SCORE入門

日本ヒューレット・パッカード
HPC技術部
原田　浩

2006年3月3日

SCore
入門

SCore

- 新情報処理開発機構（RWCP）で開発されたクラスタ計算機用並列プログラム実行環境
- 現在はPCクラスタコンソーシアム（PCCC）が開発、普及活動を行っています
- ギャングスケジューリングを用いたマルチユーザ環境
- 対話型実行環境
- バッチスケジューラ：PBS
- 高速通信ライブラリ：PMv2
- MPI,OpenMP,MPC++などの並列プログラミング環境
- チェックポイント・リスタート機能
- これらの機能をトータルに開発した実行環境です。
- 「LINUXで並列処理しよう」共立出版を参考にして下さい。

SCore
入門

SCoreの歴史

- 元々はRWC-1用のOS（実行環境）として研究開発
- RWC-1の開発が進むにつれて、コモディティなマシン上で並列プログラム実行環境の重要性が高まってきました。
  - NoW、Shrimp、Beowulf
- SUNワークステーション上にSCoreを開発しました。
  - SUN SS20 x 36 ノード　Myrinet LANAi3/SBUS
  - ノード間通信性能 30MBytes/sec 程度

SCore
入門

SCoreの歴史

Powered by SCore

- PCを用いてNetBSD上に移植
  - 66MHz DXII DEC PC
- PICMG PC 32ノードのPCクラスタを製作
  - エンクロージャ等は東清物産（当時）様に発注
  - 東清システムインテグレーションズ様（http://www.tosei-si.com/）は“COSMO”クラスタとして商品化
  - ピッツバーグで開催されたSCに展示して注目を集めました
    - サンディエゴ、ポートランド、オーランド、ダラス、コロラド、ボルチモア、フェニックス

RWC PC Cluster I
NetBSD
Diskless Boot

SCore
入門

SCoreの歴史

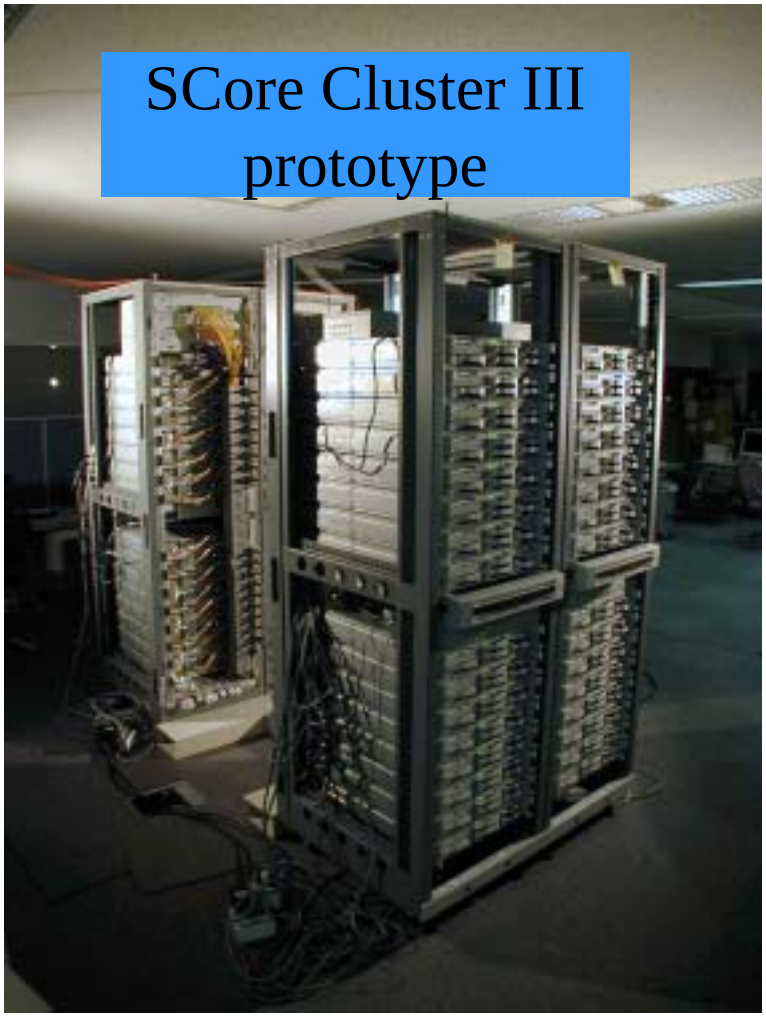
Powered by SCore

- サンノゼのSC（2回目の展示）でLINUX対応
- RWC PC Cluster IIクラスタ
  - PICMG　PenPro 200MHz　128ノード
  - Myrinet LANAI 5
  - 19インチラック4本
  - SCで展示しました。

RWC PC Cluster II
Linux

SCoreの歴史

- SCore Cluster III
  - Pen3 933MHz Dual
  - NEC Express 512 ノード
  - Myrinet 2000 シリアルケーブル



SCore
入門

SCoreの歴史

SCore
入門

SCore

- OS
  - LINUX
    - Fedora Core3
      - x86 SCore 5.8.3 バイナリパッケージがPCCCから配布
    - Redhat7,8,9
    - SuSE
- プラットフォーム
  - IA32
  - X86-64(EM64T/AMD64)
  - IA64
- 並列計算用ネットワーク
  - Myrinet2000,Myrinet-XP,Myrinet-2XP
  - Ethernet
  - Infiniband

