FUJITSU

PACS-CSシステム ソフトウェアについて

住元 真司

富士通研究所

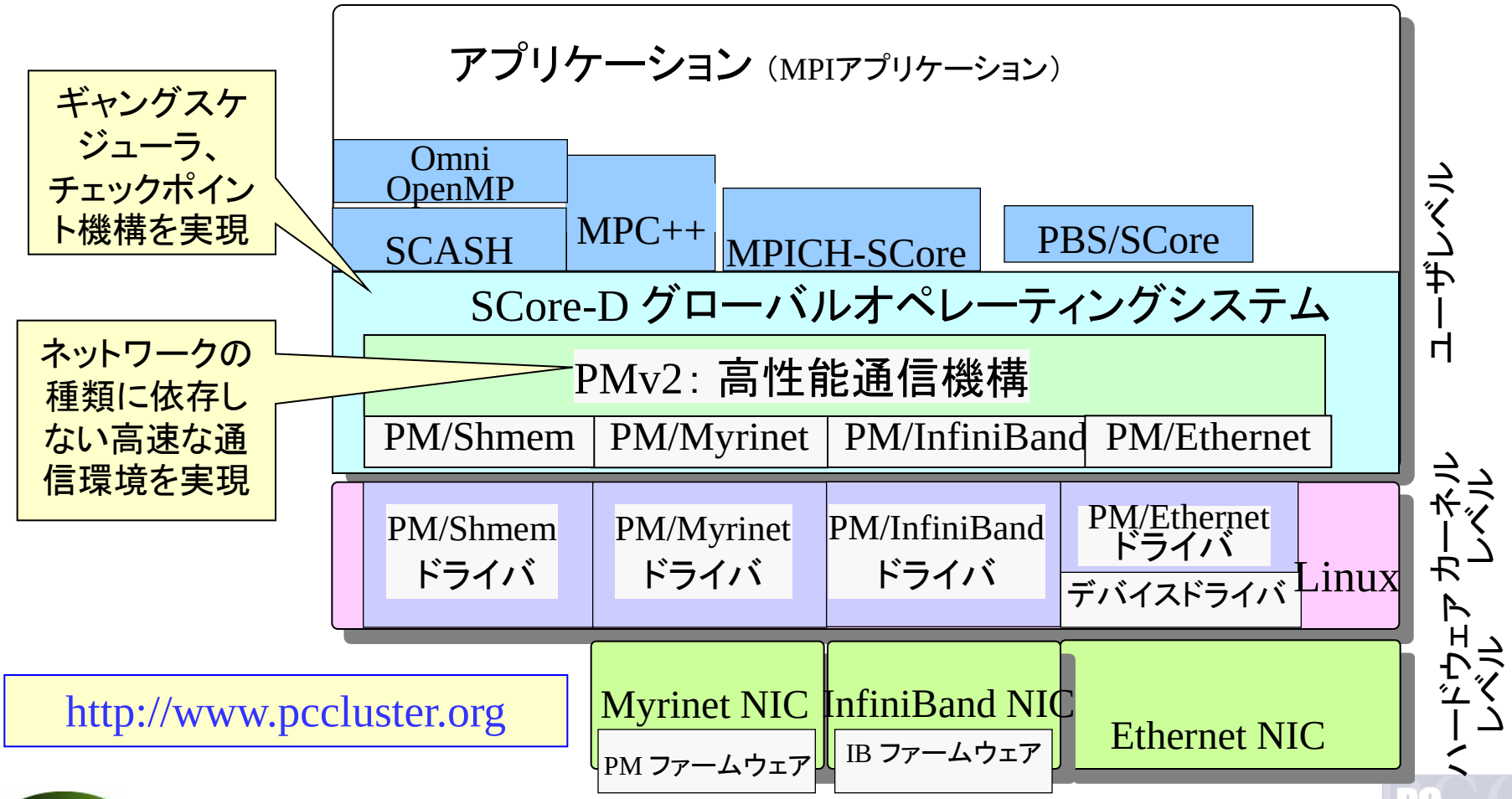
PC Cluster Consortium

発表の概要

- SCoreクラスタシステムソフトウェア&PMv2
- PACS-CSのための高性能通信機構 PM/Ethernet-HXB：
  - 高バンド幅，軽量通信プロトコルの設計
  - 評価
    - 低レベル（PMv2）通信性能
    - MPIレベル通信性能
    - アプリケーション性能
- まとめ

