FUJITSU

shaping tomorrow with you

PRIMEHPC FX10向け システムソフトウェアの概要

2012年07月06日

富士通研究所
住元 真司

発表の概要

FUJITSU

- PRIMEHPC FX10システムの概要

- PRIMEHPC FX10向けシステムソフトウェア
  - テクニカル・コンピューティング・スイート概要
  - 大規模クラスタファイルシステムFEFS
  - システム運用管理
  - ジョブ運用管理
  - 言語処理系

PRIMEHPC FX10のシステム概要

PRIMEHPC FX10の位置づけ

- マルチコアを想定したシングルCPU/ノード構成
  - FX1(4コア) → 京(8コア) → FX10(16コア)
  - 高いBF比(8 DIMM/ノード)
- 連続性に配慮しながら、超並列で必須となるアーキを段階的に導入
  - 高性能、マルチ・メニーコアへの対応：
    - SIMD拡張、自動ハイブリッド並列(VISIMPACT)
  - 高いスケーラビリティ：
    - 集団通信のHWサポート、直接網(Tofu)

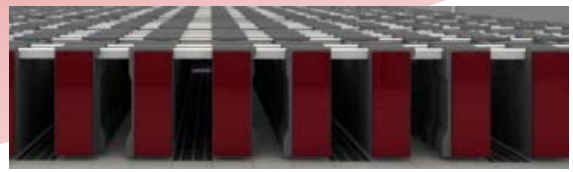
自動ハイブリッド並列
集団通信のHWサポート

FX1

CY2008～
40GF, 4コアCPU

直接網（Tofu）
SIMD拡張ほか

K computer



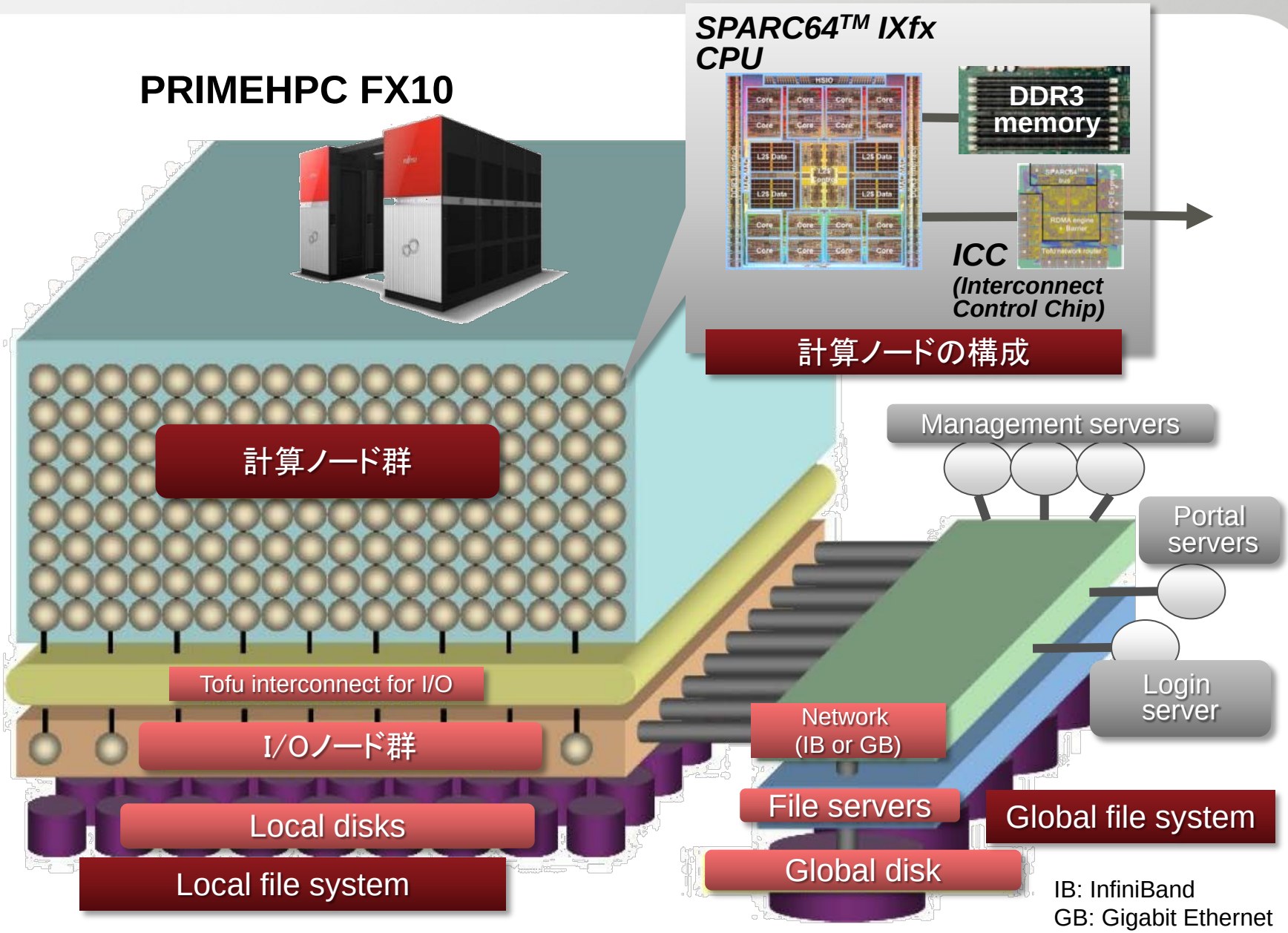
CY2010～
128GF, 8コアCPU

PRIMEHPC FX10

CY2012～
236.5GF, 16コアCPU

PRIMEHPC FX10 システム構成

FUJITSU

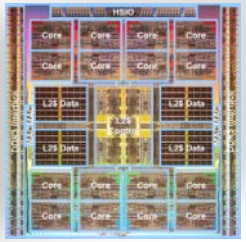


FX10 ハードウェアシステム仕様

FUJITSU

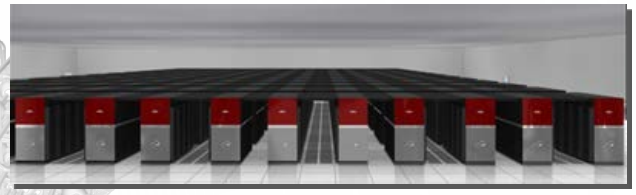
FX10ハードウェア仕様		
CPU	品名	SPARC64™ IXfx
	性能	236.5GFlops@1.848GHz
ノード	構成	1 CPU / Node
	メモリ容量	32, 64 GB
ラック	ラックあたりの性能	22.7 TFlops
システム	計算ノード数	384 to 98,304
	計算ラック数	4 to 1,024
	演算性能	90.8TFlops to 23.2PFlops
	メモリ容量	12 TB to 6 PB

- ***SPARC64™ IXfx CPU***
  - 16 cores/socket
  - 236.5 GFlops



- ***System rack***
  - 96 compute nodes
  - 6 I/O nodes
  - With optional water cooling exhaust unit

- ***System board***
  - 4 nodes (4 CPUs)



- ***System***
  - Max. 23.2 PFlops
  - Max. 1,024 racks
  - Max. 98,304 CPUs

システムハードウェア仕様の比較

FUJITSU

		「京」	FX10
CPU	品名	*SPARC64™ VIIIfx*	*SPARC64™ IXfx*
	性能	128GFlops@2GHz	236.5GFlops@1.848GHz
	アーキテクチャ	SPARC V9 + HPC-ACE extension	←
	キャッシュ構成	L1(I) Cache:32KB, L1(D) Cache:32KB	←
		L2 Cache: 6MB(12way)	L2 Cache: 12MB(24way)
	コア数/ソケット	8	16
	メモリバンド幅	64 GB/s.	85 GB/s.
ノード	構成	1 CPU / ノード	←
	メモリ容量	16 GB	32, 64 GB
システムボード	ノード数/システムボード	4ノード	←
ラック	システムボード数/ラック	24枚	←
	性能/ラック	12.3 TFlops	22.7 TFlops

システムハードウェア仕様の比較 (cont.)