SCore
入門

インストール

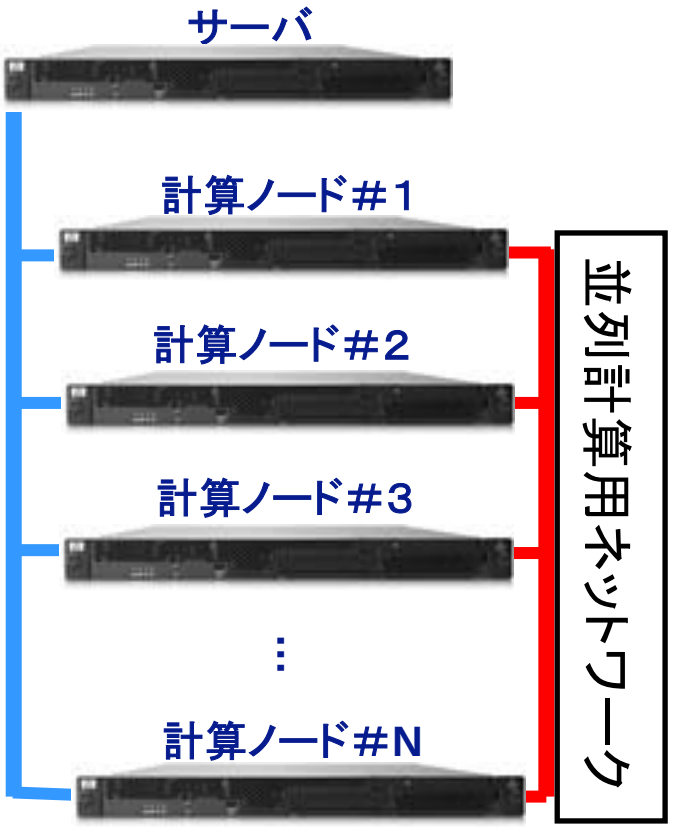
- 使ってみよう！
- SCore5.8.3 CDROMを/opt/SCORE5.8.3.CDROM
- README を読んで Install を実行
  - Install　は bininstall –server を　実行
  - サーバに必要な実行形式などをインストール
  - HTML　形式の　ドキュメントも展開されます
  - サーバマシンでCDを展開すると都合が良い
  - Fedora Core3のインストール時はFirewallとSELinux を無効にして下さい。
- WEB ブラウザを立ち上げドキュメントを開く…
  - Binary Module Extractionはサーバで実行
  - 実際には　./Install で　実行済み

SCore
入門

実習

Powered by SCore

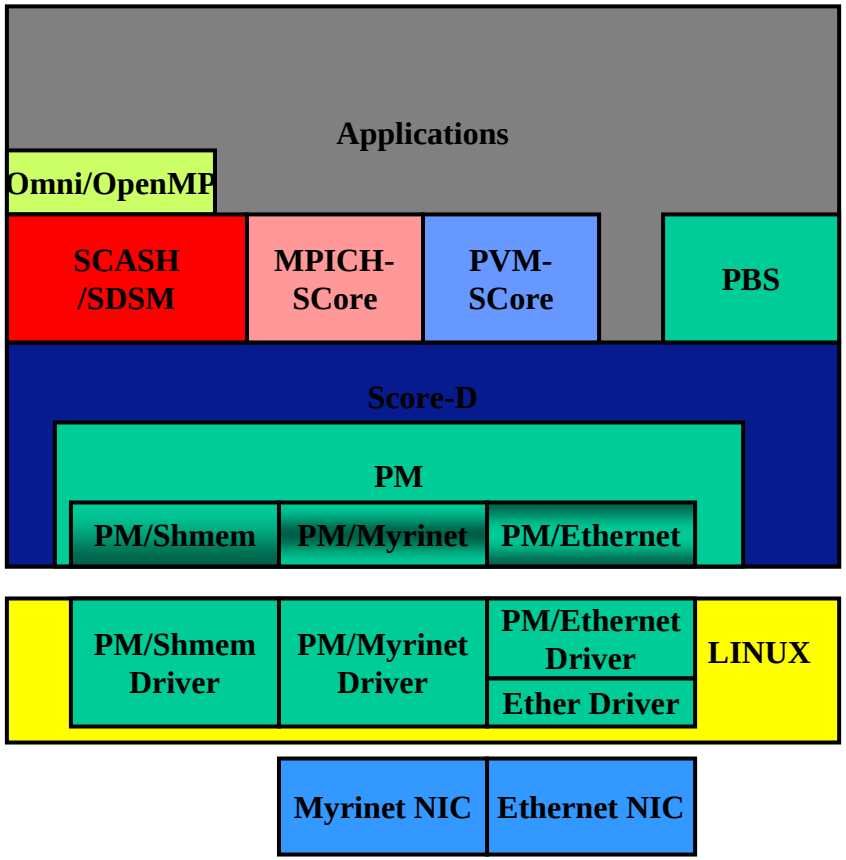
- ドキュメント通りに実行すれば、ほぼ間違いなくインストールできます
- msgb、scout 関連の動作は全てユーザレベルアプリの動作試験
  - ここまでは、ハードウェアを疑う必要はほとんどありません
  - ソフトウェア的な設定を確認して下さい
- デバイスドライバーが関連するのはPM（通信ライブラリ）の動作試験から