SCoreクラスタシステムソフトウェア

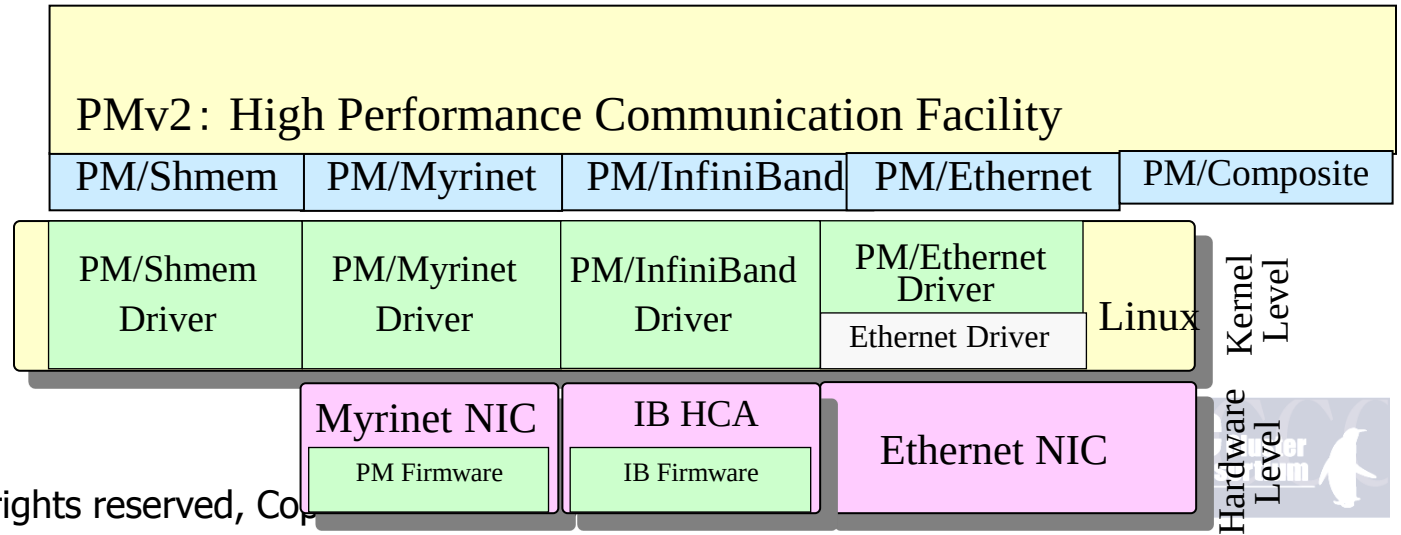
- 1990年代にRWCプロジェクトで開発



http://www.pccluster.org

PMv2通信機構の概要

- 既存OSの通信プロトコル処理オーバヘッドを取り除き高い通信性能（高バンド幅，低遅延）を実現するために開発された
- 開発当初のPMはMyrinet専用であったが、PMv2から複数種類のネットワークをサポートしている
  - PM/Myrinet, PM/Ethrernet, PM/Shmem, PM/SCI, PM/UDP(Agent),PM/InfiniBand (-FJ[Fujitsu], -TS[TopSpin])



PM Development History

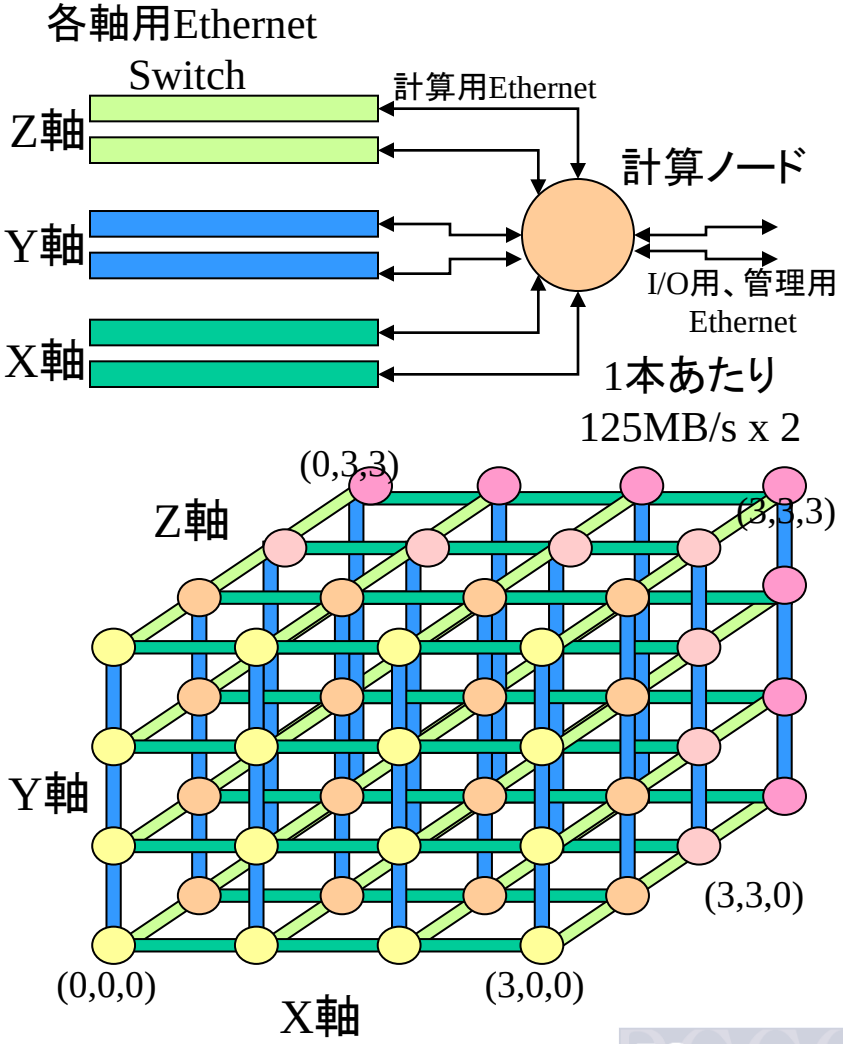
- PMは1995年手塚氏（当時RWCP）により開発
  - Myrinet専用, Sun SS20s, SBUS, SunOS 4.1.x
- 1996年、Pentium(NetBSD)上に移植された
- 1997年、Linuxに移植、i386, Alpha(SCore 2.x)
- 1998年、GigaE PM開発
  - Gigabit Ethernet上のPM
- 1999年、異種ネットワーク対応(PMv2,SCore 3.0)
  - Ethernet(NT), Shmem, Myirnet, UDP
- 2001年、SCI, Myrinet 2000サポート
- 2003年、InfiniBand, Myrinet XP(2XP)サポート
- 2006年、PACS-CS向けPM/Ethernet-HXB開発
- 2009年、PMX, Kernel変更不要のドライバ

PMv2 高性能通信のための技術

- 信頼性と順序性を確保したデータグラム通信: PMv2
  - TCP/IP等コネクションベースの通信は大規模クラスタには適さない
- ユーザレベル通信: PM/Myrinet, PM/InfiniBand
  - 低遅延ネットワークのためにシステムコールのオーバヘッド削減
- Zero-Copy 通信: PM/Myrinet, PM/InfiniBand
  - ホストCPUのコピーオーバヘッド、メモリバンド幅を節約し、高い通信バンド幅性能を実現
- Network Trunking : PM/Ethernet
  - 複数のEhternet NICを用いて高いバンド幅性能を実現
- カーネルレベルの RMA: PM/Ethernet-kRMA
  - 送受信の同期とコピーオーバヘッド削減で高い通信バンド幅を実現
- Ethernetを用いたZero-Copy通信： PM/Ethernet-HXB
  - 複数のEthernet NICを用い、かつ、Zero-Copy通信を実現

