いほど速い

Source:http://www.chm.tu-dresden.de/edv/bench/bench.html

計算化学系アプリの実効性能(2)

GAMESS(US)の実性能比較

実行時間(秒)

GAMESS (US) Ver.1 as 1998	P-III 500 128 MB	Athlon K7-550 512 MB	RS6000 397 512 MB	Origin 2000 15.5 GB	Alpha ES 40 4 GB
bench1 : RHF OPT	12.9	9.4	13.0	14.9	5.2
bench2 : RHF FREQ	6.5	4.9	6.4	9.3	3.0
bench3 : MP2 OPT	33.4	25.9	28.3	43.0	14.1
bench4 : CASSCF OPT	431.0	68.2	123.0	137.2	46.2
bench5 : CASSCF EN	79.3	63.5	77.9	90.0	32.2
bench6 : MCQDPT EN	194.3	163.0	298.0	250.4	85.4
調和平均値	21.3	15.6	20.5	27.2	9.1
Performance Ratio	**1.00**	**0.73**	**0.96**	**1.28**	**0.43**

667MHz

Pentiumは、既存の商用プロセッサと比較して遜色ない

Source:http://www.chm.tu-dresden.de/edv/bench/bench.html

Gaussian98 : Pentium3 vs Pentium 4

Gaussian 98, revision A.7.

b)Ethylene, 16 electrons, 1Ag,
D2h point group,
Basis Set = 6-311++G**, (6-term d's)
c)Ethylene, 16 electrons, 1Ag,
D2h point group,
Basis Set=6-311++G(3df,3pd),
(7-term f's, 5-term d's)
d)18-crown-6, C12H24O6, 144 electrons,
Ci, Set=aug-cc-pVDZ, (5-term d's)

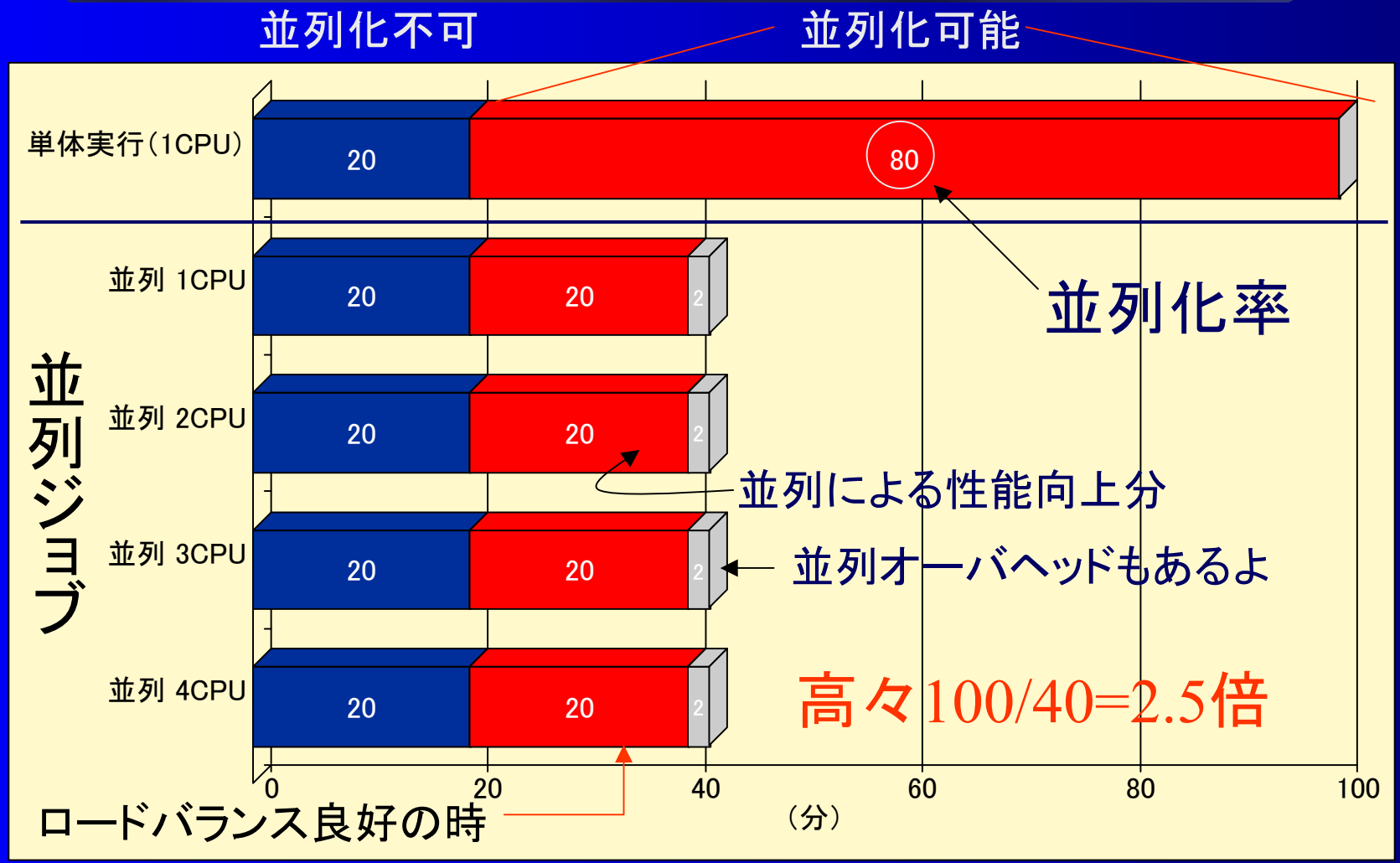
Dual Pentium® III 933 MHz
512 MB Rambus memory, 18 GB
Ultra 160 disk,
Red Hat Linux release 7.0.