FUJITSU

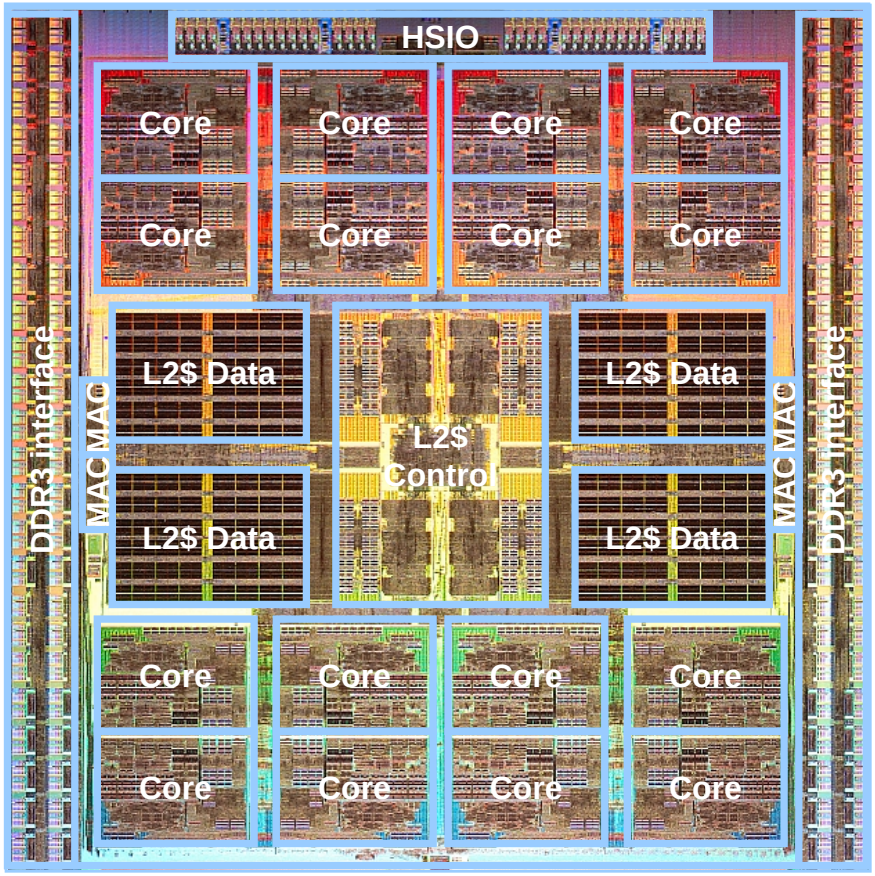
		「京」	FX10
Interconnect	トポロジ	6D Mesh/Torus	←
	性能	5GB/s x2 (双方向)	←
	ノードあたりリンク数	10本	←
	追加機能	ハードウエアバリア、リダクション	←
	実装	外部スイッチ不要	←
冷却	CPU, ICC(interconnect chip), DDCON	直接水冷	←
	DIMMs, other parts	空冷	空冷 + EXCU(Exhaust air cooling unit)

PRIMEHPC FX10は「京」とバイナリコンパチブル（ランタイムで吸収）
性能チューニングにおいては、L2キャッシュサイズ、Way数、メモリ容量、コア数の違いを意識

SPARC64™ IXfx

- 性能と省電力を徹底追及した自社開発プロセッサ
  - 最大236.5GFLOPSの演算性能
  - HPC向け機能・性能の強化（高いメモリバンド幅）
- 高い信頼性を確保

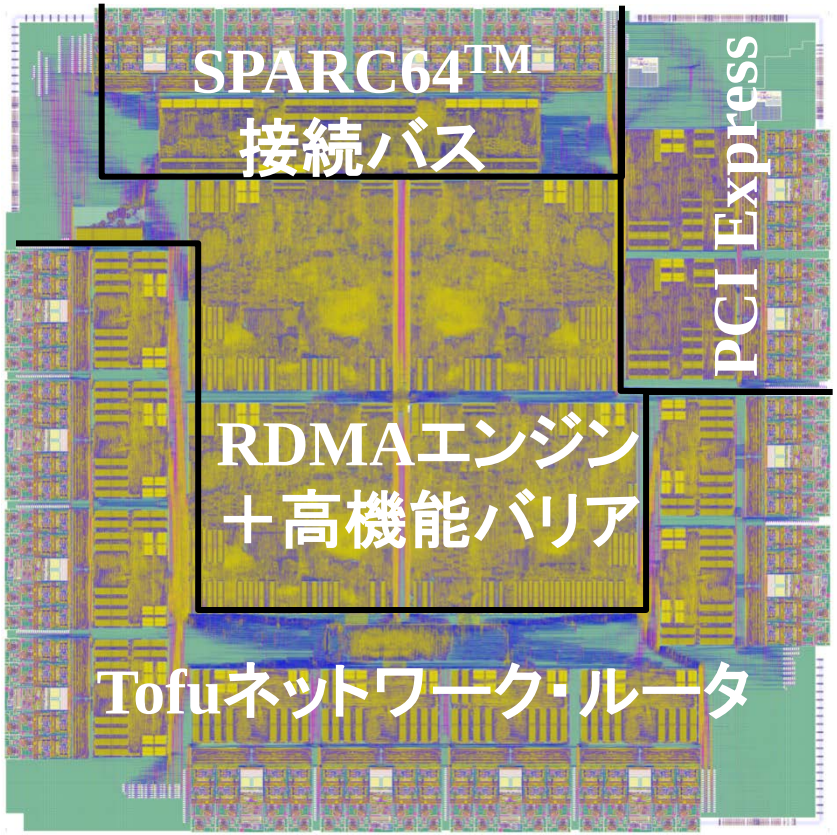
	SPARC64™ IXfx
コア数	16
動作周波数	1.848 GHz
理論ピーク性能	236.5 GFLOPS
メモリースループット	85 GB/s



Tofu インターコネクト

- 「京」で実証された高いスケーラビリティ
  - 直接網を実現する自社製インターコネクトコントローラ（ICC）
- 高い信頼性と運用性
  - ジョブごとに3次元トーラスビューを提供
  - 故障ノードを回避し運用を継続

	ICC
同時通信数	4送信+4受信
動作周波数	312.5MHz
スイッチ容量	100GB/s
リンク速度、ポート数	5GB/sx双方向x10ポート

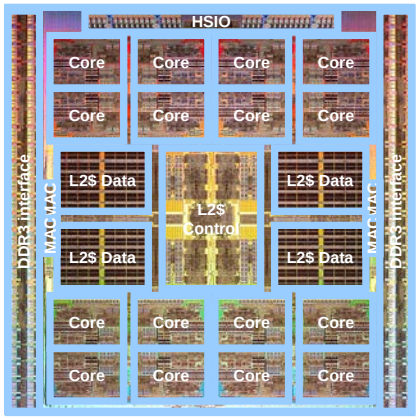


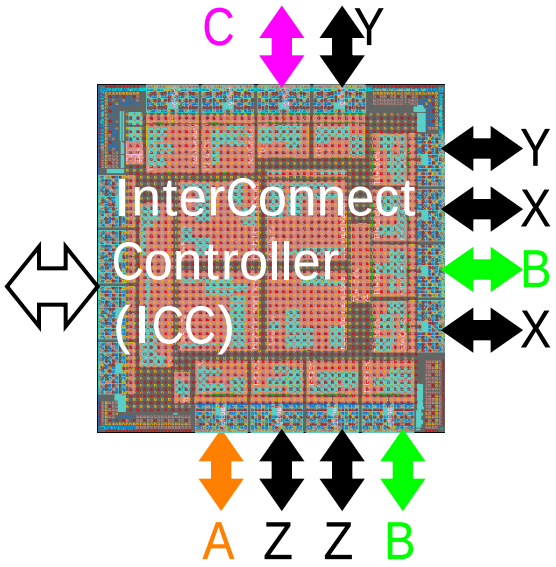
Tofuインターコネクト 概要

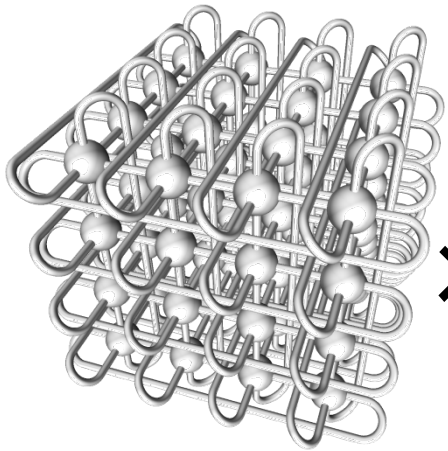
- SPARC64™ VIIIfx, IXfx専用のノード間インターコネクト
- “**To**rus **fu**sion” 3D-Torus × 3D-Torus = 6D-Torus

ネットワーク・トポロジ	6次元メッシュ／トーラス
座標軸	X, Y, Z, A, B, C
最大ネットワーク・サイズ	32, 32, 32, 2, 3, 2
FX10システム構成	トーラス軸: X, Z, B ／ メッシュ軸: Y, A, C 計算ノード: Z = 1～8 ／ IOノード: Z = 0