SCore 入門

SCore ソフトウェア構成

- プログラミング環境
  - MPICH-SCore
  - Omni/OpenMP on SCASH
  - PVM-SCore
- SCore-D
  - マルチユーザ環境
    - ギャングスケジューリング
    - 時分割空間分割スケジューリング
  - リソース管理
  - 実時間ロードモニタ
  - チェックポイントリスタート
  - デッドロック検出
- PBS
  - バッチスケジューラ
- PMv2通信ライブラリ
  - Myrinet
  - Ethernet
  - Infiniband
  - Shmem



SCore
入門

ノード間通信ライブラリ:PMv2

- 低遅延、高バンド幅
  - Myrinet-XP
  - Oneway latency 5μ弱
  - 220MBytes/sec 以上
- ユーザレベルでの実装
- ポーリングベースのAPI
- ゼロコピー通信:RDMA通信
- ヘテロなネットワーク環境に対応
  - Myrinet
  - Ethernet
  - InfiniBand

SCore
入門

PMv2：ユーザレベル通信

- システムコールのオーバーヘッドを削減する
- MmapシステムコールでMyrinet NICのレジスタ、SRAM空間をユーザ空間にマップ（通常のmapped IOはカーネル空間にマップ）
  - ユーザ空間からNICのレジスタ、RAM領域にアクセス可能
  - NICの操作（レジスタへの読み書き）、SRAM空間の操作をLoad/Store命令で行うことが可能になります
  - パケットの送受信をユーザレベルで実行できる
  - NICのデバドラの実装はprobe,open,mmap,closeがほとんどでread,writeは実装していません
- MyrinetXPの遅延時間は片道　5μsec程度

SCore
入門

PMv2：ゼロコピー通信

- Myrinetのバンド幅を生かしたい
  - 当時のMyrinetのバンド幅
    800MBps:100MBytes/sec
  - PCI:32Bitx33Mhz=132MBytes/sec
  - SBUSは実効30MBytes/sec+α　程度
  - 開発当初のPCではメモリのコピーバンド幅が
    100MBytes/secに及びませんでした
  - 通常のメッセージ通信では、送信側、受信側でそれぞれ1回、バッファからユーザ領域へコピーが行われる
  - 送受信バッファをコピーしていてはMyrinet,PCIのバンド幅を発揮できない
- 送信側のユーザ空間から直接受信側ユーザ空間へDMA転送をしてしまう

SCore
入門

PMv2：ゼロコピー通信

- MyrinetXP,2000でDMA転送を行うと220MBytes/sec以上
  - リンクスピードは片方向2Gbps:250MBytes/sec

- 現在のPCでは、メモリコピーのバンド幅が大幅に向上しています

- ピンダウンのコストが相対的に大きくなっている

- 転送サイズがnx100KBytes以下ならメッセージ通信の方が高バンド幅

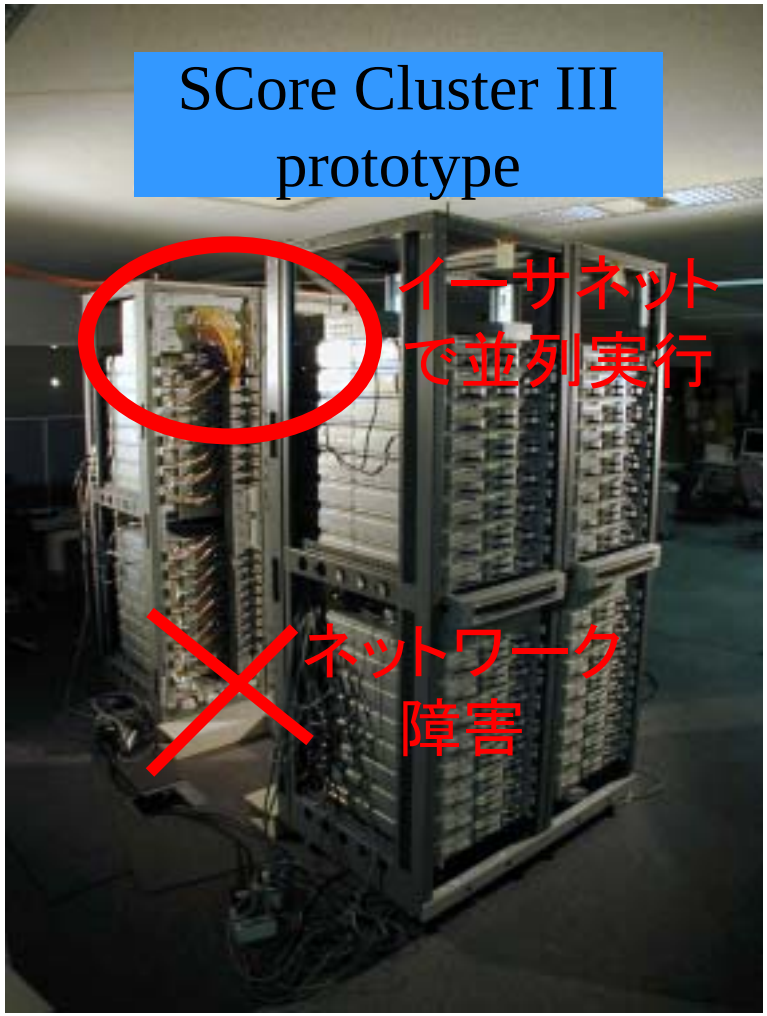
PMv2：複数通信メディアへの対応

- 元々はMyrinet専用の通信APIとして開発
  - 最初は複数の通信メディアへの対応は予定していませんでした。
- Ethernet
  - 大量生産品が高性能化するとコストパフォーマンスで有利
  - Trunking
    - 複数のNICをまとめて使うことが出来ます
    - GbE 2枚でほぼ2倍のバンド幅で通信できます
    - GbE 4枚まではバンド幅が向上することが確認されています
  - PM/Ethernet-nkpのpreliminaryを配布開始
    - サポートドライバーは　e1000, tg3
- InfiniBand
  - Fujitsu
  - TopSpin