Pentium® 4 1.5 MHz
512 MB Rambus memory,
18 GB Ultra 160 disk,
Red Hat Linux release 7.0.

From: "Nordwall, Douglas J"
<Nordwall@pnl.gov>
To: beowulf@beowulf.org
Date: Thu, 12 Apr 2001 08:54:01

		Dell Precision 420[h] (PIII 933 MHz)		Dell Precision 330[i] (P4 1.5 GHz)		
Molecule	Method	CPU	Wall Clock	CPU	Wall Clock	Ratio(wall)
Ethylene[b]	Conv. RHF	4.9	4.9	3.3	3.3	0.67
	Direct RHF	11.9	11.9	8.2	8.2	0.69
	In-core RHF	4.9	4.9	4.5	4.5	0.92
	RHF Gradient	9.4	9.4	6.1	6.1	0.65
	RHF Hessian	41.4	41.4	27.9	27.9	0.67
	UHF	6.5	6.5	4.2	4.2	0.65
	Conv. MP2	7.3	7.3	4.8	4.8	0.66
	Direct MP2	14.2	14.2	9.7	9.7	0.68
	MP2 Gradient	19.5	19.5	12.1	12.1	0.62
	MP2 Hessian	239.1	246.1	129.9	130	0.53
	MP4(SDTQ)	40.5	40.6	29.3	29.3	0.72
	SDCI	27.6	31.7	15.1	15.2	0.48
	CCSD	37.7	41.6	20	20.3	0.49
	CCSD(T)	71.2	75.2	46.6	46.6	0.62
	QCISD	38	48.6	17.2	17.2	0.35
	QCISD(T)	72	79.2	43.4	43.5	0.55
	CASSCF	57.8	80.9	33.4	58.6	0.72
	SVWN (LDA)	31.7	32.4	22.1	22.2	0.69
	BLYP (NLDA)	57	57.1	33.2	33.2	0.58
Ethylene[c]	Conv. RHF	31.9	32	21.6	21.6	0.68
	Direct RHF	119.4	119.9	77.3	77.4	0.65
	RHF Gradient	94.2	95.3	54.7	54.8	0.58
	RHF Hessian	659.8	662	398.6	398.8	0.60
	Conv. MP2	227.1	252.7	226	348.5	1.38
	Direct MP2	161.9	162.4	94.4	94.4	0.58
	MP2 Gradient	270.7	284.5	133.3	136.3	0.48
18-crown-6[d]	Direct RHF	43,589.10	43,560.70	25,994.90	26,048.20	0.60
	Total	45,946.70	46,022.90	27,499.20	27,676.90	0.60