SPARC64™ IXfx

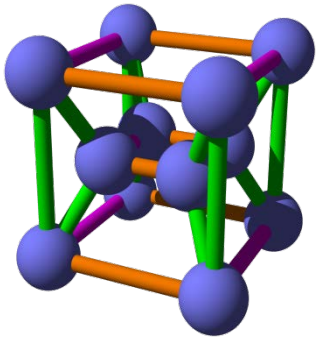






XYZ

×

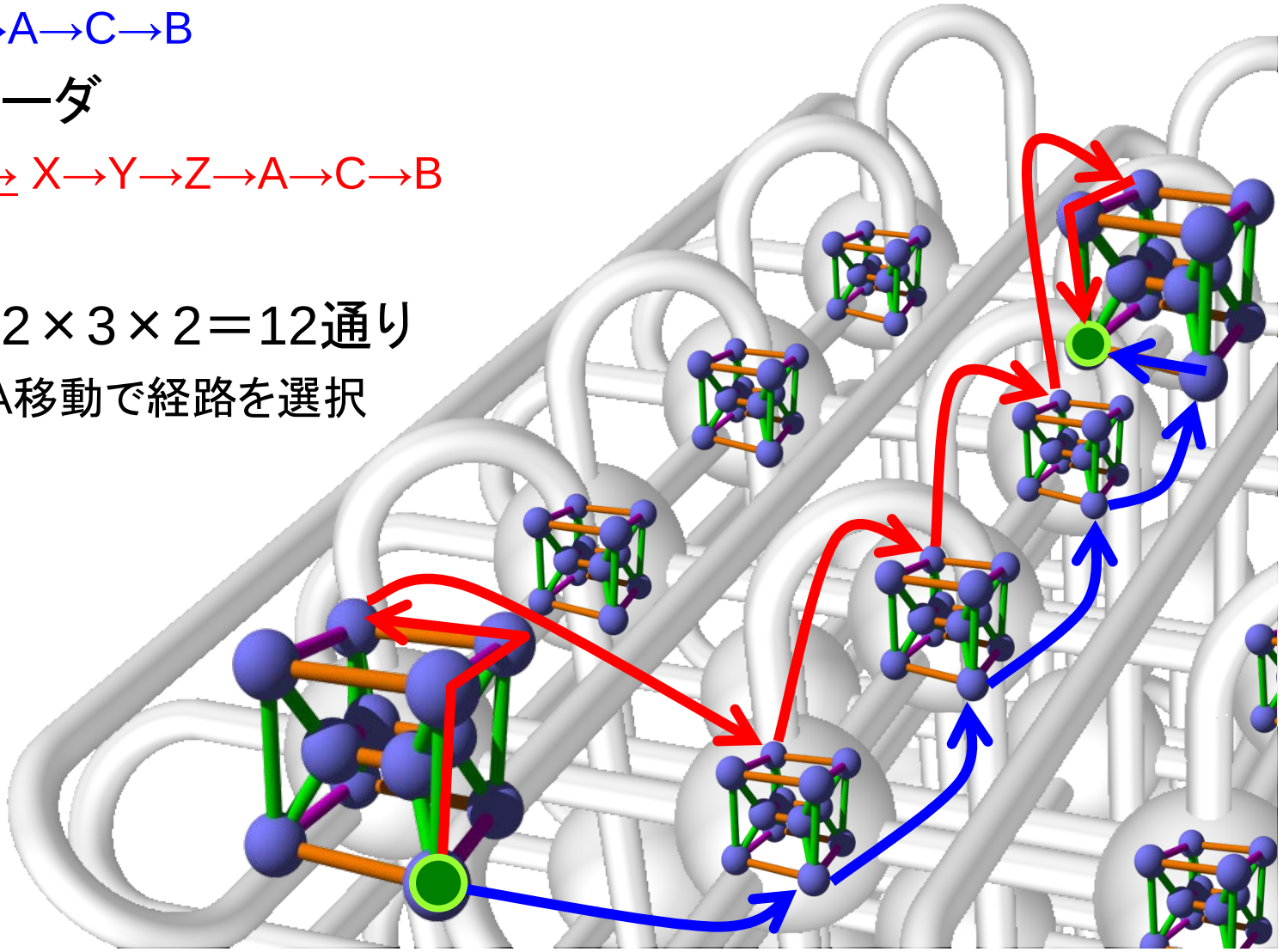


ABC

ルーティング・アルゴリズム

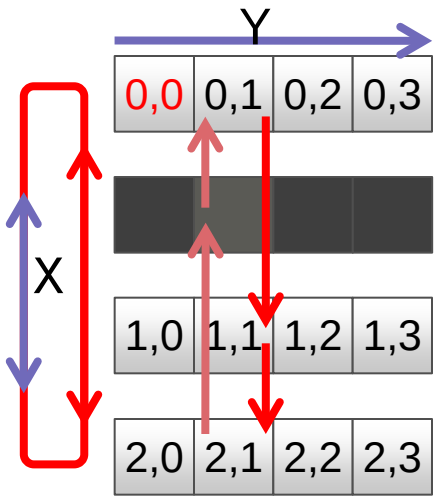
FUJITSU

- デフォルトの次元オーダ
  - X→Y→Z→A→C→B
- 拡張次元オーダ
  - **B→C→A→** X→Y→Z→A→C→B
- XYZ経路は2×3×2＝12通り
  - 最初のBCA移動で経路を選択

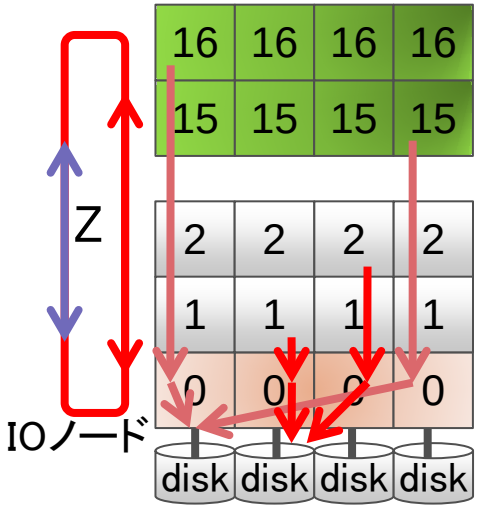


Tofu:6次元メッシュトーラス

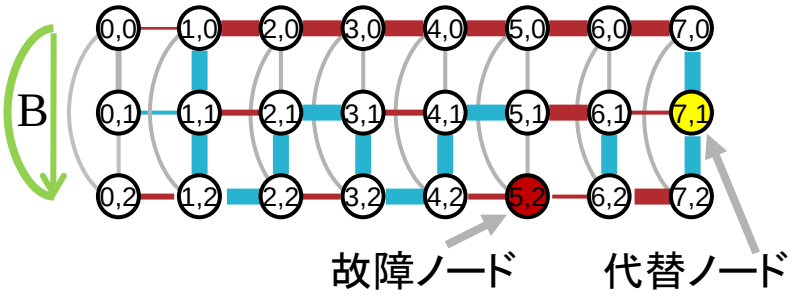
- (X,Y,Z,A,B,C)=32x32x32x2x3x2の約40万ノードまでの拡張性
- A/B/C軸とX/Y/Z軸の組み合わせることで、ジョブが割当たる部分メッシュ領域で隣接3次元トーラスが構成可能
- X/Z/Bはトーラスとして、種々の運用、故障に対応
  - X軸：保守中の運用を継続
  - Z軸：IOデータを直下のIOノード(Z=0)と通信
    - 最大ホップ数1/2、他ジョブとの輻輳を緩和
  - B軸：故障ノードを回避して、トーラスを再構成



