九州大学学術情報リポジトリ
Kyushu University Institutional Repository

HPCアーキテクチャの今後

村上, 和彰
九州大学大学院システム情報科学研究院 | 九州大学情報基盤センター

https://hdl.handle.net/2324/9110

出版情報：SLRC プレゼンテーション, 2004-10-08. 九州大学システムLSI研究センター
バージョン：
権利関係：

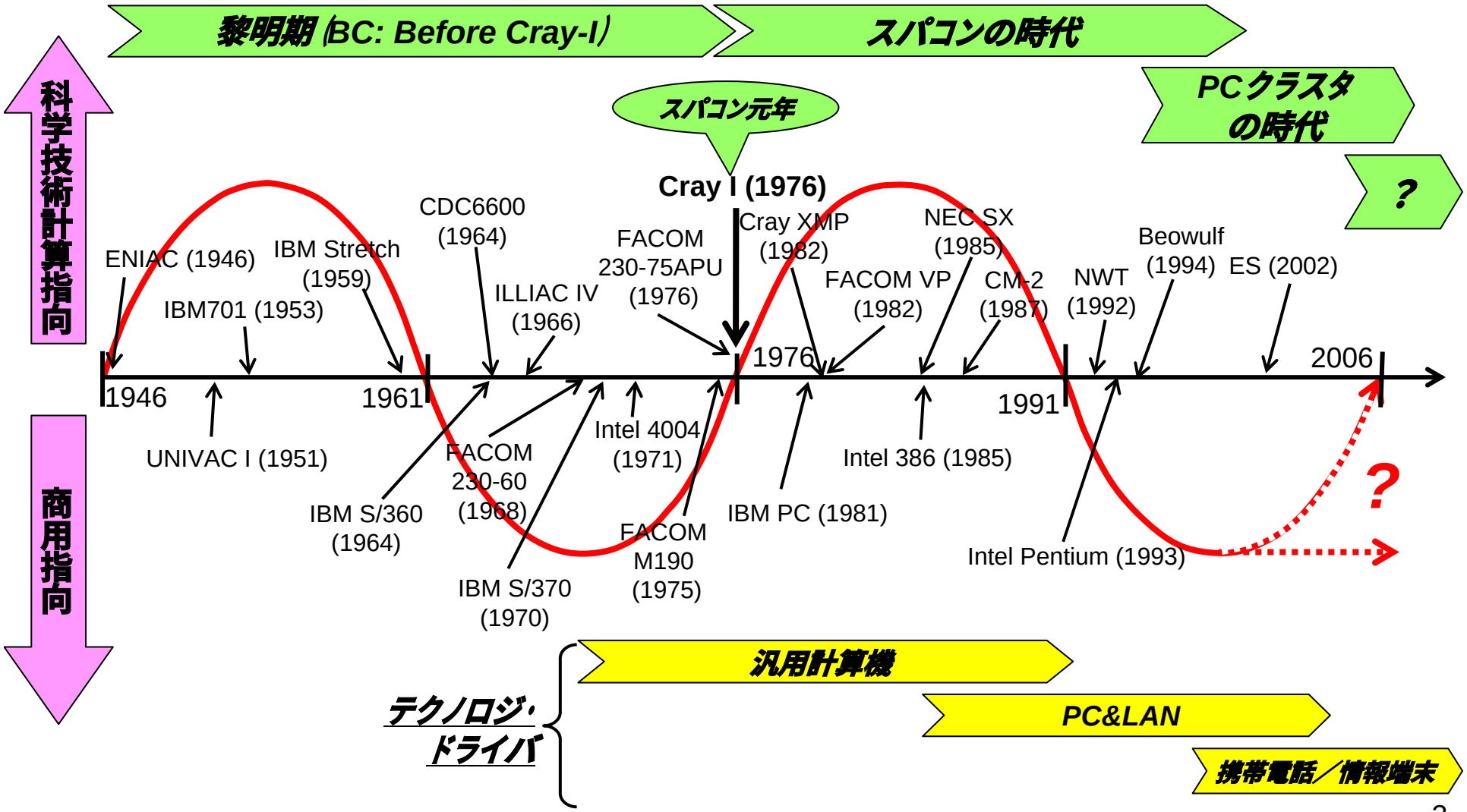
HPCアーキテクチャの今後

村上和彰
九州大学　教授
情報基盤センター長

HPCアーキテクチャの過去の推移

- **名称の推移**
  - Scientific computer
  - Supercomputer
  - HPC (High-performance computer/computing)
  - HEC (High-end computer/computing)
- **3つの視点から見た推移**
  - **商用コンピュータとのせめぎ合い**
  - **新規性／独創性と安定性／連続性との間のせめぎ合い**
  - **プロセッサとメモリとの間の性能トレードオフ**

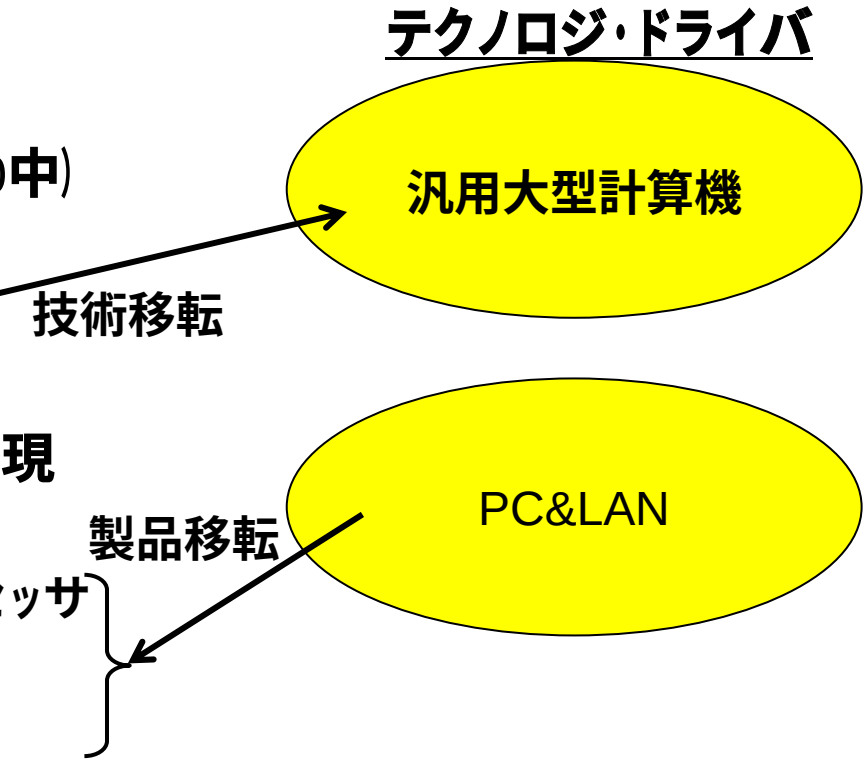
商用コンピュータとのせめぎ合い
～HPC業界におけるMakimoto's Wave～