PACS-CSのための通信機構の開発

- Motivation: Commodity利用の究極の通信機構の実現
  - 専用インターコネクトにどれだけ迫れるか？
- 目標: InfiniBandに迫る通信性能
  - ネットワーク性能を最大限に引き出す: GigaE x 6 で 片方向750MB/s (双方向 1.5GB/s)
  - 次元間のルーティング処理を高速に処理する
- 課題:
  - 高バンド幅、低遅延通信を無駄なく高速にホストプロセッサで実現 する
- これを実現する通信機構 PM/Ethernet-HXBを開発



4x4x4の構成例

PM/Ethernet-HXBの設計目標

- PACS-CSのネットワークを最大限引き出す通信機構を実現したい
- そのためには、入出力パケットを滞りなく処理できることが必須
- 目標：パケット処理を到着間隔以内で処理
  - 片方向　10.9 usec以内
  - 双方向　5.5 usec以内

（8192Bメッセージ）

5.5 usec以内のパケット処理時間の実現 が目標

最大通信バンド幅時の
パケット到着時間間隔(usec)：片方向

ネットワークバンド幅	1500B	4096B	8192B
125 MB/s	12.0	32.8	65.5
250 MB/s	6.0	16.4	32.8
500 MB/s	3.0	8.2	16.4
750 MB/s	2.0	5.5	10.9
1000 MB/s	1.5	4.1	8.2
1250 MB/s	1.2	3.3	6.6

PM/Ethernet-HXBの設計課題

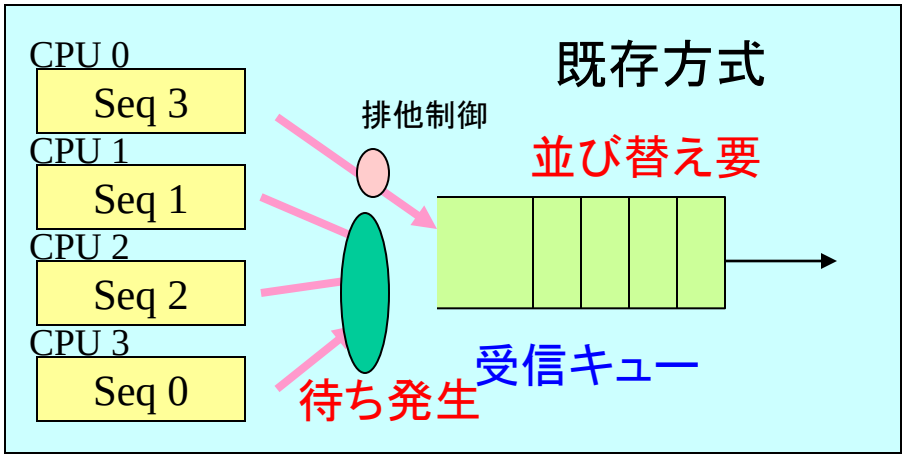
- 複数ネットワークインターフェイスカード（NIC）を用いた軽量通信プロトコル
  - 排他制御不要のPacket Ordering方式
- 通信バッファ間でのZero-Copy実現
  - ホストプロセッサデータコピー排除による
    コピー処理時間の削減
- 高速ルーティング処理の実現