保守時にもX-1列で継続



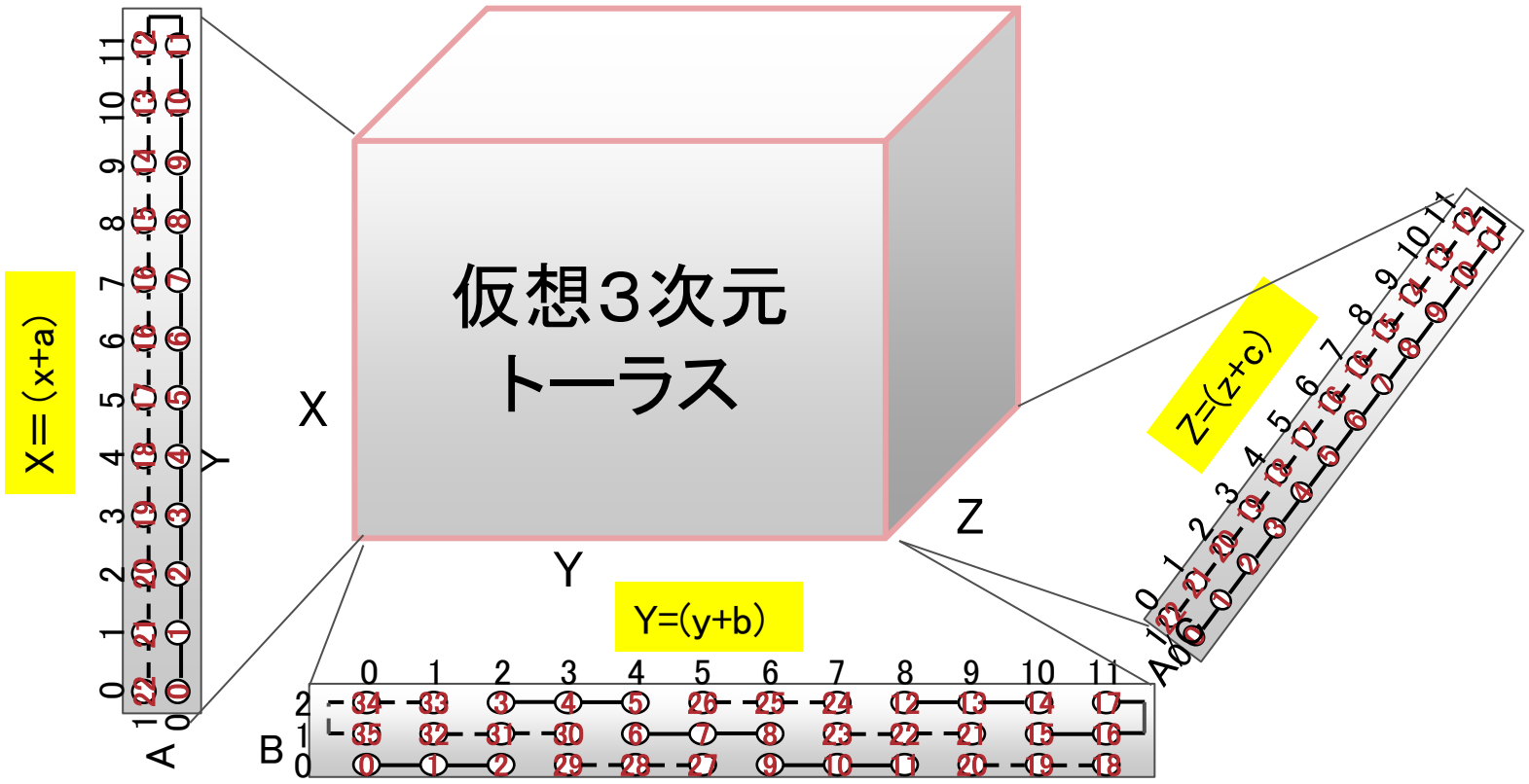
IOデータはホップ数1/2で転送



故障ノードの回避例

Tofuへのジョブ割り当てとランクマッピング

- ユーザはジョブを1～3Dトーラス指定で割り当てが可能
- TofuのXYZとABCを組み合わせた軸がトーラスの軸
  - XYZとABCの組み合わせ方と、トーラスの軸への対応は選択可能
- 3次元トーラスマップの基本は、X=x+a, Y=y+b, Z=z+cの組み合わせ



性能比較（ご参考）

FUJITSU

		FX10 SPARC64™ IXfx	京 SPARC64™ VIIIfx
CPU	動作周波数[GHz]	1.848	2.00
	コア数/チップ	16	8
	DGEMM[GFLOPS]	224.5	123.6
	対『京』比	1.8	1
	ピーク比	94.9%	96.6%
	STREAM[GB/sec]	64.8	46.2
	対『京』比	1.4	1

PRIMEHPC FX10の性能は、「京」に対して、
演算性能で約1.8倍、メモリ性能で約1.4倍

FUJITSU

PRIMEHPC FX10向け システムソフトウェア

- テクニカル・コンピューティング・スイート

テクニカル・コンピューティング・スイートの構成

FUJITSU

アプリケーション（高並列アプリ・アルゴリズムの研究開発）

HPC Portal / System Management Portal

Technical Computing Suite

システム管理ソフト

- システムの管理、制御、監視、運用サポート
- 故障の自動リカバリを実現 (365日24時間運用)

ジョブ管理ソフト

- 高速並列ジョブ起動
- 高効率スケジューラ
- 充実したセンタ運用機能

エクサバイト対応 高性能ファイルシステム（FEFS）

- Lustreをベースに拡張
- 高いスケーラビリティ (IOサーバ数千台に対応)
- 1TB/sクラスのIO転送性能実績

Whamcloudとコラボ

自動並列化コンパイラ

- Fortran, C, C++を提供
- 高レベルのSMID並列、マルチコア並列をサポート

ツール・数学ライブラリ

- 多くの海外ツールをサポート
- 高効率の数学ライブラリ (SSL II/BLAS etc.)

並列言語・通信ライブラリ

- OpenMP, MPI(Open MPIベース), XPFortranを提供
- マルチコア、超並列に対応

Linux based OS (enhanced for FX10)：OSノイズ対応

「京」/ PRIMEHPC FX10 / PCクラスタシステム

テクニカル・コンピューティング・スイートの設計指針

FUJITSU

- 設計指針：「京」、PCクラスタとの実行環境の統一
  - PCクラスタからのアプリケーション移植性、オープンソースソフトウェア(OSS)の移植性を最優先に考慮
  - 再コンパイルでPCクラスタよりも高並列、高速動作が目標
  - 「京」との親和性を最大限に確保
- OSとしてLinuxを採用し、各コンポーネントにOSSを最大限に活用：　Lustreファイルシステム、Open MPIなど
  - ただし、ノウハウが必要な運用系ソフトは独自開発

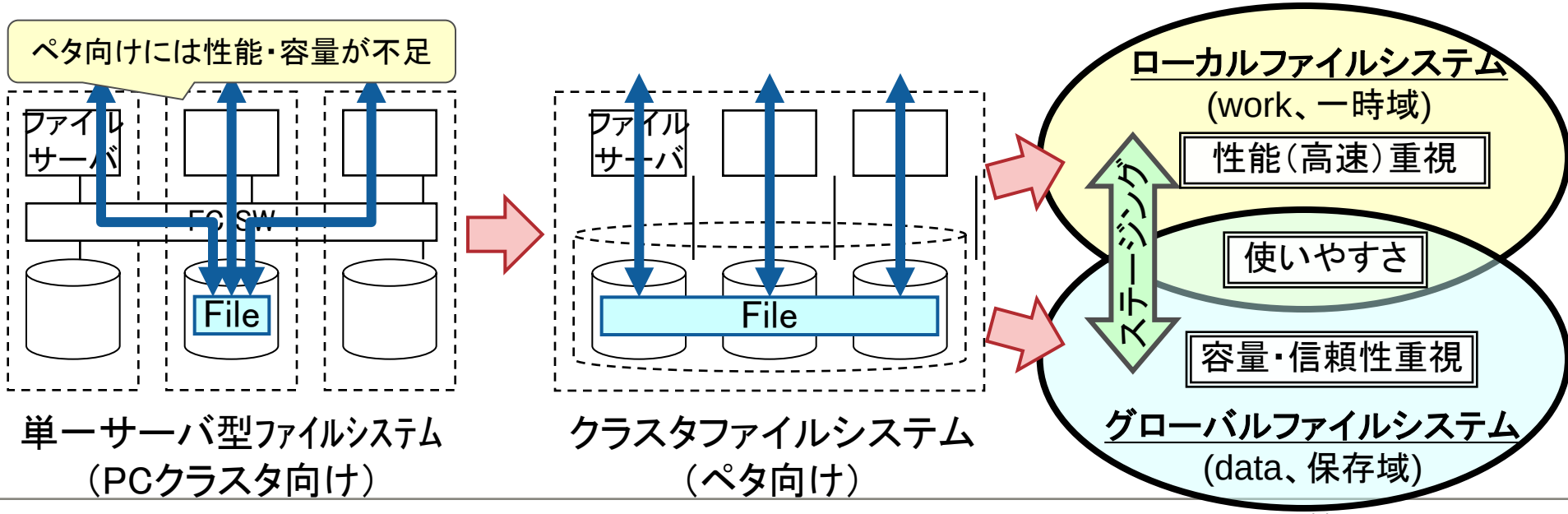
FUJITSU

大規模クラスタファイルシステム FEFS

Lustreベースファイルシステム(FEFS)の概要

FUJITSU

- 世界トップクラスに相応しい最大規模、最速IO性能が目標
  - エクサバイト対応済み、**目標** **2011年： 100PB, 1TB/s**
- 既存スパコンで実績のある「Lustreファイルシステム」（GPLv2）をベースに開発
  - 目標実現に必要な機能を追加：大規模化、性能改善、運用性、信頼性向上が主体
- 階層ファイルシステムにより適材適所化と負荷分散実現
  - ジョブ専用と共用のファイルシステム間でファイル転送
  - ファイルシステム間でファイルを転送を行う「ステージング機能」を開発
- Lustreコミュニティ（Open SFS）に参画し、Lustre標準化を推進
  - Open SFS:　Lustreの標準化と開発を担う非営利組織



Lustre仕様とFEFSの実現目標

FUJITSU

Features		Current Lustre	Our 2012 Goals
System Limits	Max file system size	64PB	100PB (8EB)
	Max file size	320TB	1PB (8EB)
	Max #files	4G	32G (8E)
	Max OST size	16TB	100TB (1PB)
	Max stripe count	160	20k
	Max ACL entries	32	8191
Node Scalability	Max #OSSs	1020	20k
	Max #OSTs	8150	20k
	Max #Clients	128K	1M
Block Size of *ldiskfs* (Backend File System)		4KB	~512KB

富士通のGoalは仕様や実現規模でLustreの将来仕様を先取りすることになった
2011/2 Lustreコミュニティに提供、コミュニティのロードマップとして公開（LUG2011）