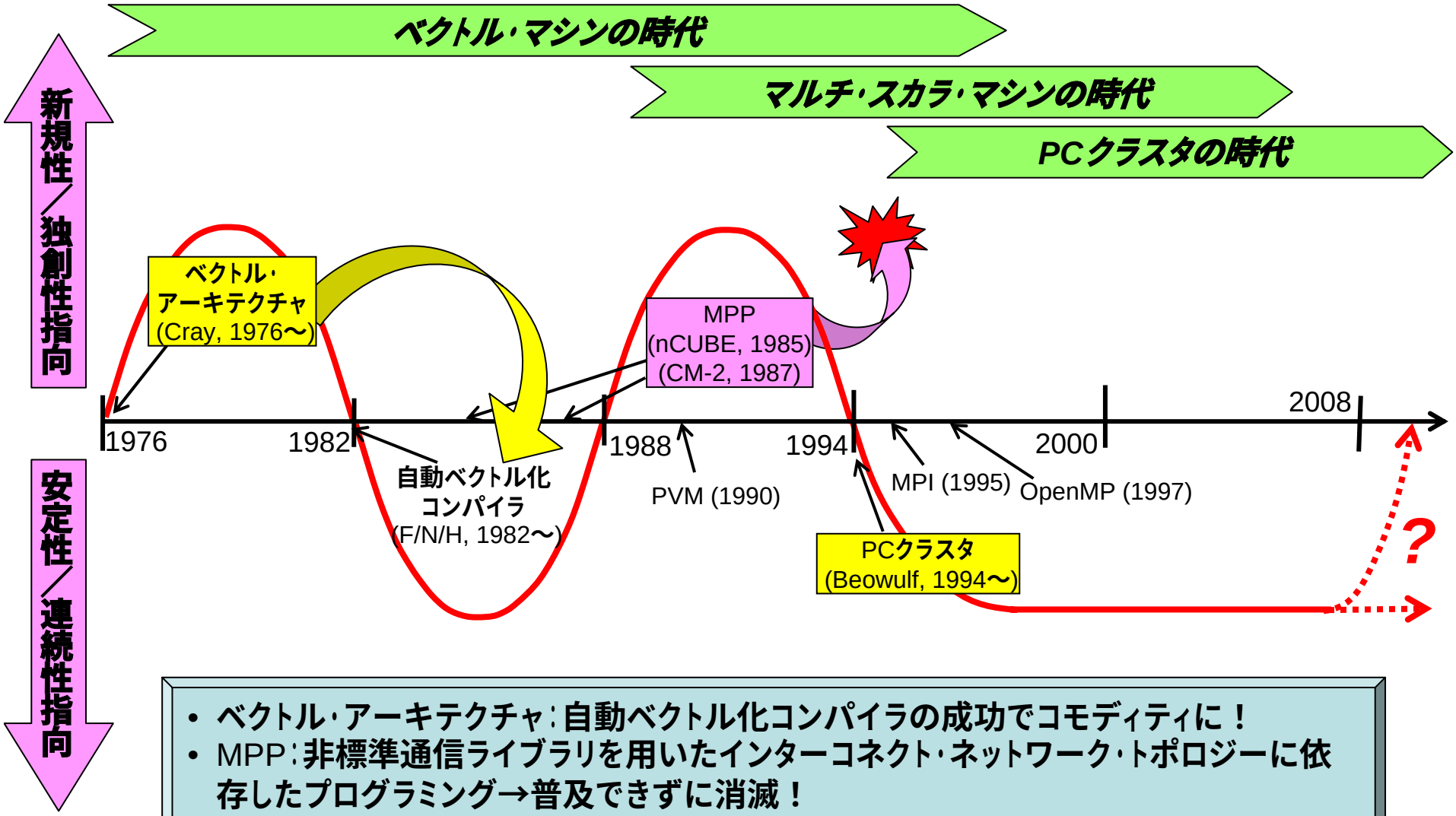
商用コンピュータとのせめぎ合い
～ビジネス･モデルの視点から～

HPCマシン

- **スパコン時代**（1980～1990**中**）
  - **集積回路技術**
  - **実装技術**
  - **コンパイラ技術**
- PC**クラスタ時代**（1990**中～現在**）
  - **高性能汎用マイクロプロセッサ**
  - **高集積メモリ**
  - **高速インターコネクト**
- **将来？**