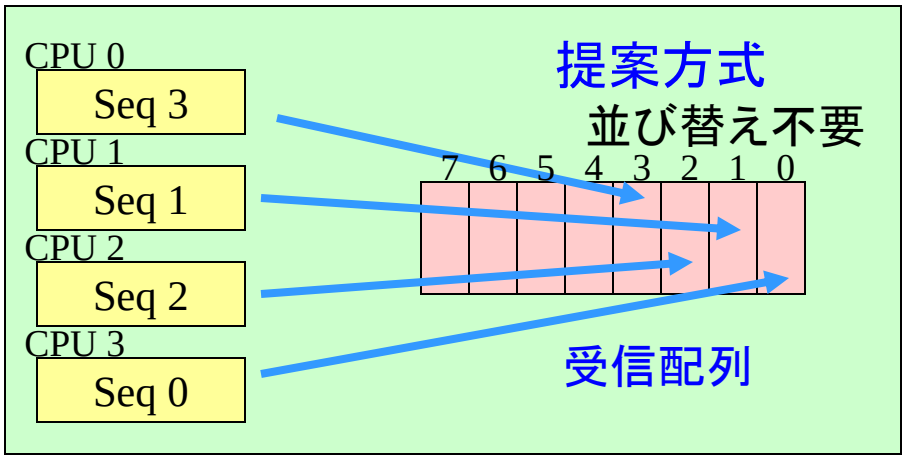
以降、この3つの点について説明

5.5usecのパケット処理時間実現には、
1 usec以下の小さな処理コストの把握と削減が必須

排他制御不要のPacket Ordering方式

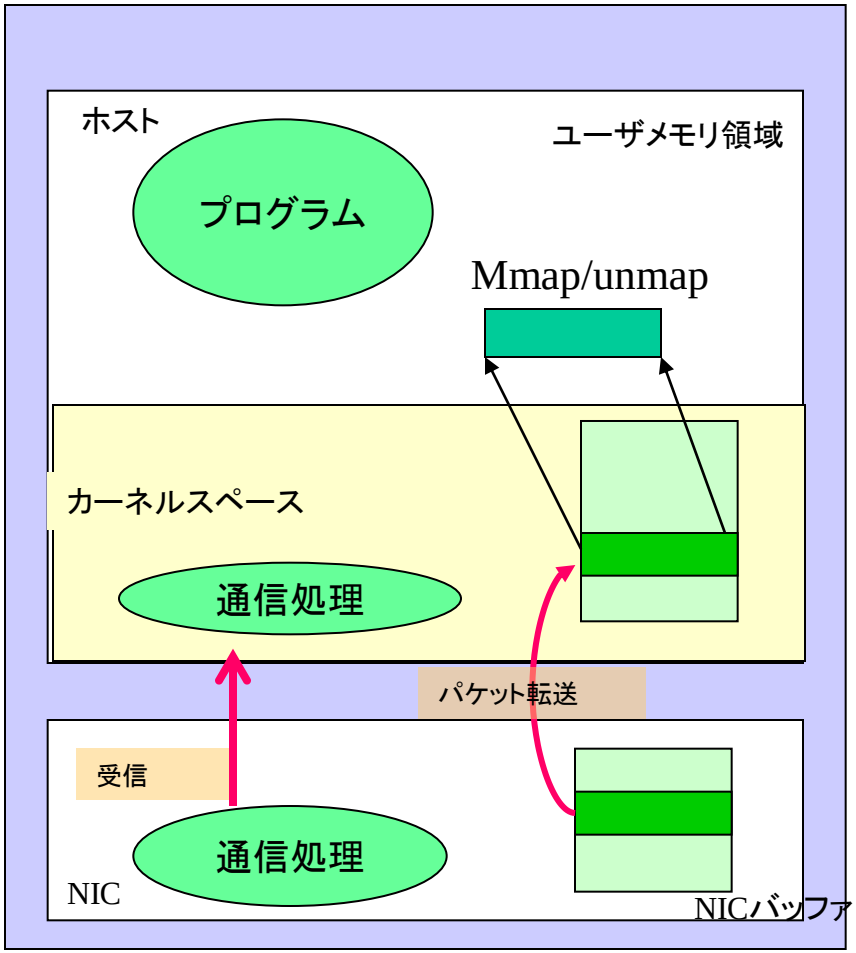
- 同じパケット番号(Seq番号)が同時に到着しない性質を利用
  - 相手先ごとに配列を準備、Seq番号をindexに配列アクセス
    - 0の場合は受信可能
    - 0以外は受信済み
  - 受信可能範囲チェックのみ
- 利点：排他制御不要、並び替え不要
- 欠点：メモリが多く必要
  - 既存：ポインタ2つとロック領域として12バイト
  - 提案方式：配列の大きさに依存 8つとして32バイト





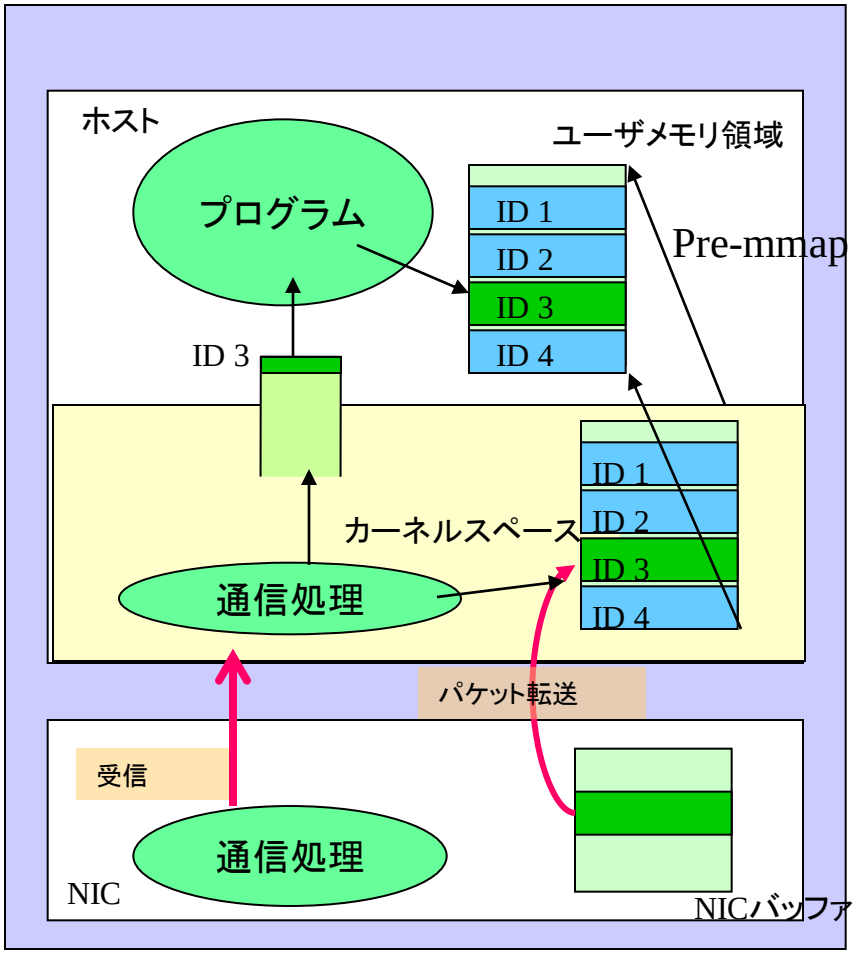
通信バッファ間Zero-Copy通信の実現

- 一般的な受信側Zero-Copy実現方式
  - 受信後にしか相手先がわからない
  - 受信後にユーザ空間にmap、受信処理後unmap
- 欠点：オーバヘッド大
  - システムコール2回必要
  - Mmapのコストがページあたり0.064usecと大きい
  - Map/unmap　9KBで0.384 usec