FEFSの特徴

FUJITSU

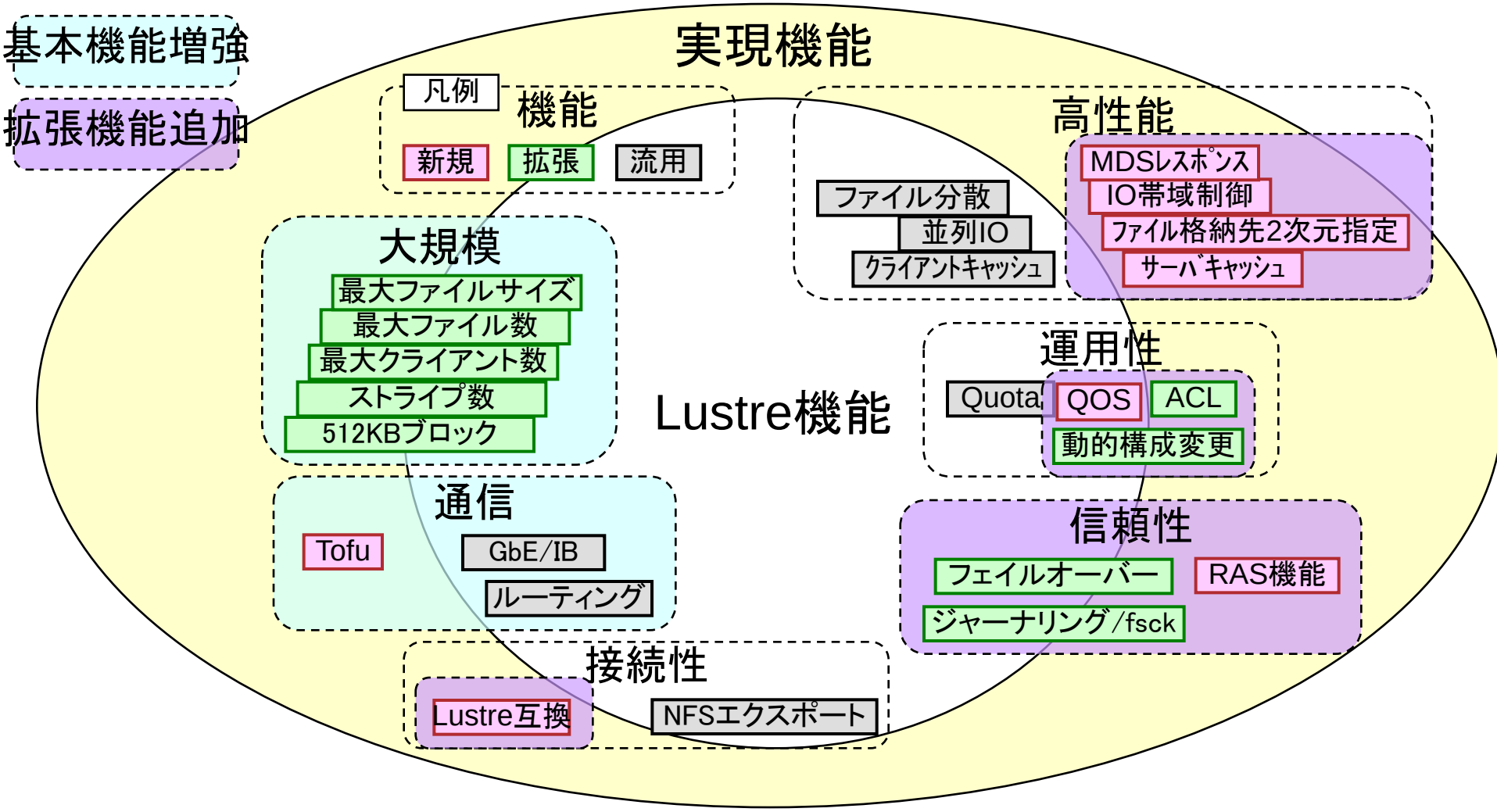
■ 目標： 10年先のエクサ世代にも対応できるファイルシステム

分類		施策
大規模化	ファイルサイズ ファイルシステム規模	10年後にはエクサバイト（EB）へ ⇒ EBのサイズに対応
スケーラビリティ	性能 容量	ハード増設で性能・容量がスケールアウト ⇒ 2年で5～10倍のペースでup
運用ポリシー選択（センター運用）	フェアシェア	一人の悪行が他人に迷惑をかけることを防ぐ（QoS）
	総スループット優先	帯域を最大限有効利用（ベストエフォート）
	安定したジョブ実行	ジョブ実行時間のバラツキを抑える
性能向上	メタデータアクセス	①ソフト処理改善（メタDisk分散，ロック・ハッシュ改善等） ②MPI-IO対応（ファイル数減⇒メタアクセス緩和） ③メタアクセスの分散化
	TSSレスポンス保障	TSSノードの要求に対する帯域保障
	小サイズIO・多数ファイル	クライアントキャッシュ（データ，属性）
	ディスク性能超え	サーバキャッシュ（大容量キャッシュ）
高可用性	フェイルオーバー	サーバ・パス二重化 ⇒単点故障はJOBは自動継続
使い易さ	動的なハード増設	運用中にディスクやサーバの増設が可能
	運用状態の監視	稼動状態の見える化 ⇒監視・チューニングを容易に

FEFSにおけるLustre拡張

FUJITSU

- Lustreをベースに不足機能を拡張・新規開発



FUJITSU

システム運用管理

- 大規模システムの運用に向けた取り組み
- 階層化による負荷分散
- 徹底した二重化
- ジョブ実行効率の向上（外乱抑止）

大規模システムの運用に向けた取り組み

FUJITSU

- 大規模システムによる処理の集中
  - 全体の情報管理のためのノードは一台とせざるを得ないが、その一台だけで多数ノードに対する指示や制御は処理遅延の要因
  - ○ 指示や制御の負荷分散が必要
  - ⇒ 階層化による負荷の分散
- 故障時の運用停止リスク
  - 管理系やファイルシステム系のノード故障時に運用停止の可能性
  - ○ 重要ノードが故障してもシステムは継続できる必要がある
  - ⇒ 可能な限りの冗長構成とサーバ機能切り替え時でもジョブは継続
- システムノイズの波及
  - ノード単体で発生するノイズ自体がジョブ実行の妨げ
  - ノード単体のノイズはノード間並列ジョブではシステム全体に波及
  - ○ ノード単体のノイズの削減やシステムへのノイズ波及を最小限にとどめる必要がある
  - ⇒ OSノイズ削減の施策とノイズの同期によるノイズ波及をとどめる施策

階層化による負荷分散

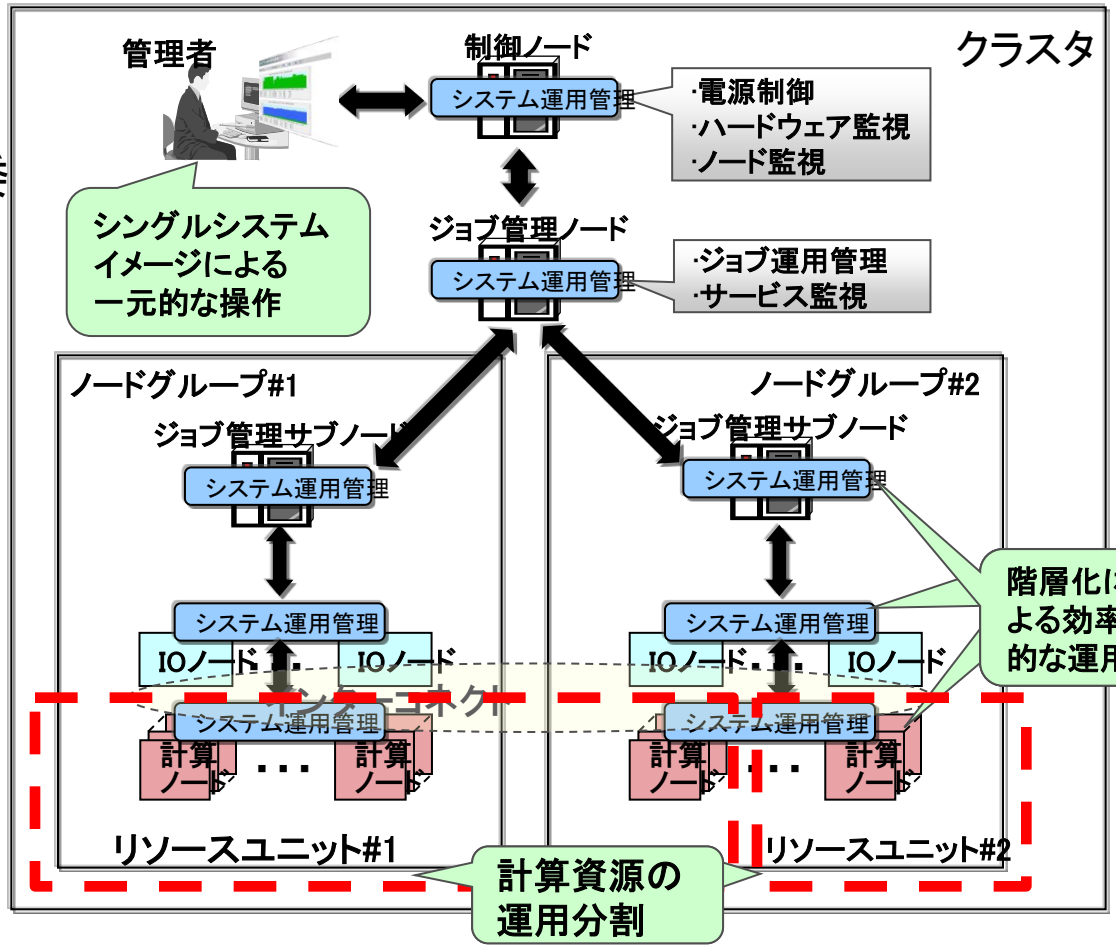
効率的な監視と制御

ノードグループという単位で処理を分散

- ジョブ管理サブノードにノード監視やジョブ実行の処理を委託
- ノードグループ単位にシステムを追加可能であり、高い拡張性を持つ
- 情報もグループ単位に集約して収集するなど、効率化