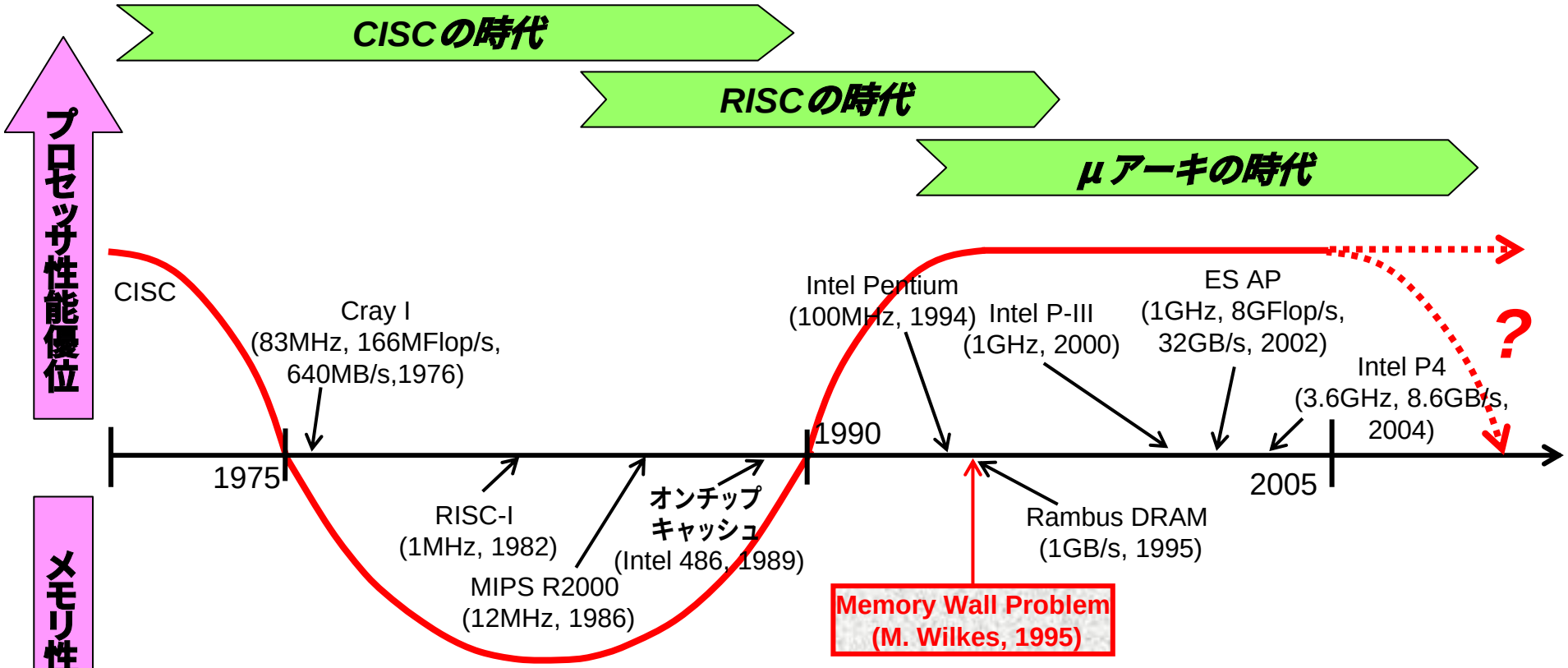


新規性／独創性と安定性／連続性との間のせめぎ合い



- ベクトル・アーキテクチャ：自動ベクトル化コンパイラの成功でコモディティに！
- MPP：非標準通信ライブラリを用いたインターコネクト・ネットワーク・トポロジーに依存したプログラミング→普及できずに消滅！

プロセッサとメモリとの間の性能トレードオフ



- メモリ（DRAM）性能：RISCおよびオンチップキャッシュが登場した1980年代のピークを境に，プロセッサ性能からの乖離が深刻化
  - そもそもRISCは「プロセッサ性能＝メモリ性能」を前提としたアーキテクチャ
- 今や命令セットアーキテクチャ（CISC vs. RISC）の時代ではなく，マイクロアーキテクチャ（OOOスーパースカラ，ディープパイプライン，SMT，等）の時代
  - 唯一の例外は，SIMD演算命令の成功

そもそもHPCアーキテクチャに対する要件は何なのか？

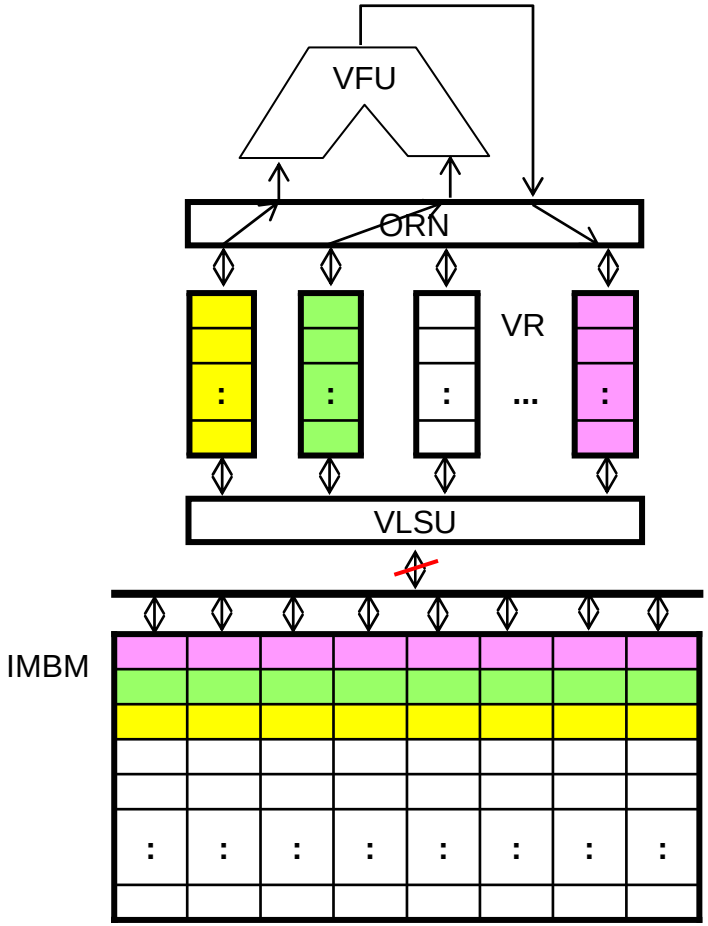
		対Non-HPC	対HPC
プロセッサ	演算	• 整数論理演算中心	• 浮動小数点演算中心 – (単精度はもちろん) 倍精度, 4倍精度 – (実数に加えて) 複素数 – 四則演算に加えて初等関数などの特殊演算
	制御	• IF文	• ループ構造
メモリ		• Pointer chasing • Garbage collection	• 大規模配列アクセス – 連続アクセス – ストライド・アクセス – 間接参照 (リスト・アクセス)
通信		• 同期 • 排他制御	• p-to-p通信 (大量データ転送) • Collective通信 – バリア同期 – リダクション
黄金律		1op : 1B/s : 1bps	1Flop : 4B/s : 4bps