並列の効果（どの位？）

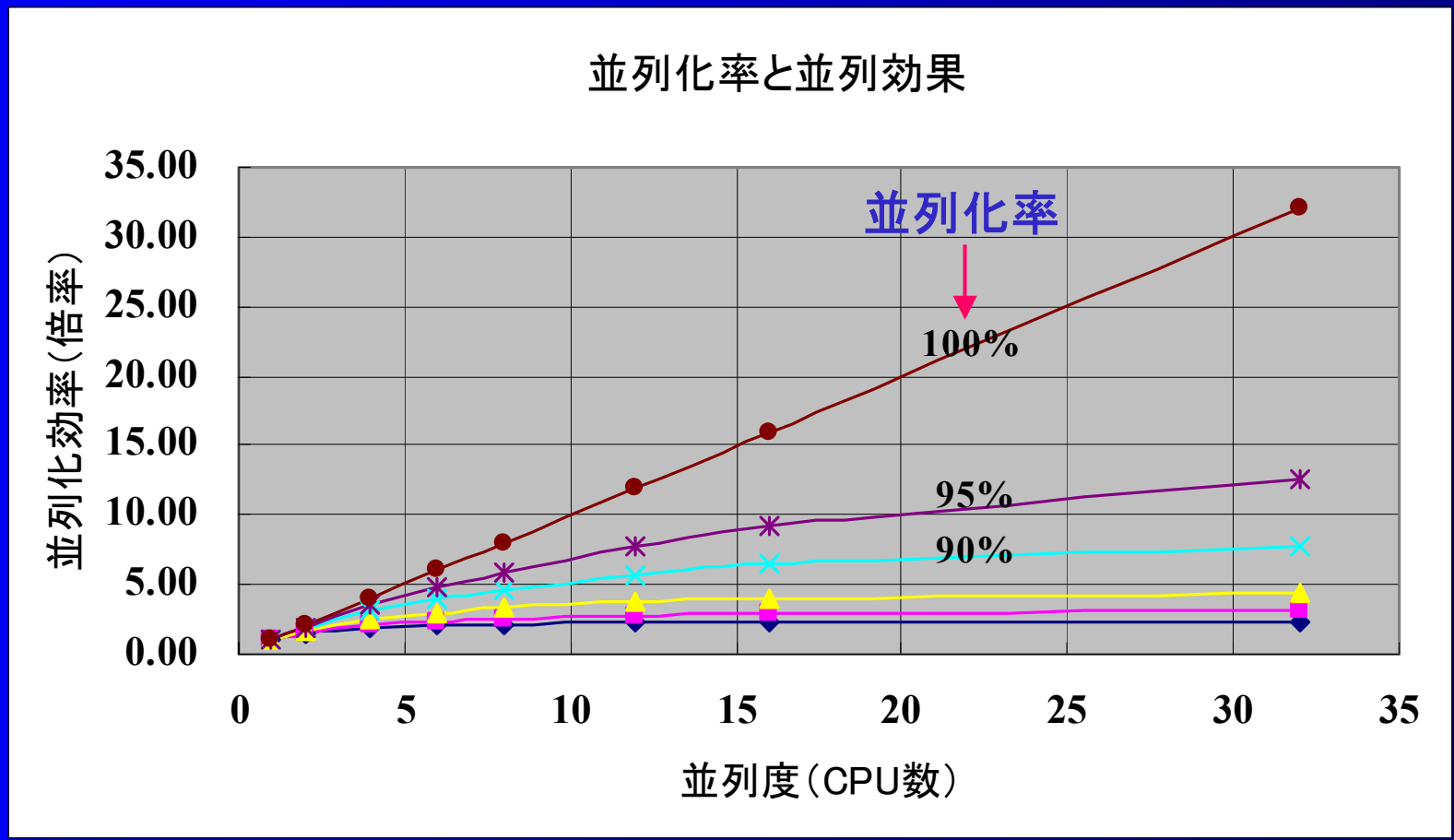


並列の効果 (Scalability)