SCore
入門

PMv2：複数通信メディアへの対応

·SCoreでは複数の通信メディアを1つのバイナリから使うことができます。つまり1つのa.out でMyrinetもEthernetも使うことができます
  ·% scrun −network=myrinet
  ·%scrun −network=gigaethernet
  ·%scrun −network=ethernet
·通信メディ別にa.outをlinkする必要はありません
  ·但しMPIはMPICHベースのためコンパイラごとにランタイムが存在します。。。。
•デフォルトのネットワークが故障してもa.outを再コンパイルせずにEthernet で並列処理を行う事が可能です。
·PMは通信メディアごとに実装されています。
  ·Type = myrinet,myrinet2k,myrinetxp,ethernet,etc



PMv2：その他の機能

- コンテキストのセーブ・レストア
  - ネットワークコンテキストとは、、、、
    - NICのレジスタ
    - 送受信バッファ
    - ピンダウンメモリのアドレス
    - ノード番号テーブル
    - 経路表
    - etc
  - ギャングスケジューリングはネットワークコンテキストを入れ替えることによって実現しています。

PMv2：その他の機能

- 初期化
- ルーティングテーブルの作成
- ノード番号の割り当て
- トランキング（PM/Ethernet）
- Barrier,lock等の同期機構は提供していません
- 詳しくは マニュアルのReference Guide に PM　APIの記述があります

SCore
入門

MPICH-SCore

ここからプログラミング環境

- MPICHは米国立アルゴンヌ研究所が開発したフリーソフトウェア
- MPI（Message Passing Interface)の実装の一つ
- MPICH-SCoreはMPICHをSCoreに移植
  - 通信部分をPMv2で実装
- マニュアルに ch_score2 の記述がありますが、現在は配布されていません。
- クラスタ上のほとんどのプログラムはMPIで記述されています。
- これからクラスタを導入される方は、、、
  - 既存のソフトウェア資産の移植等を考慮する必要があります
  - 特に日本はベクターユーザが多い…
  - 是非MPI環境に精通してください
  - MPIに精通したパートナーさんを探す

MPICH-SCore

- コンパイラ
  - mpicc、mpic++、mpif77、mpif90
  - Intel,PGI,Fujitsu,Pathscaleコンパイラを使用することも可能です
    - %mpicc –compiler intel ……
  - デフォルトコンパイラはGNUコンパイラ
  - デフォルトコンパイラは設定ファイルを変更することで変更できます
  - mpif90については、GNUのF90コンパイラがないので、別途上記コンパイラを購入する必要があります
    - gcc4.0以降はf90 に対応している

SCore
入門

MPICH-SCore

通信プロトコル切り替え

- Short protocol
  - MPIヘッダーとメッセージデータを1つのパケットで送信する
- Eager Protocol
  - MPIヘッダー＋複数のメッセージデータパケットで送信
  - 受信側が受信待ち状態になっていなくともメッセージを送信します
  - 受信待ち状態にない場合、受信側はメッセージを受信バッファに一時的にコピーします
- Randevous Protocol
  - MPIヘッダー＋複数のメッセージデータパケットで送信
  - 受信側が受信待ち状態になってからメッセージを送信します
  - 受信側はアプリケーション領域にコピー可能
  - 受信側でメッセージのコピーを削減
  - 受信待ち状態を送信側に通知する必要があります
- Short/Eagerの切り替えは 1KBytes　固定
  - 変更するには、ソース書き換え、リコンパイル
- Eager/Randevousの切り替えは16KBytes
  - % scrun –nodes=8,mpi_eager=40960

SCore
入門

MPICH-SCore

- PMv2のRMA通信機能をMPICH-SCoreから使うことが出来ます
  - % mpirun –np 8 –score mpi_zerocopy=on ./mpi_app
  - % scrun –nodes=8,mpi_zerocopy=on ./mpi_app
- ゼロコピー通信はRandevous プロトコルにのみ使われます
  - 1KBytes以下のメッセージ（Shortプロトコル）には適用されません
  - 上記の例では16KBytes以上（Eager/Randevous切り替え)にのみ適用されます
- プロトコル切り替え・ゼロコピー通信はPM/Myrinetで効果を発揮する
- PM/Ethernetはmaxnsend,backoff等PM/Ethernetのパラメータチューニングが効果的
- PM/Ethernetはトランキングも効果的

MPICH-SCore

- 通信ログ
  - % mpicc –mpilog –o hello hello.c
- 視覚化ツール
  - Upshot
    - Alog 形式
    - % setenv MPE_LOG_FORMAT=ALOG
    - % Scrun ….
    - % logviewer hello.alog
  - Jumpshot3
    - Slog 形式
    - % setenv MPE_LOG_FORMAT=SLOG
    - % scrun ….
    - %logviewer hello.slog
- 通信の視覚化
  - % mpicc –mpianim –o hello hello.c –LX11 –lm –L/usr/X11R6/lib