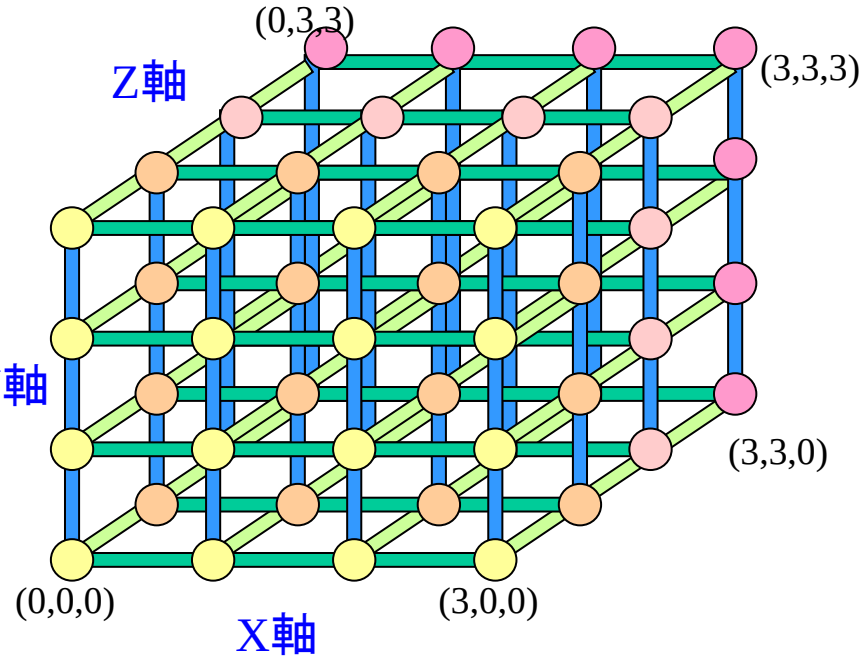
受信skbuf pre-map方式

- 受信用skbufを予めmmapしてmap/ unmapのオーバヘッドを排除
  - 各skbufにIDを設定しユーザ領域のmapアドレスと関連付
  - パケット受信時に受信skbufのIDをプログラムに通知
  - プログラムはIDより予めmapされたskbufを参照
- 利点：コピー排除、システムコール排除
- 欠点：
  - 受信skbをmapするための仮想空間が必要
  - 受信skbの固定化必須



高速ルーティング処理

- ルーティングアルゴリズムはCP-PACSと同様
  - X軸→Y軸→Z軸の順に転送軸の順序を固定
- ルーティング処理高速化のため受信skbをそのままEthernetヘッダのみ更新して送信
  - パケットヘッダに送信元、送信先の座標を格納
  - 中継ノードは自分の座標から送信先の座標に対してルーティング処理を実行
- ヘッダ書き換えとデバイスドライバ送信処理のみのオーバヘッド

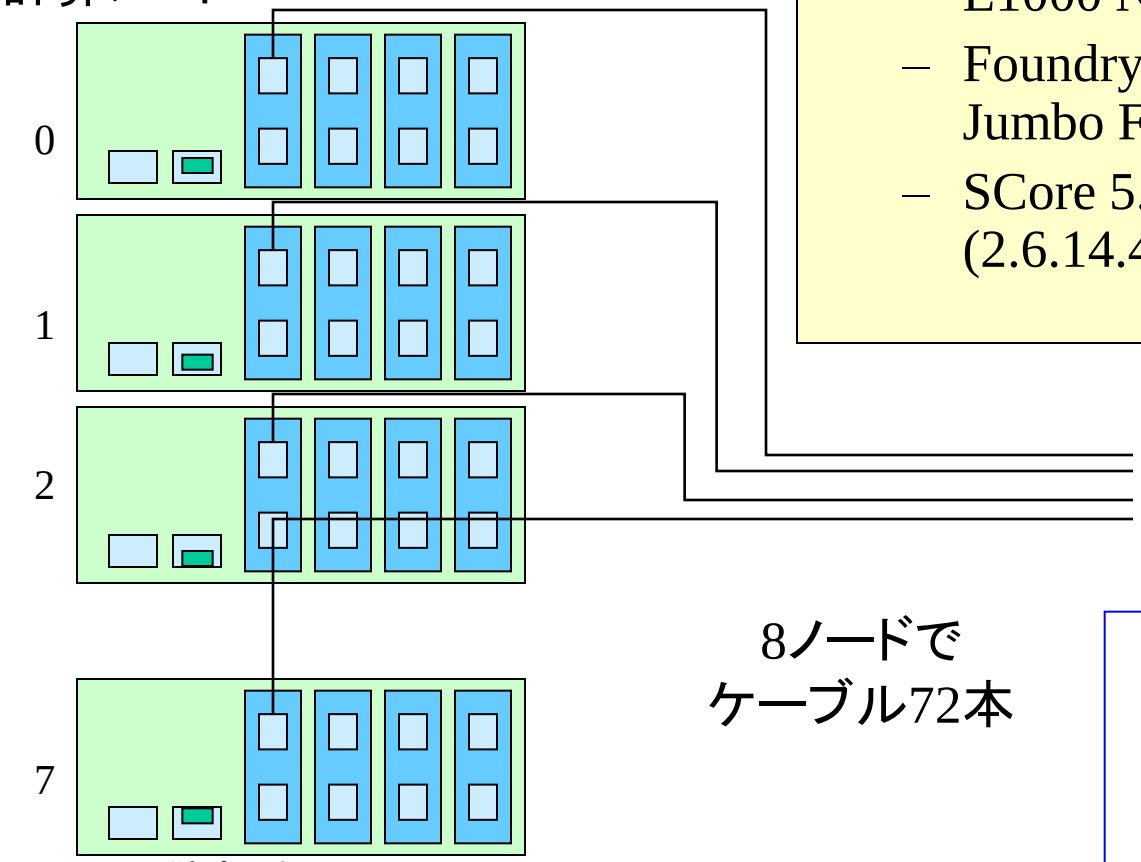


4x4x4の構成例

評価環境

8ノードPCクラスタ

計算ノード



独立したPCI-X 100MHz
バスに接続

- 計算ノード：Xeon 3.6GHz, FSB 800MHz, Intel E7520搭載、Dual E1000 NIC x 4 + Onborad E1000
- Foundry FWFX 448 GigE Switch Jumbo Frame利用
- SCore 5.8.3, Fedora Core-3 EM64T (2.6.14.4 UP Kernel)