これらの要件に関して
従来のHPCマシンはどう応えたか？

		要件	ベクトル・マシン	PCクラスタ
プロセッサ	演算	• 浮動小数点演算中心 – （単精度はもちろん）倍精度，4倍精度 – （実数に加えて）複素数 – 四則演算に加えて初等関数などの特殊演算	• ベクトル演算命令 –倍精度浮動小数点データを主対象とした四則演算 –さらに，平方根などの特殊演算もサポート –副作用として，命令供給ボトルネックを解消	• 高性能汎用マイクロプロセッサ – 浮動小数点四則演算ユニット（倍精度サポート） – SIMD演算命令を追加して短ベクトル演算をサポート
	制御	• ループ構造		• 分岐予測，OOO実行による制御依存ペナルティの隠蔽
メモリ		• 大規模配列アクセス – 連続アクセス – ストライド・アクセス – 間接参照（リスト・アクセス）	• ベクトル・レジスタ＋ベクトル・ロード／ストア命令 – リスト・ベクトル・サポート • マルチバンク＆インターリーブ・メモリ	• 大容量キャッシュ・メモリ – ノンブロッキング – 多階層（L1～L3） • HW/SWプリフェッチ機能
通信		• p-to-p通信（大量データ転送） • Collective通信 – バリア同期 – リダクション	• クロスバー・ネットワーク等の非（低）閉塞網による低レイテンシ＆高スループット通信 • バリア同期，collective通信専用の別ネットワーク装備	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造
黄金律		1Flop : 4B/s : 4bps	• ES) 1Flop:4B/s:3bps	• P4) 1Flop:2.4B/s:?

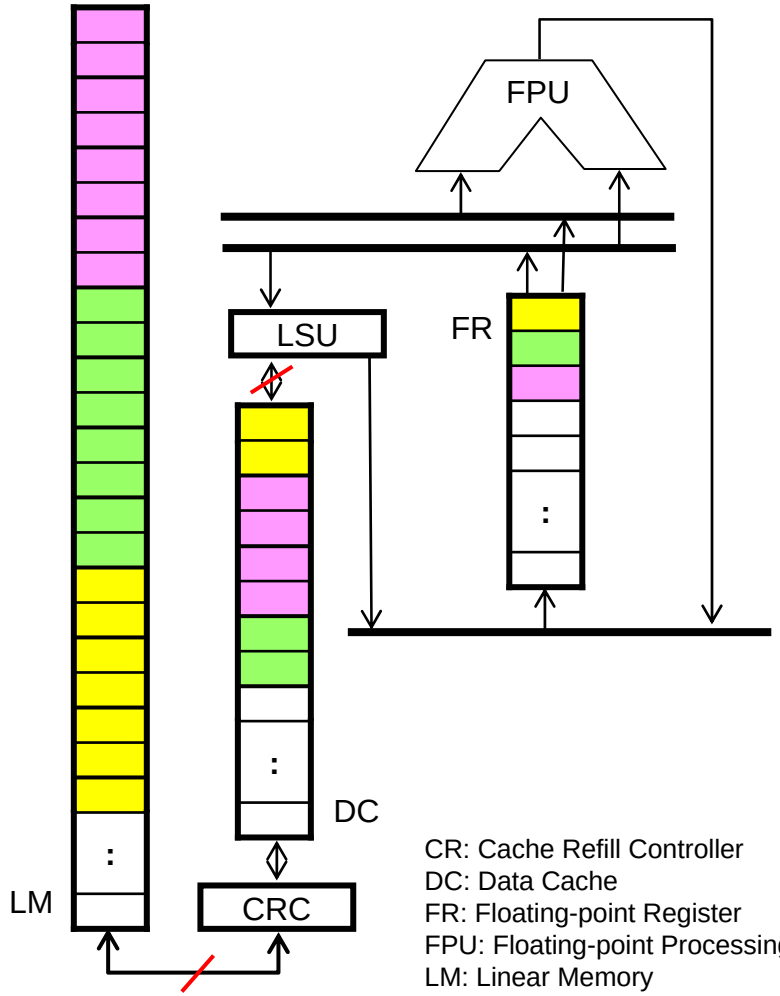
ベクトル・プロセッサ vs. スカラ・プロセッサ

- ベクトル・プロセッサ



IMBM: Interleaved Multi-Bank Memory
ORN: Operand Routing Network
VFU: Vector Functional Unit
VLSU: Vector Load/Store Unit
VR: Vector Register

- スカラ・プロセッサ



CR: Cache Refill Controller
DC: Data Cache
FR: Floating-point Register
FPU: Floating-point Processing Unit
LM: Linear Memory
LSU: Load/Store Unit

2010年のHPCアーキテクチャは？ ～前提／制約条件～

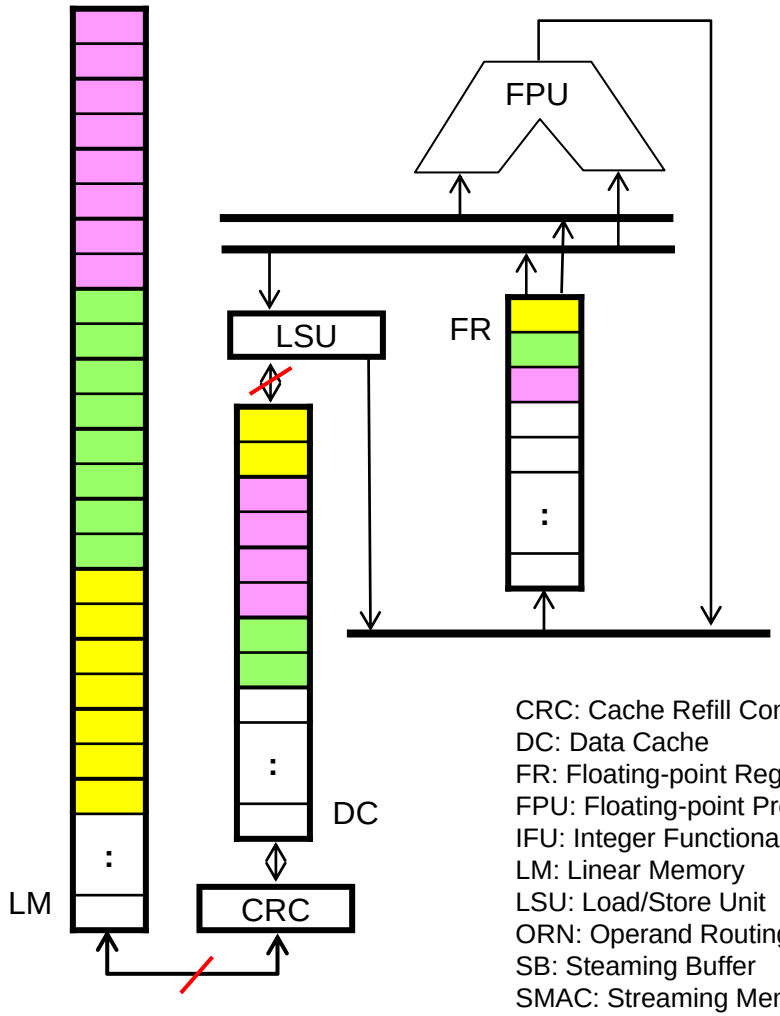
- ビジネス的に成り立たなければならない
  - 何をテクノロジ・ドライバとするか？
- ソフトウェア開発者に対して，現在の「安定性／連続性」を引き続き保証しなければならない
  - MPPの時のような「フィーバー」はもう懲り懲り
- (45nmまで) CMOSのクロック速度の伸びは鈍化するものの，集積度の伸びはまだまだ堅調
  - クロック速度の伸びに頼るのではなく，トランジスタ・バジェットに頼る
    - トランジスタ・バジェットを如何に性能に還元するか？
  - 「Memory Wall」問題はさらに深刻化
    - メモリ・アクセスするより (無駄な計算でも) 計算した方が良い!
  - 消費電力＆発熱問題も同様に深刻化