SCore
入門

Omni OpenMP

- Cluster Enabled OpenMP
  - クラスタ計算機上で共有メモリプログラム環境であるOpenMPプログラムを動作させることが出来ます
  - クラスタ計算機上での共有メモリプログラミング環境としてSCASHを用いています
- コンパイラ
  - omcc
  - omf77
  - % omcc –omniconfig=scash –o laplace laplace.c
  - % scrun ./laplace

SCore
入門

デバッガ

- アプリケーション実行中の例外発生時にデバッガを起動することが出来ます
- 環境変数　DISPLAY　を設定して下さい
- % scrun –nodes=8,debug ./hello

SCore
入門

並列プロセスの起動

- SCoreでは scrun コマンドで並列プロセスを起動します。
- MPICH/SCore では mpirun コマンドも提供されています。
  - MPICH/SCore のmpirun コマンドでは、引数“-score”で score オプションを指定します
  - % mpirun –np 4 –score monitor=load　a.out
- scrun/mpirun コマンドを実行すると、全てのノード計算機上に、実行ファイルがコピーされます。(SCORE OPTIONでコピーを抑制することも可能)

- 他のMPI環境（mpich等）の mpirun コマンドのように、全てのノードで実行ファイルを共有する必要はありません。

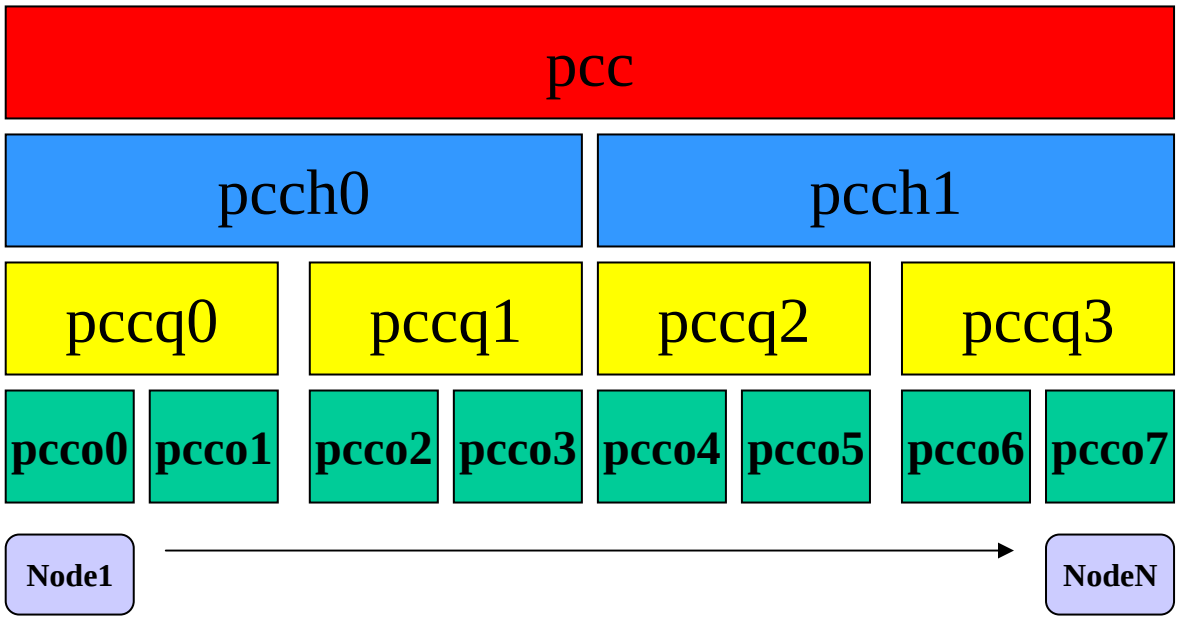
- ファイルサーバがなくても、クラスタを構築することができます。
  - 大規模クラスタを構築する場合に有利

- クラスタ計算機の規模拡大、ディスクの大容量化への対応は今後の課題の一つ

SCore
入門

ノードのグループ管理

- SCoreでは 複数のノードをまとめてグループとして管理できます。
- % scrun –group=pcch0,nodes=4x2,network=ether a.out
- % scorehosts –g pcch0