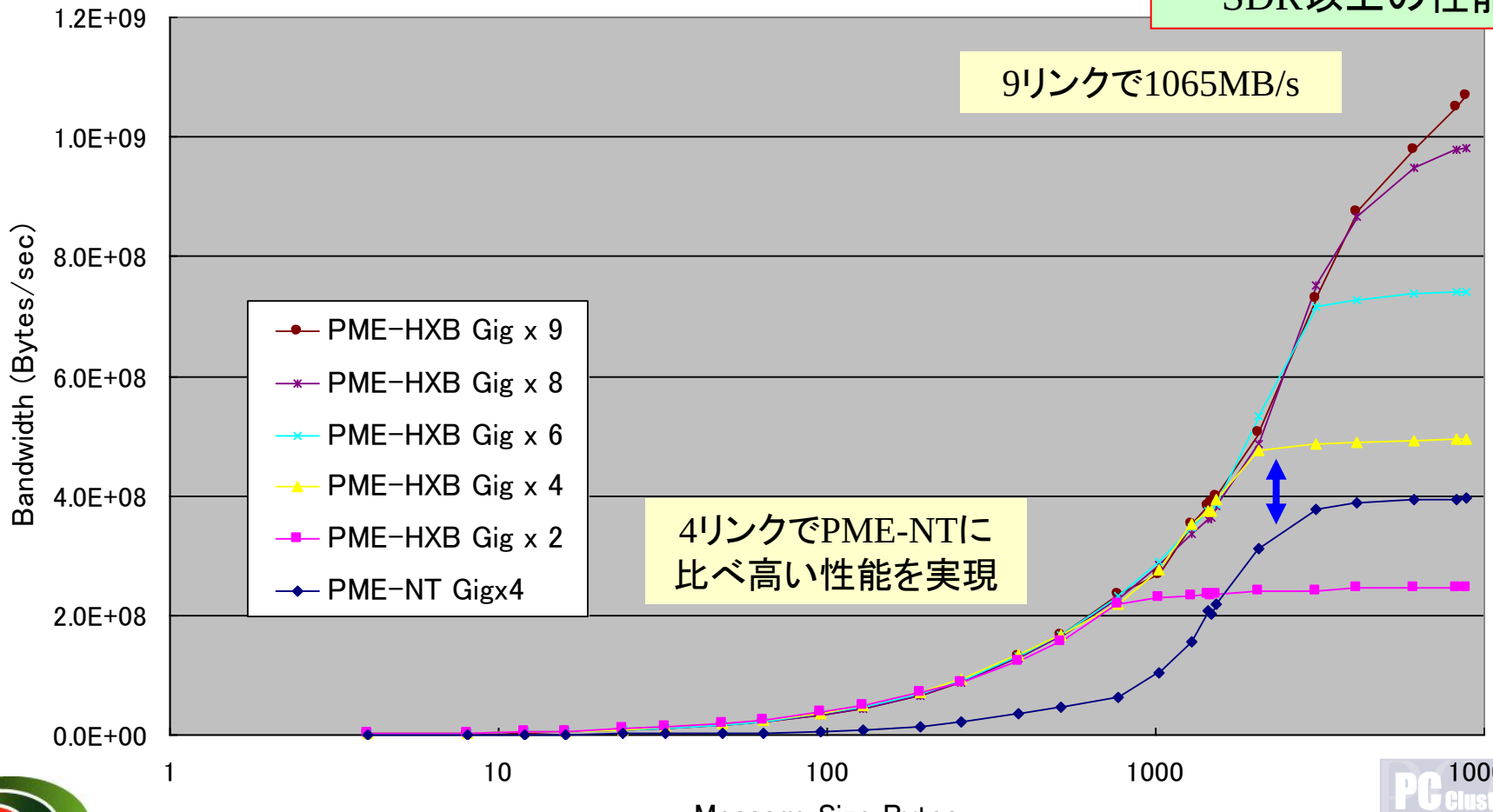
8ノードで
ケーブル72本

PACS-CSシステムとの違い
プロセッサ周波数
Ethernet SWの機種
（チップセットは同じ）

PMレベル通信性能：1次元片方向

- スケーラブルな通信性能を実現

当時のInfiniBand x4 SDR以上の性能



PMバンド幅通信性能と処理時間：1次元

- PM-Ethernet-HXB(PME-HXB)はGig x 9まで高いバンド幅スケーラビリティを実現
- パケット処理時間についても目標を達成

1次元	PMバンド幅	処理時間
PME-HXB Gigx2	248 MB/s	2.38 usec
PME-HXB Gigx4	494 MB/s	2.58 usec
PME-HXB Gigx6	741 MB/s	2.71 usec
PME-HXB Gigx8	981 MB/s	2.81 usec
PME-HXB Gigx9	1065 MB/s	2.90 usec
PME-NT Gigx4	393 MB/s	15.7 usec

2.8GHz換算で 3.5 usec

2.8GHz換算で 3.7 usec

FUJITSU

通信性能:ルーティング処理

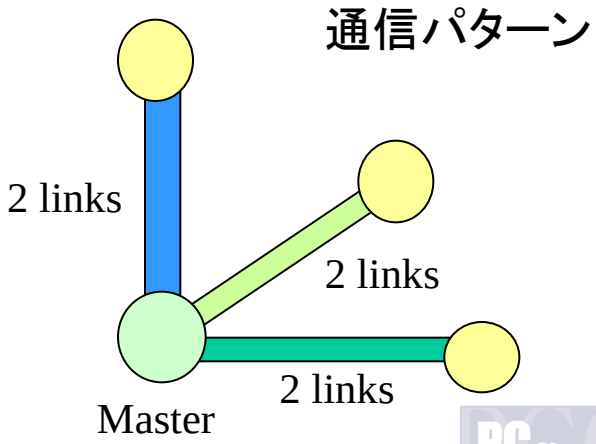
	Giga x 2 バンド幅	Giga x 4 バンド幅	Giga x 2 1/2 RTT
1次元	248 MB/s	494 MB/s	14.9 usec
2次元	248 MB/s	489 MB/s	27.5 usec
3次元	248 MB/s	--	39.3 usec

- 通信バンド幅: 理論性能の 97%以上
  - Giga x 2: 3次元で 劣化なし
  - Giga x 4: 2次元で 1%の劣化
- 次元あたり通信遅延: 12 usec (Switch 5usec)
- ルーティング遅延: 6.9 usec (割り込み 5usec)