HPCアーキに対する要件に関して 2010年のHPCマシンはどう応えるか？

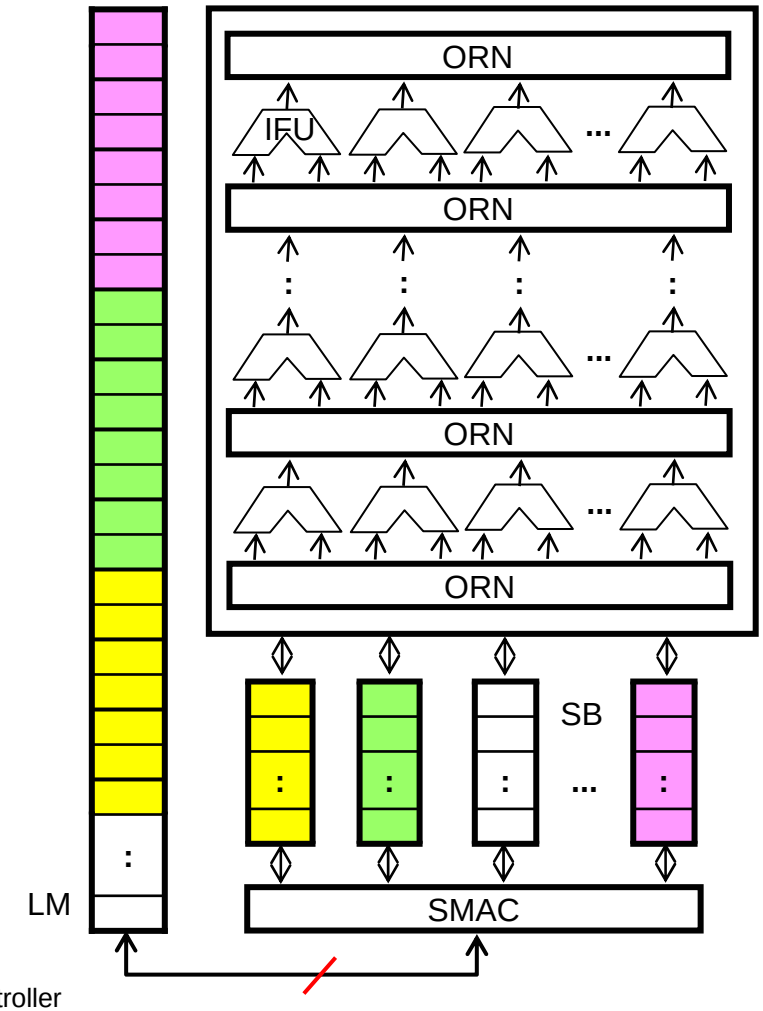
		要件	2004年のPCクラスタ	2010年のHPCマシン（私案）
プロセッサ	演算	• 浮動小数点演算中心 – （単精度はもちろん）倍精度，4倍精度 – （実数に加えて）複素数 – 四則演算に加えて初等関数などの特殊演算	• 高性能汎用マイクロプロセッサ –浮動小数点四則演算ユニット（倍精度サポート） –SIMD演算命令を追加して短ベクトル演算をサポート	• 高性能＆低消費電力組込みプロセッサ • 動的再構成可能データパス – 2次元ALUアレイで浮動小数点四則演算（倍精度，4倍精度，複素数）を実現 – ループボディのデータフローをデータパス（HW）として直接実装 – ストリーミング・データ供給 – 結果的に，命令＆データ供給ボトルネックを解消
	制御	• ループ構造	• 分岐予測，OOO実行による制御依存ペナルティの隠蔽	
メモリ		• 大規模配列アクセス – 連続アクセス – ストライド・アクセス – 間接参照（リスト・アクセス）	• 大容量キャッシュ・メモリ – ノンブロッキング – 多階層（L1～L3） • HW/SWプリフェッチ機能	• 大容量キャッシュ・メモリ • メモリ・アクセス・ヘルパー – プリフェッチ機能 – データ・ストリーミング機能 – 「ならず者」ロード対策
通信		• p-to-p通信（大量データ転送） • Collective通信 – バリア同期 – リダクション	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造 • NIC／スイッチでcollective通信サポート
黄金律		1Flop : 4B/s : 4bps		• 1Flop:1B/s:1bps

スカラ･プロセッサ vs. 動的再構成可能データパス

- スカラ･プロセッサ



- 動的再構成可能データパス



CRC: Cache Refill Controller
DC: Data Cache
FR: Floating-point Register
FPU: Floating-point Processing Unit
IFU: Integer Functional Unit
LM: Linear Memory
LSU: Load/Store Unit
ORN: Operand Routing Network
SB: Steaming Buffer
SMAC: Streaming Memory Access Controller