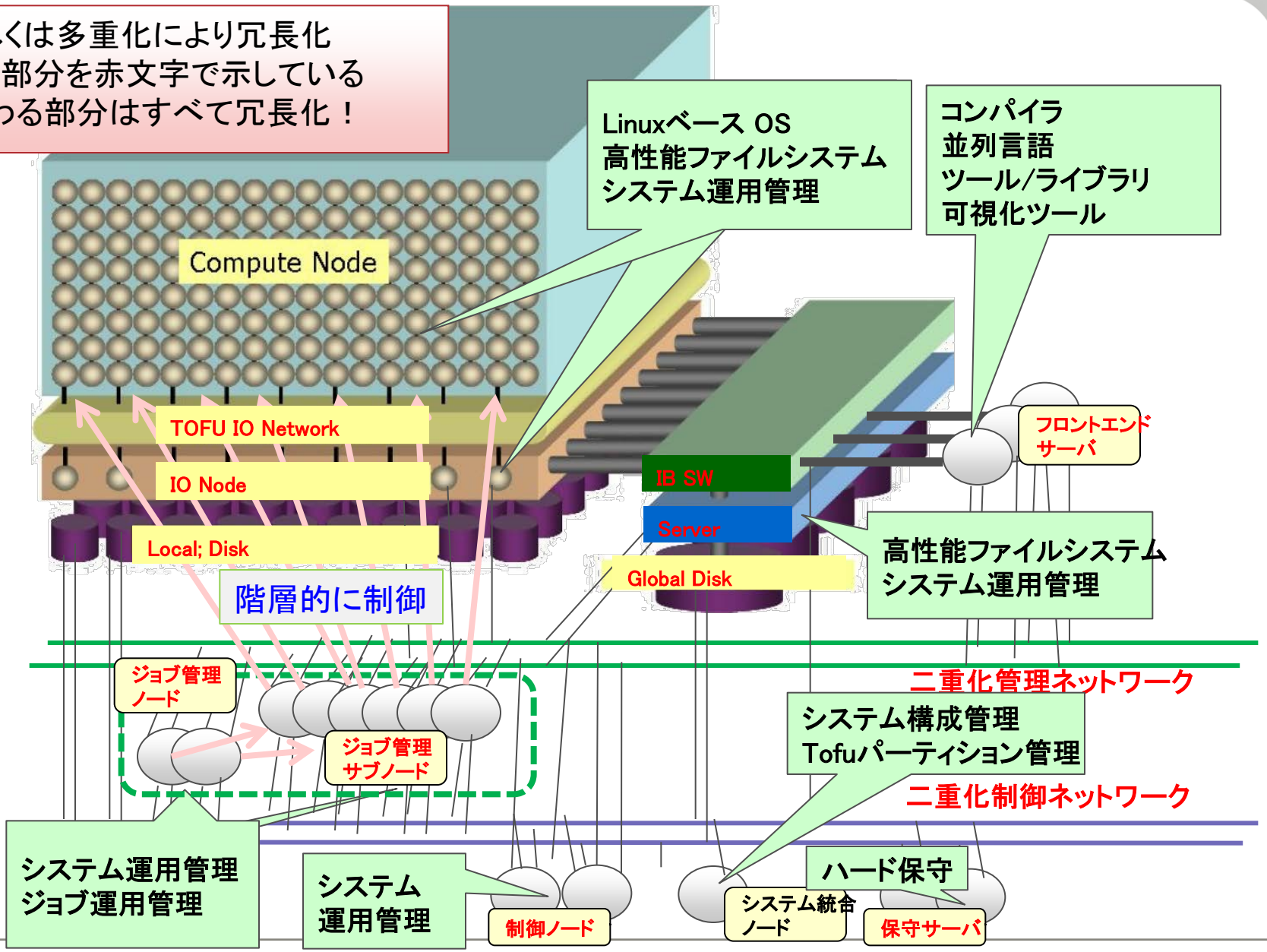
計算ノード分割運用

- リソースユニットという論理的に計算資源を運用分割可能
- ノードグループとは独立的に設定することが可能



徹底した二重化

二重化もしくは多重化により冗長化
されている部分を赤文字で示している
運用に関わる部分はすべて冗長化！



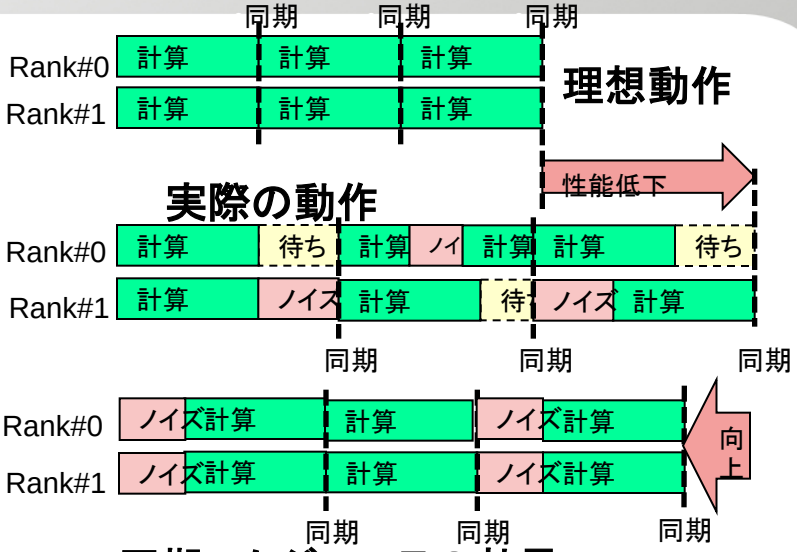
ジョブ実行効率の向上（外乱抑止）

FUJITSU

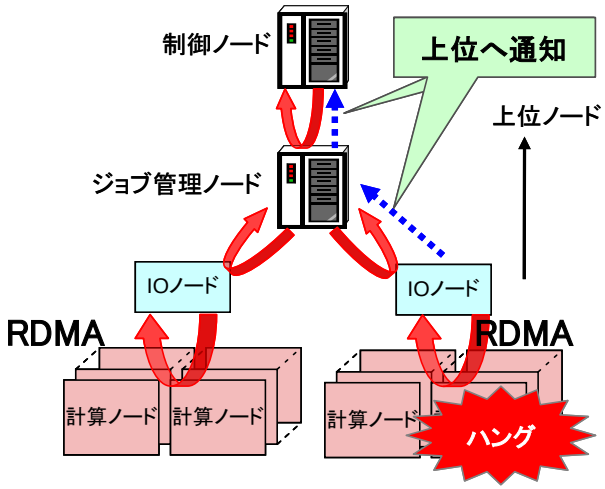
- OSノイズ対策
  - OS内のノイズ源を把握し規定内のノイズ間隔と継続時間(200ms周期で50us以内の処理)に抑える
  - このためにLinux OS内のノイズ源をすべて調査し、OSコードを改良
- システムノイズ対策
  - Tofuのハードウェアバリア活用の同期スケジューリングにより大規模並列実行時における影響を最小化
  - システムの監視動作などもTofuのRDMA機能を活用し、影響を最小限

	ノイズ間隔	継続時間
一般Linux	98ms	215us
『京』目標	200ms	50us

OSノイズの影響と「京」の目標



同期スケジューラの効果



RDMA利用のノード死活監視機構

FUJITSU

ジョブ運用管理

- 柔軟なジョブスケジューリング
- ジョブの実行順序の選択
- ジョブの実行資源の選択
- ジョブ間の相互干渉を抑制する資源割り当て

柔軟なジョブスケジューリング

FUJITSU

- 共同利用センターの多種多様な要件への対応
  - きめ細かな運用ポリシー設定
    - ジョブの実行順序の選択
    - ジョブの実行資源の選択
- 計算資源の効率実行をサポート
  - 未来割り当て（バックフィルスケジュール）による資源の効率利用
  - ジョブ間の相互干渉による性能劣化を抑制する資源割り当て

ジョブの実行順序の選択

FUJITSU

- 柔軟にスケジューリング・ポリシーを設定可能
  - 種別の優先度と種別内の優先度によりジョブ実行順序を決定
  - つまり、優先度の高い種別から順番にソートを実施する

優先度種別※	説明
ジョブ投入の順序	先に投入されたジョブが高優先度。(デフォルト)
フェア・シェアリング	各ユーザグループ / ユーザごとに設定された「ポイント」の残量の順に優先度を決定。
要求資源量による優先度	ジョブ投入時に指定した資源量により決まる優先度。 例：高並列ジョブを優先的に実行
ユーザグループ / ユーザ / ジョブごとの優先度指定	ユーザグループおよび個々のユーザごとに優先度を指定。 個々のユーザがさらに自分のジョブの優先度を指定可能。