SCore
入門

実行環境

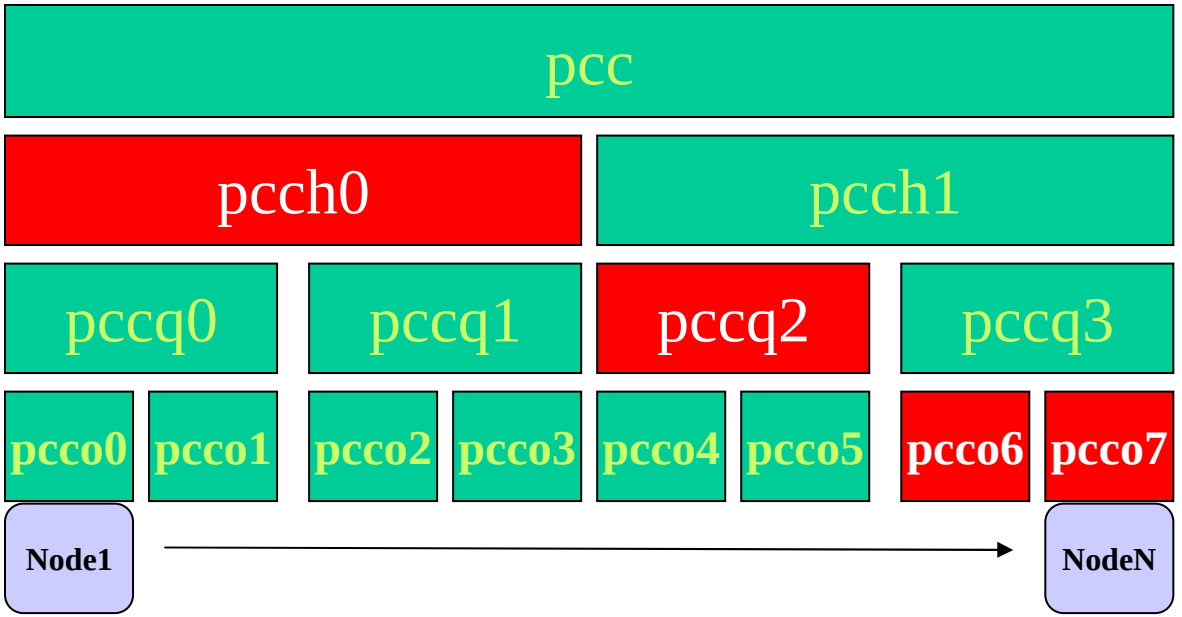
- シングルユーザ環境
  - クラスタ環境の全体または一部を排他的に使用する
- マルチユーザ環境
  - SCoreが目指した実行環境
  - 複数ユーザの複数プログラムを同時処理
  - ギャングスケジューリングによる時分割処理
  - 空間分割
  - あまり使われていないという話も聞きますが、クラスタ計算機が身近になるにつれて普及してほしい

SCore
入門

シングルユーザ環境

- 通常のクラスタ計算機の実行環境
- クラスタの全体又は一部を排他的に使用する
- 1つのプロセッサグループで1つだけジョブを起動する
  - % scout –g pcc
  - % scrun –nodes=8 ./infinite-loop
- プロセッサグループ　pcc 上では他のSCoreプロセスを起動できない
  - %scrun –nodes=8 ./hello

  SCOUT:Failed to lock Messageboard
- 他人数での対話型環境には不向き
- 以下のように直接プログラムを起動することも出来ます
  - %scout –g pcc –e scrun –nodes=2x2 a.out



SCore
入門

マルチユーザ環境

- 同時に複数のプログラムを実行可能

```
% scout –g pcc
% scored –server  nodeX
SYSLOG:  ++++++++++++++++++
SYSLOG: -----SCore-D (5.8.3) bootup ----
```

```
% mpirun –np 32 –score scored=nodeX yellow
```

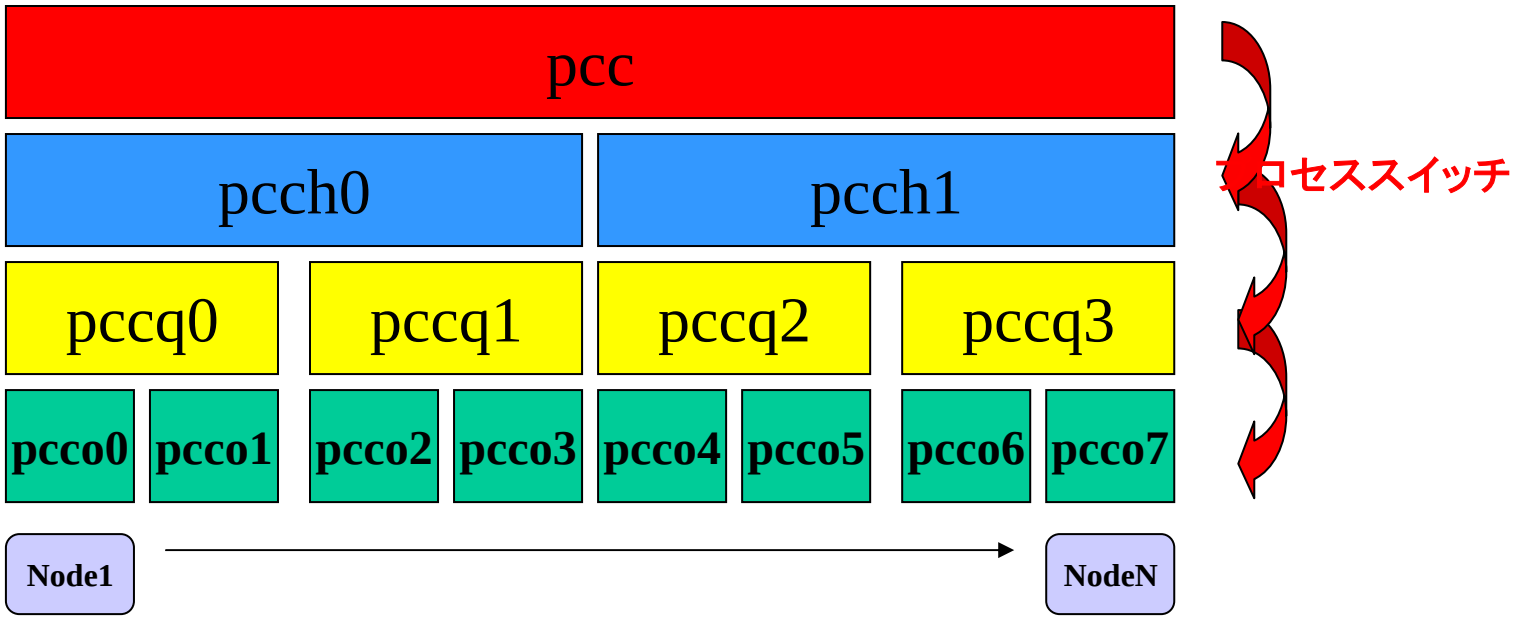
```
% scrun –nodes=8x2,scored=nodeX    green
```

```
% export SCORE_OPTIONS=scored=nodeX
% scrun –nodes=16x2,checkpoint=8h    red
```

```
% sc_console  nodeX –c   shutdown
```