動的再構成可能データパスの応用例
～分子軌道法における二電子積分計算（$\mu\nu \| \lambda\sigma$）の場合～

```
tei(4,4,4,4)=(((3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*f(0,t))/(p**2
*q**2)+(4*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*PQx*(QCx+QDx)*(3+2*q*QCx*QDx)*f(1,t))/(p*q*(p+q))(4*(PAx+PBx)*(3+2*
p*PAx*PBx)*PQx*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*f(1,t))/(p*q*(p+q))(8*(PAx+PBx)*(3+2*p*PAx*PBx)*(QCx+QDx)*(3
+2*q*QCx*QDx)*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p*q*(p+q)**2)+(2*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3+q*(QCx**
2+4*QCx*QDx+QDx**2))*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p*q**2*(p+q)**2)+(2*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*(3+2*q*(4*QCx*QDx
+QDx**2+QCx**2*(1+2*q*QDx**2)))*(((p+q)*f(1,t))+2*p*PQx**2*q*f(2,t)))/(p**2*q*(p+q)**2)+(4*(3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*
PBx**2)))*PQx*(QCx+QDx)*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(q*(p+q)**3)¥+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(3+
2*q*QCx*QDx)*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(p*(p+q)**3)(8*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(3+q*(QCx**2+4*QCx*QDx+QDx**2))
*(3*(p+q)*f(2,t)+2*p*PQx**2*q*f(3,t)))/(q*(p+q)**3)(4*(PAx+PBx)*PQx*(3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(3*(p+q)*f(2,t)
+2*p*PQx**2*q*f(3,t)))/(p*(p+q)**3)+((3+2*p*(4*PAx*PBx+PBx**2+PAx**2*(1+2*p*PBx**2)))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)
+p*PQx**2*q*f(4,t))))/(q**2*(p+q)**4)(8*(PAx+PBx)*(3+2*p*PAx*PBx)*(QCx+QDx)*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2
*q*f(4,t))))/(q*(p+q)**4)(8*(PAx+PBx)*(QCx+QDx)*(3+2*q*QCx*QDx)*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*PQx**2*q*f(4,t))))/(p*
(p+q)**4)+(4*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*P
Qx**2*q*f(4,t))))/(p*q*(p+q)**4)+((3+2*q*(4*QCx*QDx+QDx**2+QCx**2*(1+2*q*QDx**2)))*(3*(p+q)**2*f(2,t)+4*p*PQx**2*q*(3*(p+q)*f(3,t)+p*
PQx**2*q*f(4,t))))/(p**2*(p+q)**4)(4*p*(PAx+PBx)*(3+2*p*PAx*PBx)*PQx*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t
))))/(q*(p+q)**5)+(8*(3+p*(PAx**2+4*PAx*PBx+PBx**2))*PQx*(QCx+QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t
))))/(p+q)**5+(4*PQx*q*(QCx+QDx)*(3+2*q*QCx*QDx)*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t))))/(p*(p+q)**5)(8*(
PAx+PBx)*PQx*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(15*(p+q)**2*f(3,t)+4*p*PQx**2*q*(5*(p+q)*f(4,t)+p*PQx**2*q*f(5,t))))/(p+q)**5+(8*(PA
x+PBx)*(QCx+QDx)*(15*(p+q)**3*f(3,t)+30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))8*p**3*PQx**6*q**3*f(6,t)))/(p+q)**6+(2*(3
+p*(PAx**2+4*PAx*PBx+PBx**2))*(15*(p+q)**3*f(3,t)30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))+8*p**3*PQx**6*q**3*f(6,t)))/
(q*(p+q)**6)+(2*(3+q*(QCx**2+4*QCx*QDx+QDx**2))*(15*(p+q)**3*f(3,t)30*p*PQx**2*q*(p+q)*(3*(p+q)*f(4,t)+2*p*PQx**2*q*f(5,t))+8*p**3*P
Qx**6*q**3*f(6,t)))/(p*(p+q)**6)
```

➔ **787 MUL, 261 ADD, 69 FUNC**

```
tei(3,1,1,1)=((PAy*(1+2*p*PAx*PBx)*(1+2*q*QCx*QDx)*f(0,t))/q+((((p+q)**4*((PAy+PQy)*q*(1+2*q*QCx*QDx)+p*(PAy+2*PAx*PAy*PQx*q+2*PAy*P
Bx*PQx*q+2*PAx*PBx*PQy*q2*PAx*PAy*q*QCx2*PAy*PBx*q*QCx2*PAy*PQx*q*QCx+2*q*((PAy*(PAx+PBx+PQx))+2*(PAy*(PAx+PBx)*PQx+PAx*P
Bx*PQy)*q*QCx)*QDx)2*p**2*PAx*PAy*PBx*(1+2*PQx*q*(QCx+QDx)))*f(1,t))/q)+(p+q)*((p+q)*((p+q)*(3*p*PAy+6*p**2*PAx*PAy*PQx+6*p**2*PAy*
PBx*PQx+2*p**2*PAy*PQx**2+4*p**3*PAx*PAy*PBx*PQx**2+p*PQy+2*p**2*PAx*PBx*PQy+2*p*PAy*PQx**2*q+PQy*q+2*p*PAx*PQx*PQy*q
```