※ 優先度種別の間の調整・入れ替えは可能

ジョブの実行資源の選択

FUJITSU

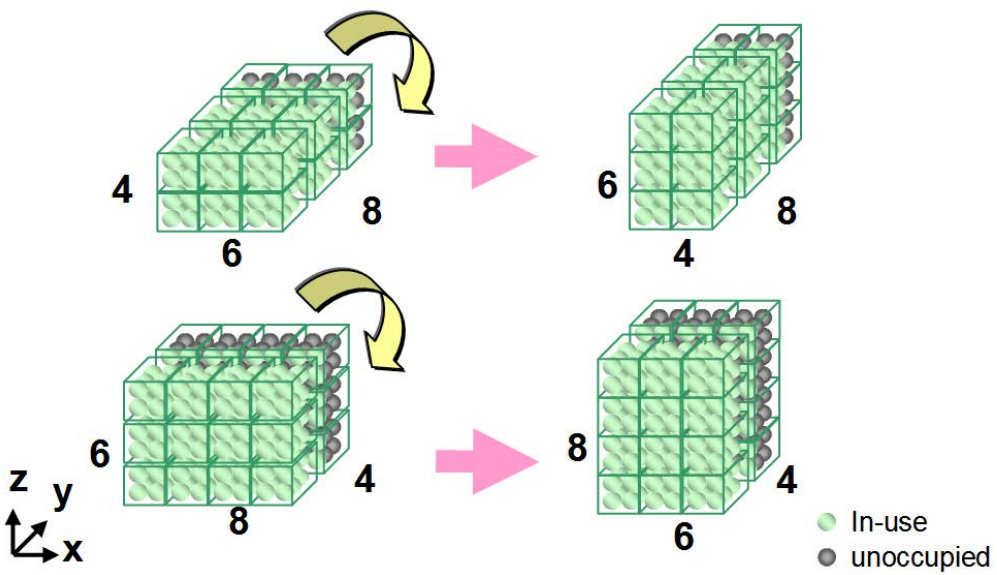
- 実行順番で選択されたジョブへ割り当てる資源の選択ルールがカスタマイズ可能
  - 資源選択ルールの有効/無効、設定値などをセンター運用に合わせてカスタマイズ

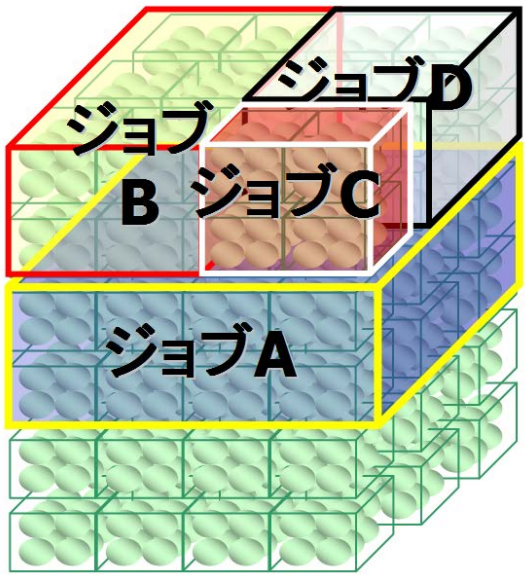
資源選択ルール	説明
計算ノードの固定的な分割運用	計算ノードを論理的に分割(リソースユニット)。 リソースユニットを指定し、その中から空き資源を探す。
同時実行ジョブ数制限、 同時実行サブジョブ数制限、 同時実行ノード数制限	特定ユーザのジョブが資源を占有しないように空き資源を探す(同一時間帯に実行数制限値を超えるような場合は未来に割当てる)。
ジョブ形状の最適化 (回転)	割当て形状を回転させ、空き資源を探しやすくすることにより、無駄な空き資源の発生を抑制。
バックフィル(高優先度ジョブの実行を妨げない範囲でジョブ実行順序を無視する)	優先度の高いジョブの資源割り当て予約後の空き資源(隙間)を探す。システム稼働率の向上に繋がる。
デッドライン	資源にデッドラインを設定し、設定期間中はその資源を避けて空き資源を探す。計画停止や保守への対応。

ジョブ間の相互干渉を抑制する資源割り当て

FUJITSU

- インターコネクト構成を意識した資源割当て
  - Tofu単位を基準として割当て
  - インターコネクトで形作る部分直方体をジョブに割当て
  - ➔隣り合うTofu単位群を割り当てることでノード間通信の効率を高め、他のジョブからの干渉を防ぐ
  - ジョブ割当て形状の最適化
  - ➔ジョブの割当て形状を回転、システムの空き領域にはめ込む