マルチユーザ環境

- タイムシェアリング環境のようにクラスタ計算機を使用できる
- 複数ユーザの複数ジョブを同時処理
- 時分割
- 空間分割



SCore-D

- **SCore-D のモニタリング**

```
log_Server %    scbcast  syslog
```

```
server % sc_syslog   log_server  /var/messages/scored
server %    scout –g pcc –e scored syslog   log_server
```

```
  server  %  cat /var/messages/scored
……..
……..
```

SCore-D

- **SCore-Dの自動運転**

```
server %  sc_watch –g pcc scored
```

SCore
入門

SCoreのノード障害対応（1）

- /opt/score/etc/scorehosts.defectsに記述されたノードには、ジョブが割り当てられなくなります。
- 上記のファイルに障害が発生したノードを記述することによって縮退運転を行うことが出来ます

% scorehosts –g pccg0
node0 node1 node 2 node4
① /opt/score/etc/scorehosts.defects に　node1 を記述
% sceptic –g pcc
…… 障害ノードの検出
② /etc/init.d/scoreboard reload
% scorehosts –g pccg0
node0 node2 node3

SCore
入門

SCoreのノード障害対応（2）

- /opt/score/etc/scorehosts.db の各ノードに対して、スペアノードを記述することができます。
- /etc/score/etc/scorehosts.defects に記述されたノードに対しては、スペアノードにジョブが割り当てられます。
- 予めホットスペアノードを準備することによって、ノードに障害が発生しても、被害を最小限に食い止めることが出来ます。

% scorehosts –g pccg0
node0 node1 node 2 node4
① /opt/score/etc/scorehosts.defects に　node1 を記述
② /etc/init.d/scoreboard reload
% scorehosts –g pccg0
node0 sparenode node2 node3

SCore
入門

SCore-D

- チェックポイント・リスタート
  - 実行中のプログラムをファイルに退避
  - 退避したプログラムを再実行
- % scrun –checkpoint=10m ./hello
  - -checkpoint　オプションでチェックポイント間隔を指定
  - チェックポイント間隔毎にプログラムが退避されます
  - 時間指定は sS,mM,hH,dDを指定可能
- %scored　–restart
  - -restartオプション付きでscoredを起動すると退避されたプログラムが再起動されます

SCore
入門

バッチ環境

- PBSバッチ環境
- qsub, sc_qsub コマンドによってジョブを投入
  - % qsub job.sh
- qmgr,qstat,pbsnodesなど通常のPBSのコマンド郡を使用可能
- しかし、、、版が古い
  - 1999年 PBS2.2
- 現在は TORQUE,MAUIか？

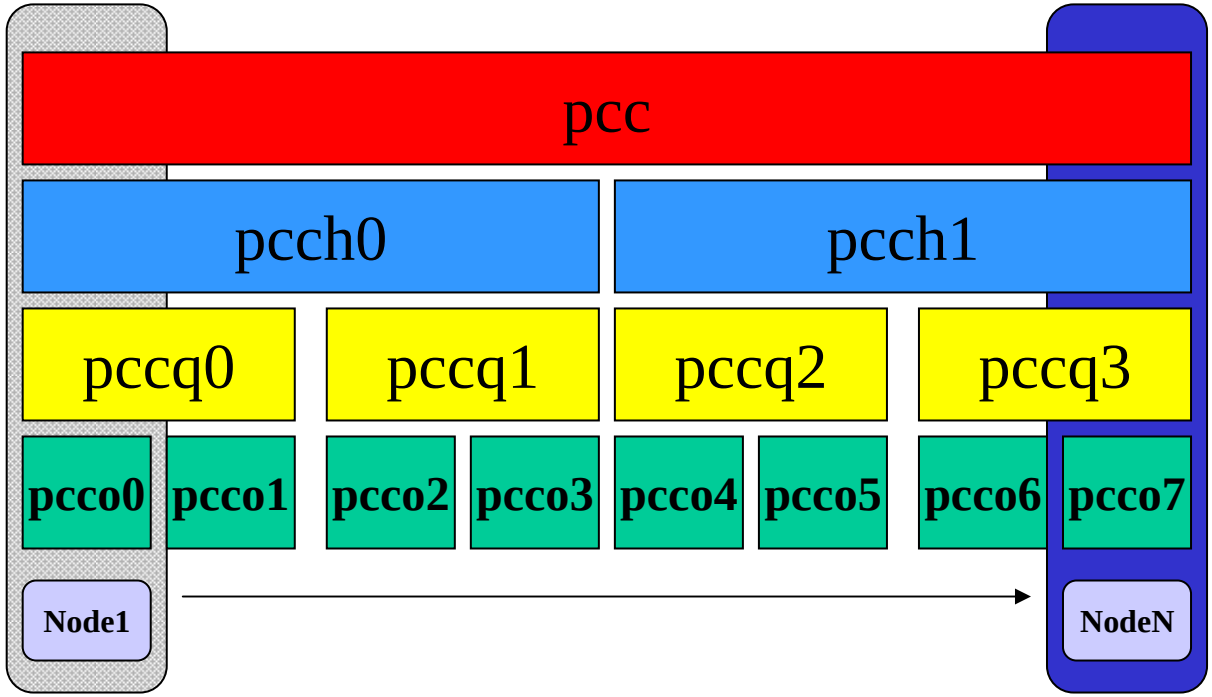
```
Sample_job.sh
#!/bin/sh
#PBS –j oe
#PBS –l nodes=4:score
#PBS –q queue
Scout –wait –F ${PBS_NODEFILE} –e scrun –nodes=8x1 job
```