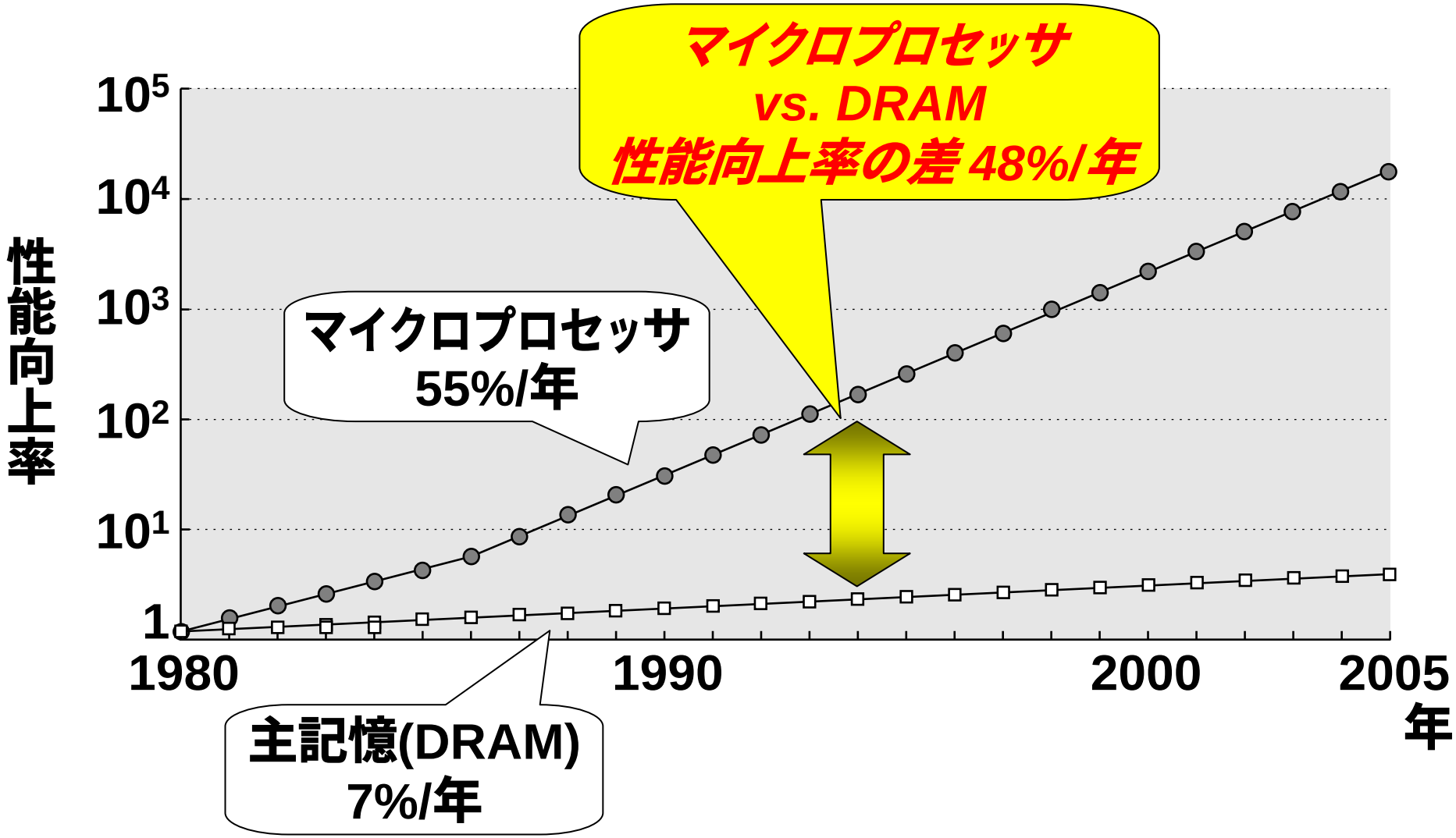
➔ **116 MUL, 31 ADD, 2 FUNC**

【出典：青柳 睦 教授（九大）】

HPCアーキに対する要件に関して 2010年のHPCマシンはどう応えるか？

		要件	2004年のPCクラスタ	2010年のHPCマシン（私案）
プロセッサ	演算	• 浮動小数点演算中心 – （単精度はもちろん）倍精度，4倍精度 – （実数に加えて）複素数 – 四則演算に加えて初等関数などの特殊演算	• 高性能汎用マイクロプロセッサ –浮動小数点四則演算ユニット（倍精度サポート） –SIMD演算命令を追加して短ベクトル演算をサポート	• 高性能＆低消費電力組込みプロセッサ • 動的再構成可能データパス – 2次元ALUアレイで浮動小数点四則演算（倍精度，4倍精度，複素数）を実現 – ループボディのデータフローをデータパス（HW）として直接実装 – ストリーミング・データ供給 – 結果的に，命令＆データ供給ボトルネックを解消
	制御	• ループ構造	• 分岐予測，OOO実行による制御依存ペナルティの隠蔽	
メモリ		• 大規模配列アクセス – 連続アクセス – ストライド・アクセス – 間接参照（リスト・アクセス）	• 大容量キャッシュ・メモリ – ノンブロッキング – 多階層（L1～L3） • HW/SWプリフェッチ機能	• 大容量キャッシュ・メモリ • メモリ・アクセス・ヘルパー – プリフェッチ機能 – データ・ストリーミング機能 – 「ならず者」ロード対策
通信		• p-to-p通信（大量データ転送） • Collective通信 – バリア同期 – リダクション	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造 • NIC／スイッチでcollective通信サポート
黄金律		1Flop : 4B/s : 4bps		• 1Flop:1B/s:1bps

「Memory Wall」問題



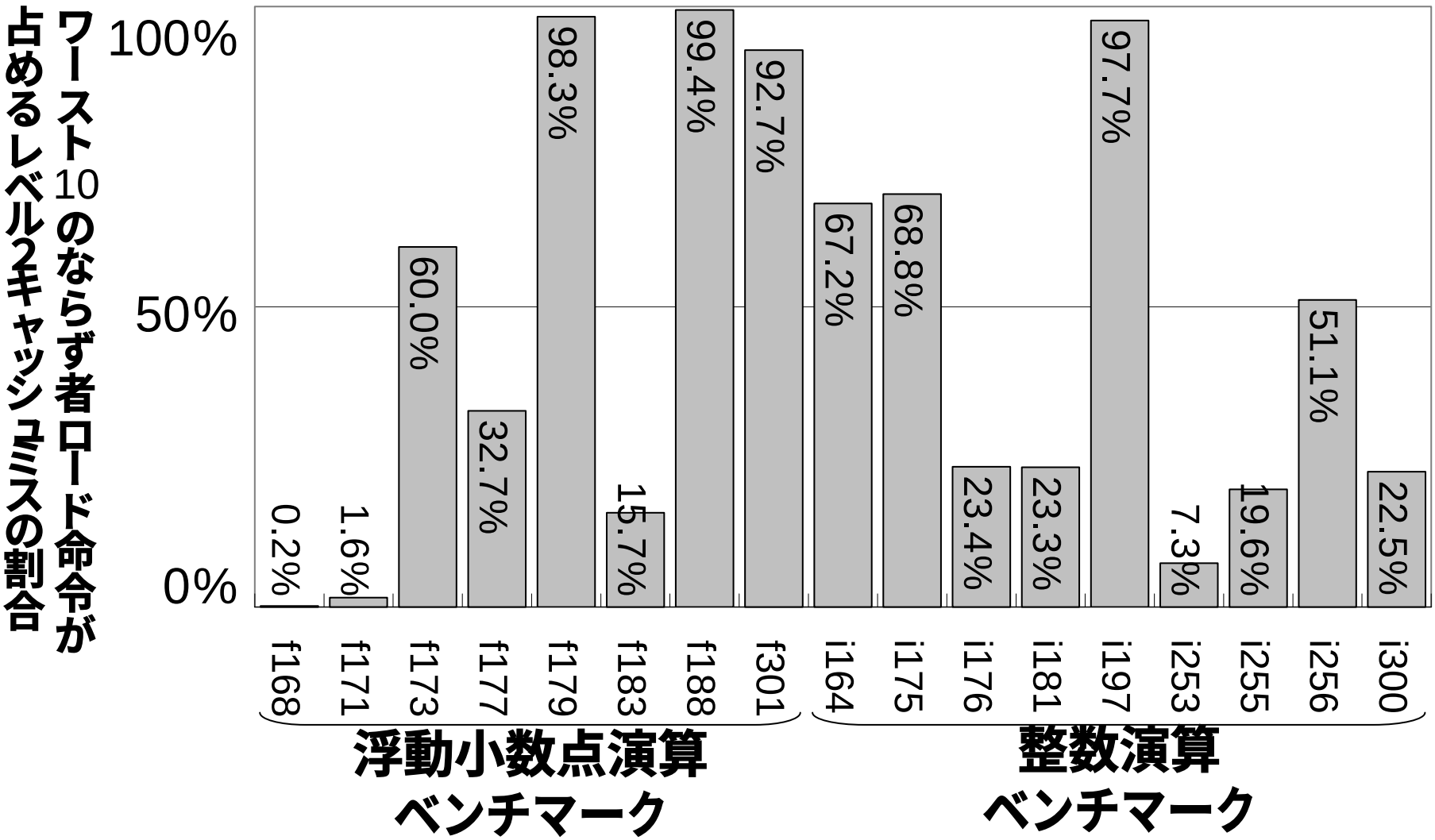
【出典】J.L. Hennessy, D.A. Patterson, *Computer Architecture: A Quantitative Approach 3rd Edition*, Fig. 5.2