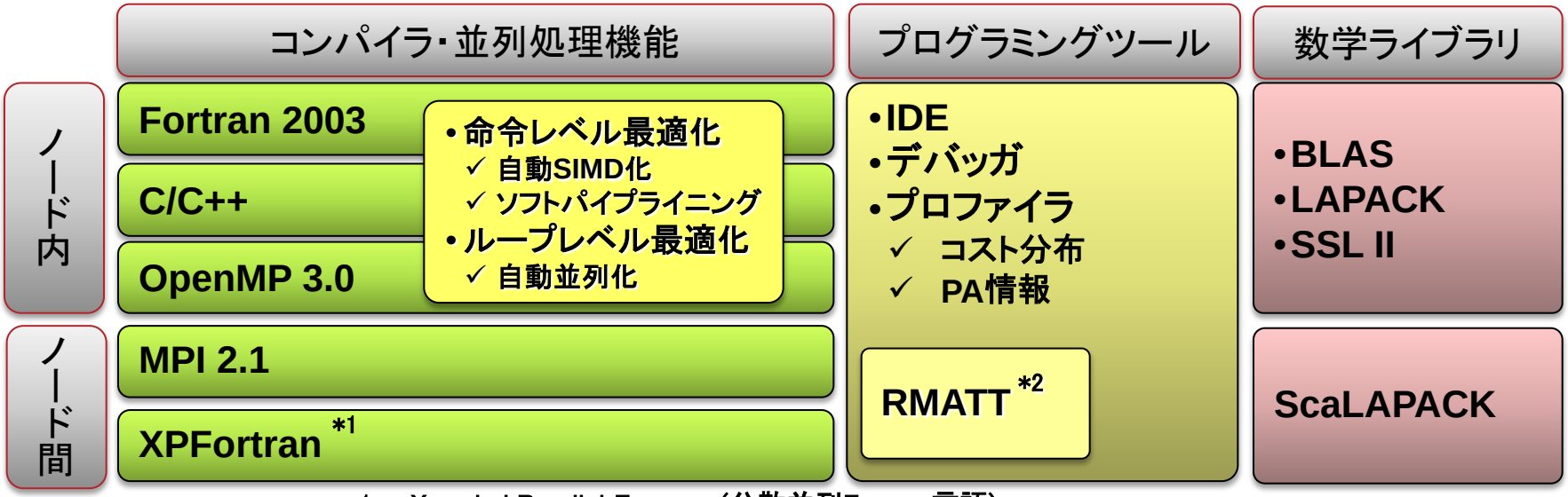
FUJITSU

言語処理系

言語処理系

FUJITSU

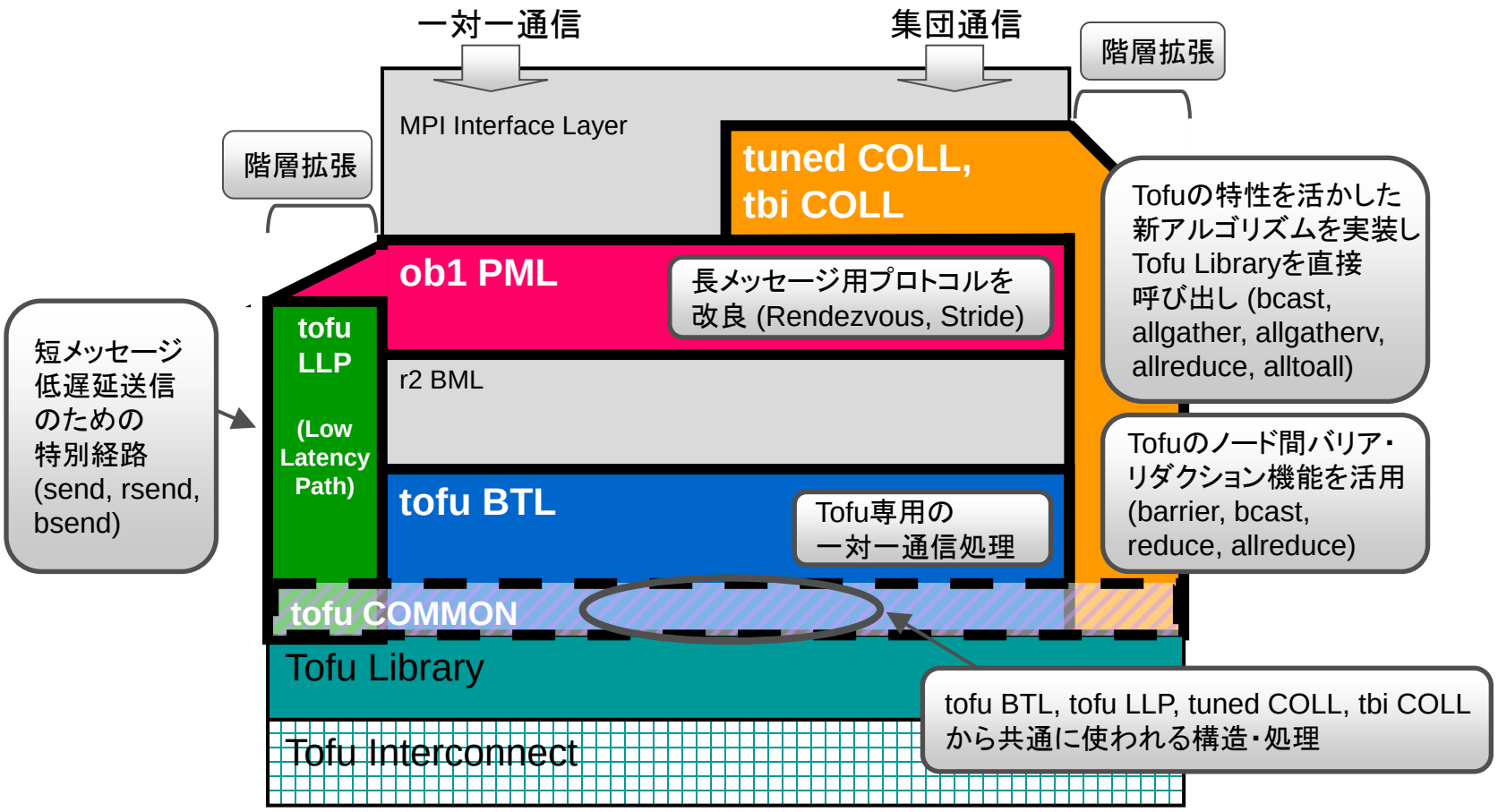
- 超高並列アプリケーション開発環境をトータルにサポート
  - 高性能・高効率を実現するコンパイラ・並列処理機能
    - 自動ベクトル化技術を発展させた高効率な自動SIMD化/並列化コンパイラ
    - 世界的に著名なOpen MPIをベースとして超高並列時の通信性能を大幅に改善したMPI
  - 超高並列アプリ開発に対応できるデバッグツール・チューニングツール
  - マシン性能をギリギリまで引き出す高効率な数学ライブラリ



*1: eXtended Parallel Fortran（分散並列Fortran言語）
*2: Rank Map Automatic Tuning Tool（ランクマッピング最適化）

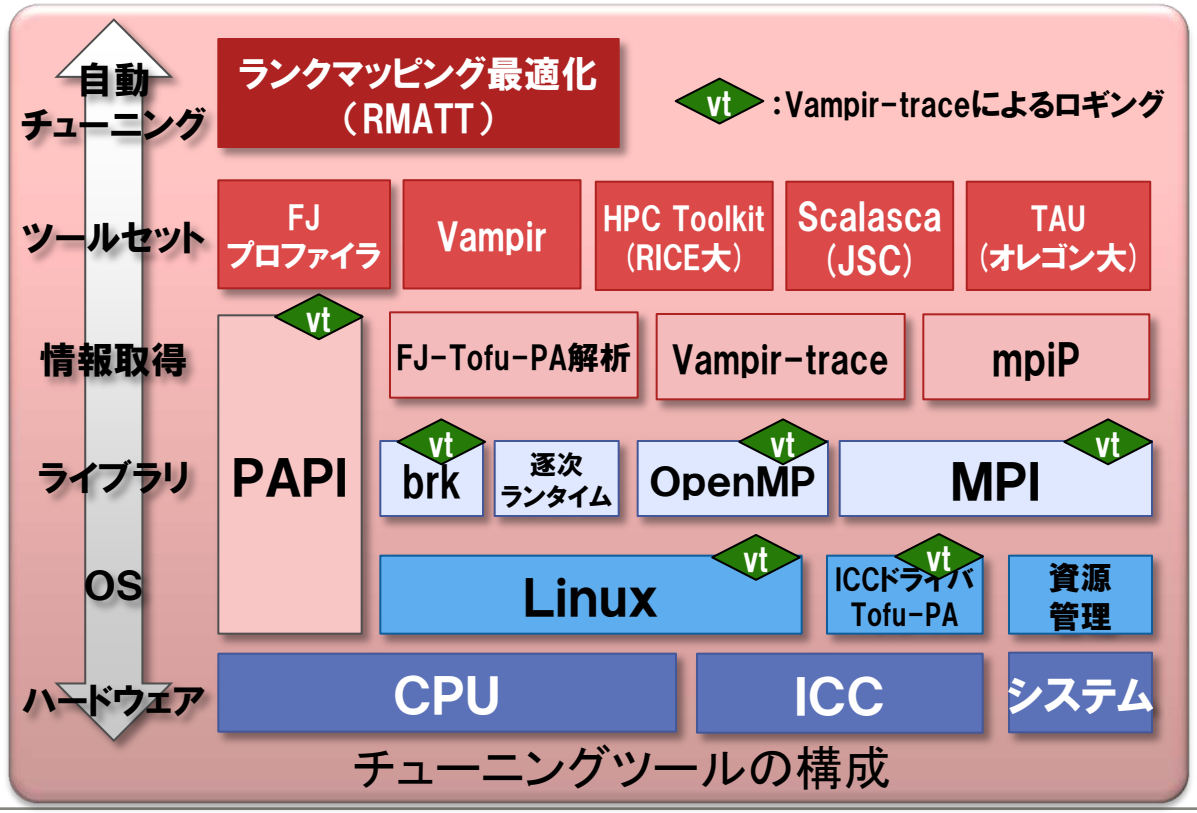
MPIライブラリの構造

- 先進性/拡張性/コミュニティのオープン性に優れたOpen MPIにTofu向け高速化機能をアドオン

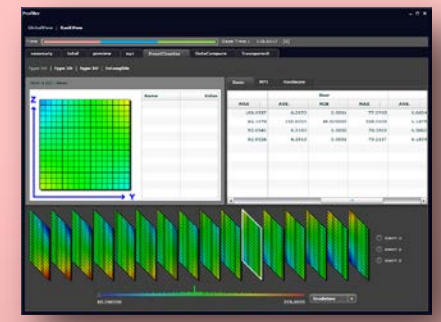


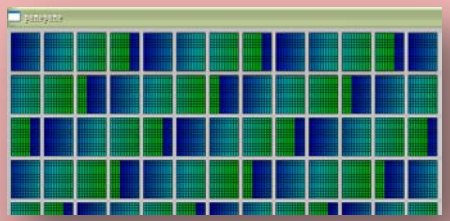
チューニングツールの概要

- 従来のプロファイラを超高並列に対応できるように改善
  - ロードインバランス解析機能
  - Tofuのハードモニタを利用した通信プロファイリング機能（Tofu-PA解析）
  - トーラス上でのランクマッピング最適化機能（RMATT）
  - 超高並列でも分かり易い情報の表示機能
- デファクトのインタフェース採用により外部ツールの容易な導入を実現



チューニングツールの構成


アプリトポロジ表示

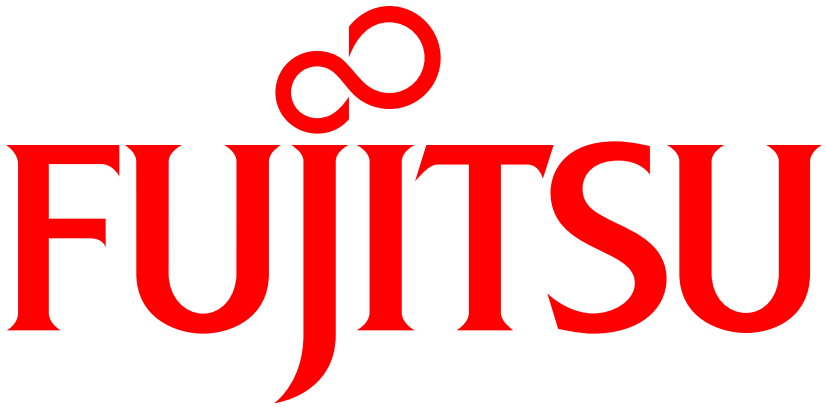

物理レイアウト表示

情報の表示例

まとめ

FUJITSU

- PRIMEHPC FX10のシステム概要
  - プロセッサSPARC64™ IXfx新規開発(236.5GFLOPS)
  - 直接網Tofuインタコネクトの適用(98,304ノードまでの拡張性）
- PRIMEHPC FX10向けのシステムソフトウェア「京」とのアプリケーション互換性確保
  - テクニカルコンピューティングスイート概要
  - 大規模クラスタファイルシステムFEFS
  - システム運用管理
  - ジョブ運用管理
  - 言語処理系



shaping tomorrow with you