アムダールの法則＝並列化率 P% の時の効果は
＝ 100 / (100-P) : 理想値

並列化効率	Number of CPU					
並列化率	2	4	6	8	12	16
60.0%	1.43	1.82	2.00	2.11	2.22	2.29
70.0%	1.54	2.11	2.40	2.58	2.79	2.91
80.0%	1.67	2.50	3.00	3.33	3.75	4.00
85.0%	1.74	2.76	3.43	3.90	4.53	4.92
90.0%	1.82	3.08	4.00	4.71	5.71	6.40
91.0%	1.83	3.15	4.14	4.91	6.03	6.81
92.0%	1.85	3.23	4.29	5.13	6.38	7.27
93.0%	1.87	3.31	4.44	5.37	6.78	7.80
94.0%	1.89	3.39	4.62	5.63	7.23	8.42
95.0%	1.90	3.48	4.80	5.93	7.74	9.14
96.0%	1.92	3.57	5.00	6.25	8.33	10.00
97.0%	1.94	3.67	5.22	6.61	9.02	11.03
98.0%	1.96	3.77	5.45	7.02	9.84	12.31
99.0%	1.98	3.88	5.71	7.48	10.81	13.91
99.5%	1.99	3.94	5.85	7.73	11.37	14.88

並列実行での性能低下の傾向

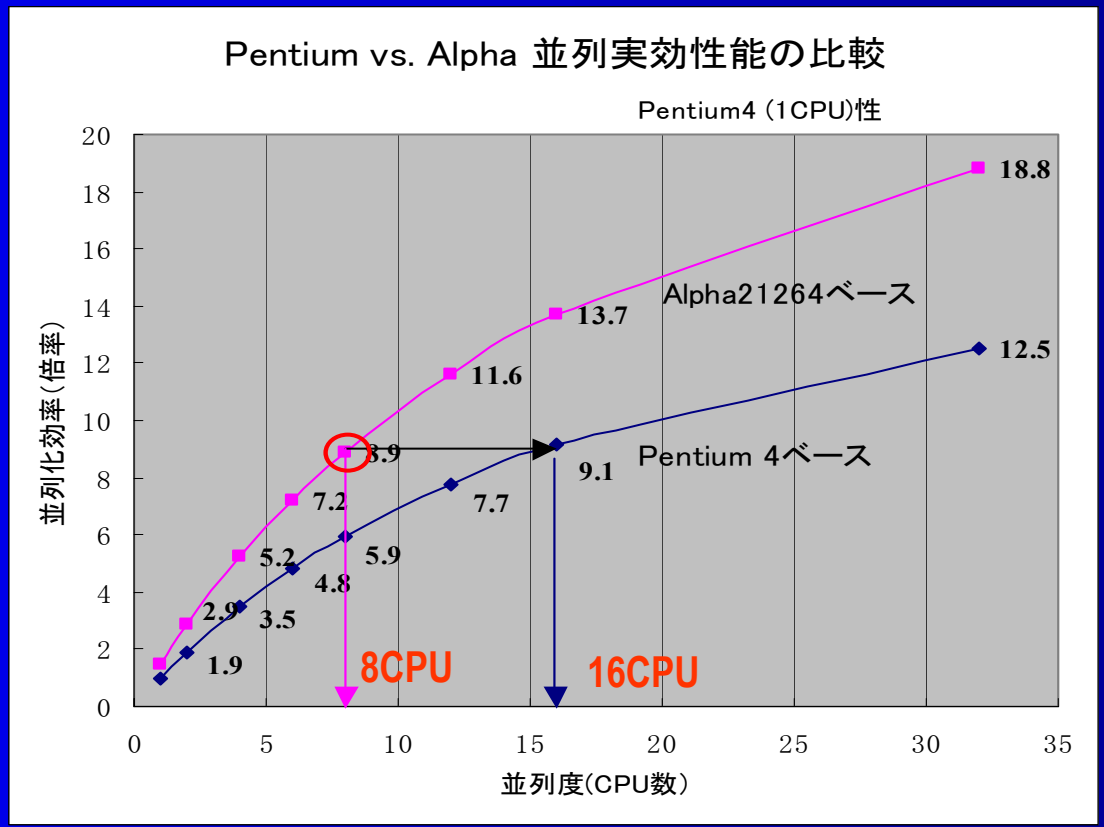


Pentium-box並列 or Alpha-box並列？

1台(CPU)当たりの価格性能比= 1：？

並列化率
95%時

一例



例えば、あるプログラムの1CPU実効性能比
Alpha866MHz：Pen4 1.8GHz **= 1.5：1**