隣接通信性能: 多次元

	送信 MB/s (%)	受信 MB/s (%)	送受信 MB/s (%)
1次元	247.6 (99.0)	247.5 (99.0)	481.1 (96.2)
2次元	493.6 (98.7)	494.9 (99.0)	951.3 (95.1)
3次元	741.3 (98.8)	742.3 (99.0)	1401.3 (93.4)

- 4ノードを用い3次元クロスバー通信性能を評価
- PMレベルでは93.4%以上の実効性能を実現



FUJITSU

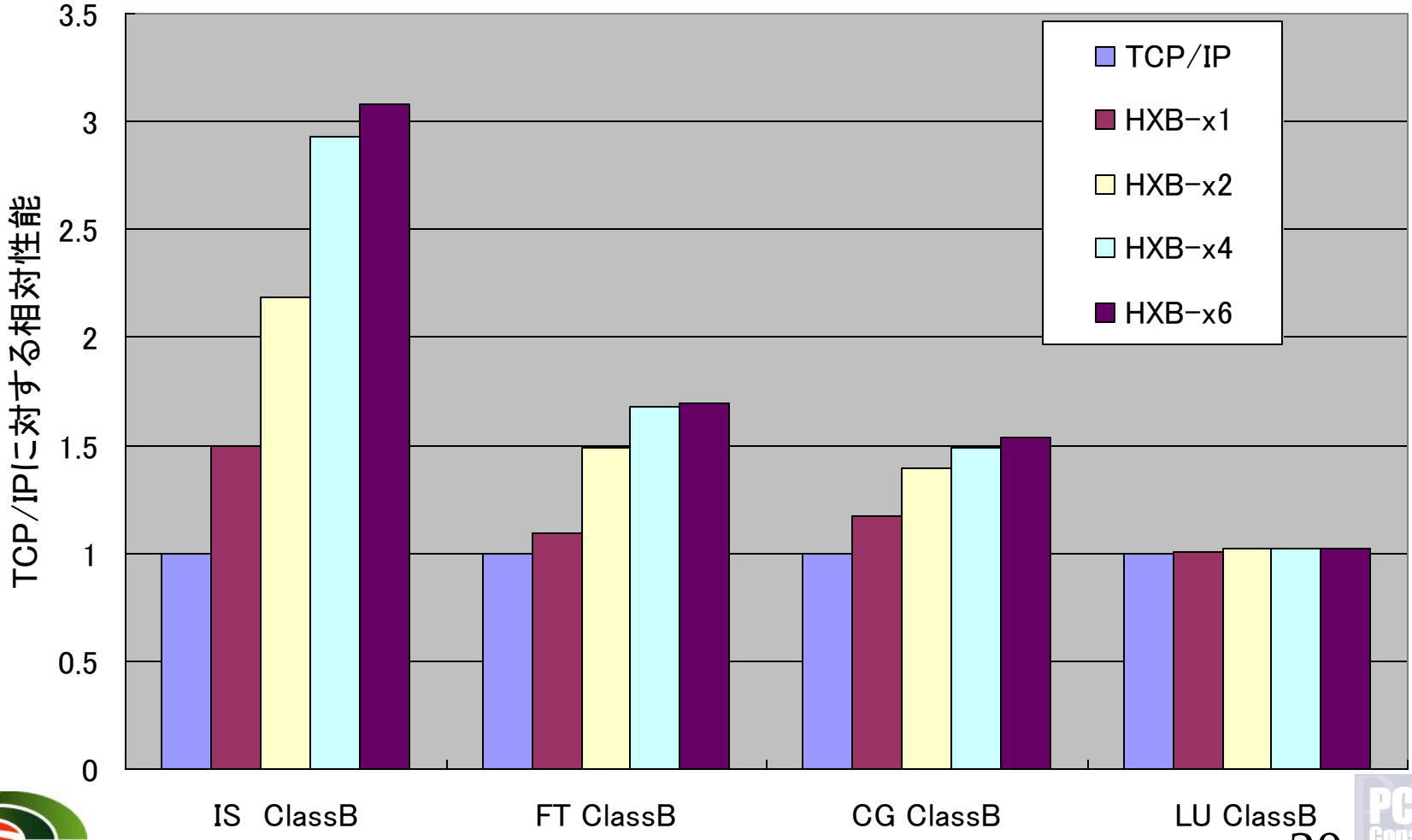
MPI通信性能比較：MPICH vs YAMPII

	MPICH 片方向	MPICH 双方向	YAMPII 片方向	YAMPII 双方向
Giga x 2	244 MB/s	478 MB/s	244 MB/s	490 MB/s
Giga x 4	493 MB/s	791 MB/s	494 MB/s	896 MB/s
Giga x 6	737 MB/s	811 MB/s	739 MB/s	937 MB/s
Giga x 8	862 MB/s	800 MB/s	915 MB/s	932 MB/s
Giga x 9	858 MB/s	790 MB/s	890 MB/s	921 MB/s

- PM/Ethernet-HXBはMPIレベルで800MB/s以上の性能を実現：専用インターコネクトに匹敵する性能
- YAMPII利用で900MB/sクラスの通信を達成