SCore 入門

PBS の導入（1）

Powered by SCore

- PBSのノードプロパティ
  - SCoreのノードグループに対応
  - PBSと SCoreで同じグループ名で参照可能になります




```
# qmgr : set node node1 properties="pcc,pcch0,pccq0,pcco0"
# qmgr : set node nodeN properties="pcc,pcch1,pccq3,pcco7"
```

SCore
入門

PBS の導入（2）

Powered by SCore

- PBSのTimeShared ノード
  - SCoreのログインノードを指定
  - ジョブをキックするノードにできる
  - # qmgr : set node login01 ntype=time-shared
  - # qmgr : set node login02 ntype=time-shared
- PBSのClusterノード
  - 計算ノードに対応
  - # qmgr: set node score01 ntype=cluster
  - # qmgr: set node score02 ntype=cluster
  - # qmgr: set node score03 ntype=cluster

SCore
入門

PBS の導入（3）

- PBSの実行スクリプトで指定するノードグループを以下のように指定
  - ログインノード ＋ 計算ノード
  - ログインノード ＋ 計算ノードのプロパティ

Sample_job.sh

```
#!/bin/sh
#PBS –j oe
#PBS –l nodes=1:login+4:score
#PBS –q queue
 scout  -F $PBS_NODEFILE –e  scrun  -nodes=4x1 job
```

SCore
入門

PBS の導入（4）

Powered by SCore

- ログインノードから常にジョブが起動される
- Scrun は　ログインノードから起動される
- Scrun コマンドを実行すると、全てのノード計算機上に、実行ファイルがコピーされます。
- 他のMPI環境（mpich等）の mpirun コマンドのように、全てのノードで実行ファイルを共有する必要はありません。
- SCoreでは　a.out を　計算ノード間で共有する必要がない
- A.out 等、ジョブの実行に必要なファイルはログインノードのみで共有すれば良い
- 大規模なファイルサーバ無しでも、大規模並列クラスタを構築可能

SCore
入門

SCORE OPTION

- SCore では　コマンド引数を SCORE OPTION で指定します
- 環境変数　SCORE_OPTIONSでも指定可能
- 　scrun コマンドでは以下のように指定
  - # scrun –group=pcc,network=etherx2,nodes=4x2
- 　mpirun コマンドでは以下のように –score に続けて指定
  - % mpirun –np 4 –score network=etherx2,monitor=load
- SCoreオプションはランタイムライブラリが個別に評価します

SCore
入門

SCORE OPTION

- group
- nodes
- network
- priority
- monitor
- debug
- statistics
- scored
- file
- restart
- checkpoint
- cpulimit
- memorylimit
- disklimit
- wait
- message
- resource
- passhup
- ts
- nobincopy
- checkpoint
- mpi_eager
- mpi_zerocopy

SCore
入門

SCoreの導入

SCoreに限らずクラスタ計算機の導入にあたって以下を考慮しなければなりません

- 運用形態
  - バッチ、対話型
  - ユーザ管理（NIS？）
  - ファイルの共有
    - ホームノード、共有データの管理
    - ネットワークファイルシステム
    - クラスタファイルシステム
- ソフトウェア資産の継承
  - 稼動中のアプリケーション、システム
  - 分散メモリ並列プログラミング（MPI）への対応
    - 計算能力、メモリバンド幅、ノード間通信性能、etc
- クラスタの構成
  - ノード（プロセッサ）、メモリ、ディスク
  - ネットワーク
- 特に台数規模が大きくなると
  - 故障検出、診断
  - コンソール管理
    - VGAコンソール、シリアルコンソール
  - ソフトウェア管理（システムの版管理、バージョンアップ）
  - 運用/保守人員の確保

SCore
入門

SCore導入

パイロットシステムを構築してほしい

- 32ノード以下程度
- コストを削減するには、、、、
  - 余っているPCを寄せ集める（Beowulf方式）
    - x86は非常のコストパフォーマンスが高い
  - ネットワークはイーサネット（トランキング）
    - ギガイーサのスイッチが廉価
- 場所がない、、、
  - ノートPC
  - Bladeのような高密度実装マシン
- いいパートナーを見つける
  - クラスタの構築、導入、運用、管理には、ハードウェアからシステムソフト、アプリケーションまで幅広い技術が要求されます。