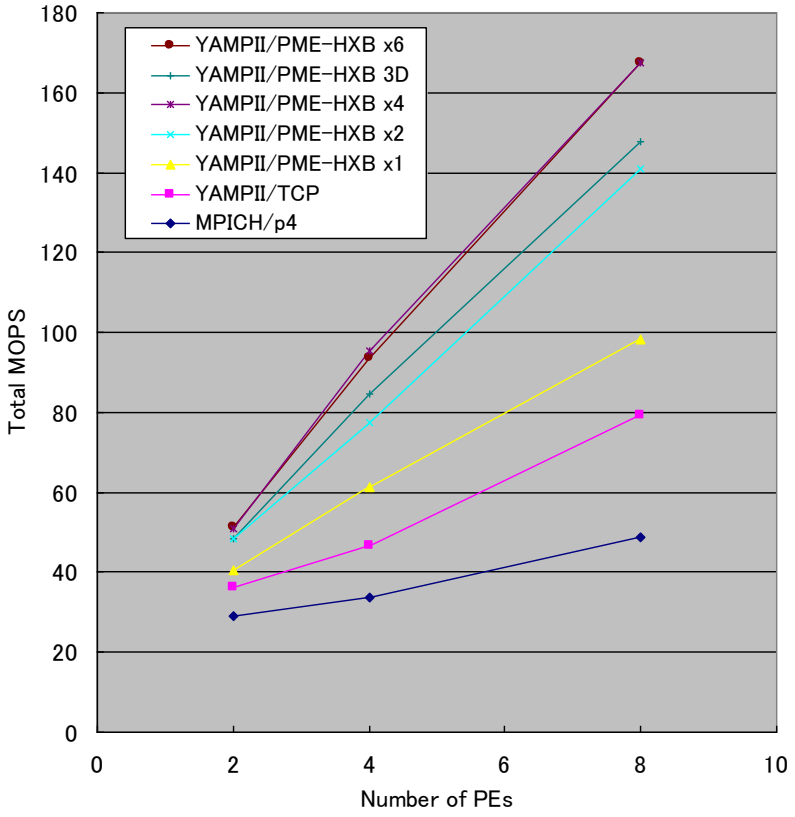
NAS並列ベンチマーク性能：MPICH

- Xeon 3.6GHz(EM64T) 4 Nodeの結果 MPICH

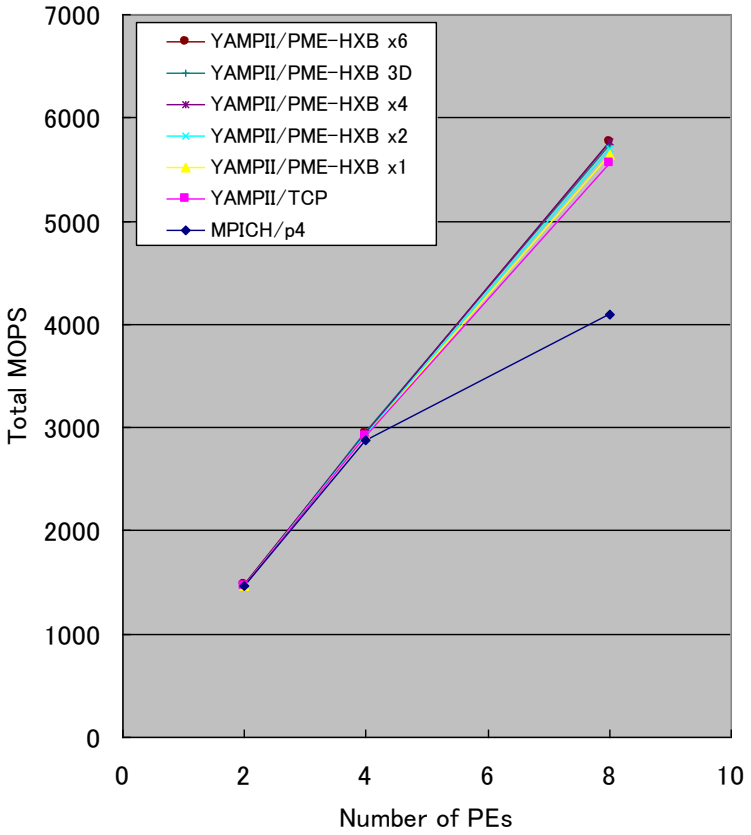


NAS並列ベンチマーク:1次元、3次元

- NPB Class C IS

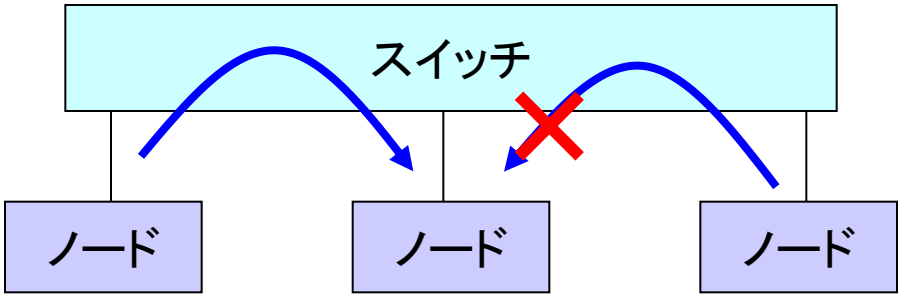


NPB Class C LU



大規模ネットワークの安定化
3次元隣接通信時の問題

- 原因はEthernetスイッチによるパケットロス
  - ノード間の処理時間のずれにより、Ethernetスイッチでパケットロスが発生し、通信性能劣化が発生
  - 全体パケットの比では0.02-0.63%がロス
  - 一部リンクのパケットロスが全体に影響



Ethernetスイッチでのパケットロス率

Broadcom 12 port

len	2	3	4node
1400	0.00	0.02	0.00 %
2200	0.00	0.19	0.99 %
4400	0.00	3.17	5.08 %
8800	5.76	8.11	8.32 %

Summit 7i(32port)

len	2	3	4node
1400	0.00	0.00	0.00 %
2200	0.00	0.00	0.00 %
4400	0.00	0.14	0.34 %
8800	0.26	0.87	1.22 %

2Linkでの結果

- メッセージ長を変化させて全対全通信（scstest）
  - JUMBO FRAME時のパケットロスが増加傾向
  - PACS-CSのスイッチは4node,8800バイト時にパケットロス率1.17%であった

まとめ

- PACS-CS向けのシステムソフトウェア
  高性能通信機構PM/Ethernet-HXBの設計
  - 複数Gigabit Ethernetで高通信性能を実現
  - 通信バッファ間でのZero-Copy通信
  - 排他制御不要のプロトコル処理
- 性能評価
  - Gigabit Ethernet x 6本で単方向735MB/s
  - 3次元クロスバー結合でも双方向1401MB/s
  - MPI通信性能: 937MB/s (双方向)
  - NAS並列ベンチマーク
- 成果は、ACM ICS-06, SACSIS-2006で発表
  - Multi-Railの通信機構としては先駆者的な存在
- 謝辞: チャレンジの機会を与えていただいた関係者の皆様に感謝致します。