「Memory Wall」問題への対策

- 2重の問題：スループットとレイテンシ
- 抜本的解決策
  - プロセッサ-DRAM混載
    - PPRAM, IRAM, PIM, ...
    - しかし，容量不足でオフチップメモリ・アクセスが生じる際には解にならず
- スループット対策
  - スループット向上
    - ハイバンドDRAM：Rambus DRAM, SDRAM, ...
    - データ・ストリーミング
- レイテンシ対策
  - （見た目の）レイテンシ削減
    - キャッシュメモリ，プリフェッチ，…
  - レイテンシ隠蔽
    - ノンブロッキング・キャッシュ，マルチスレッディング，…

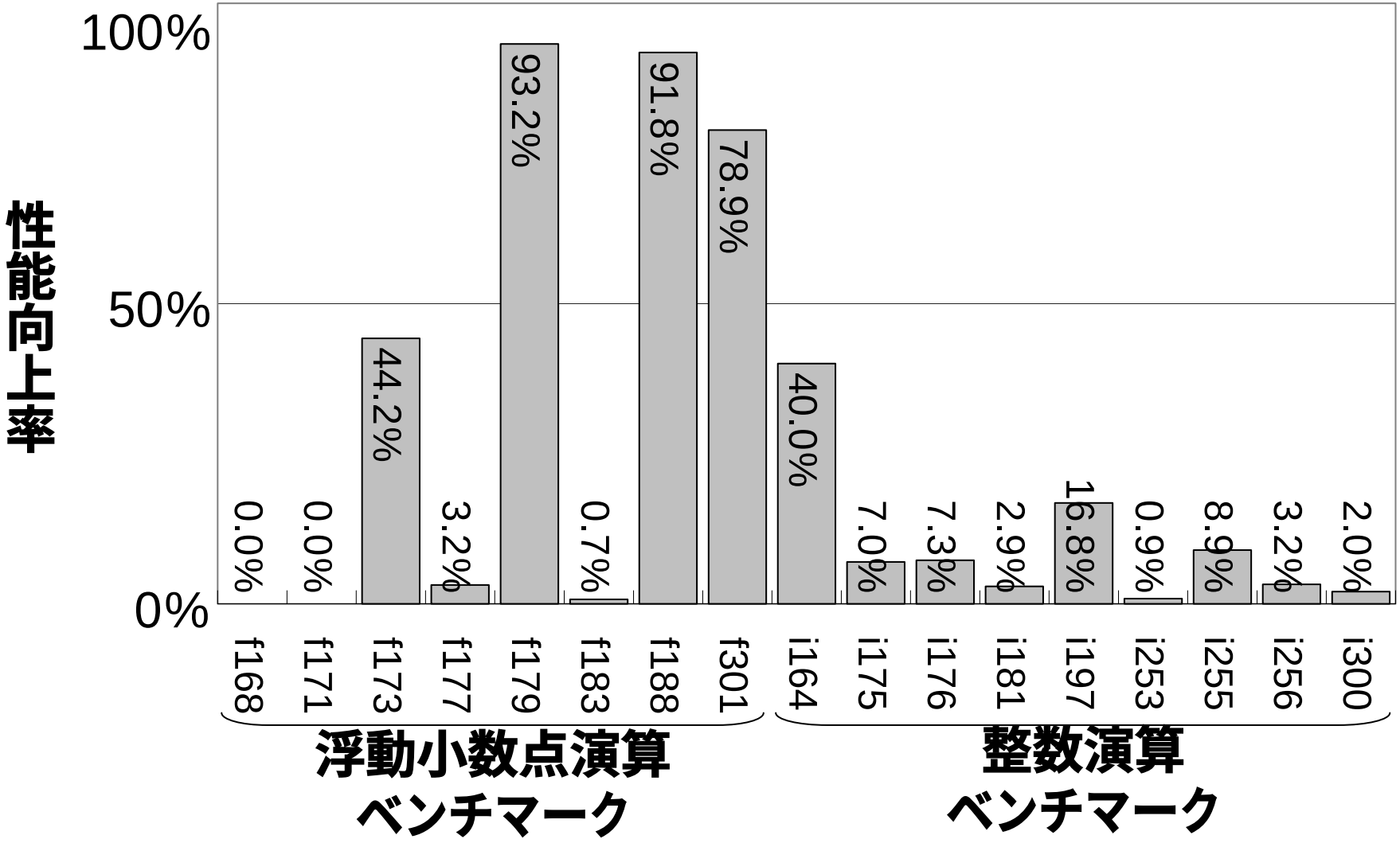
キャッシュミス率の低減が鍵!

「ならず者」ロード命令の存在
～キャッシュメモリの仇～

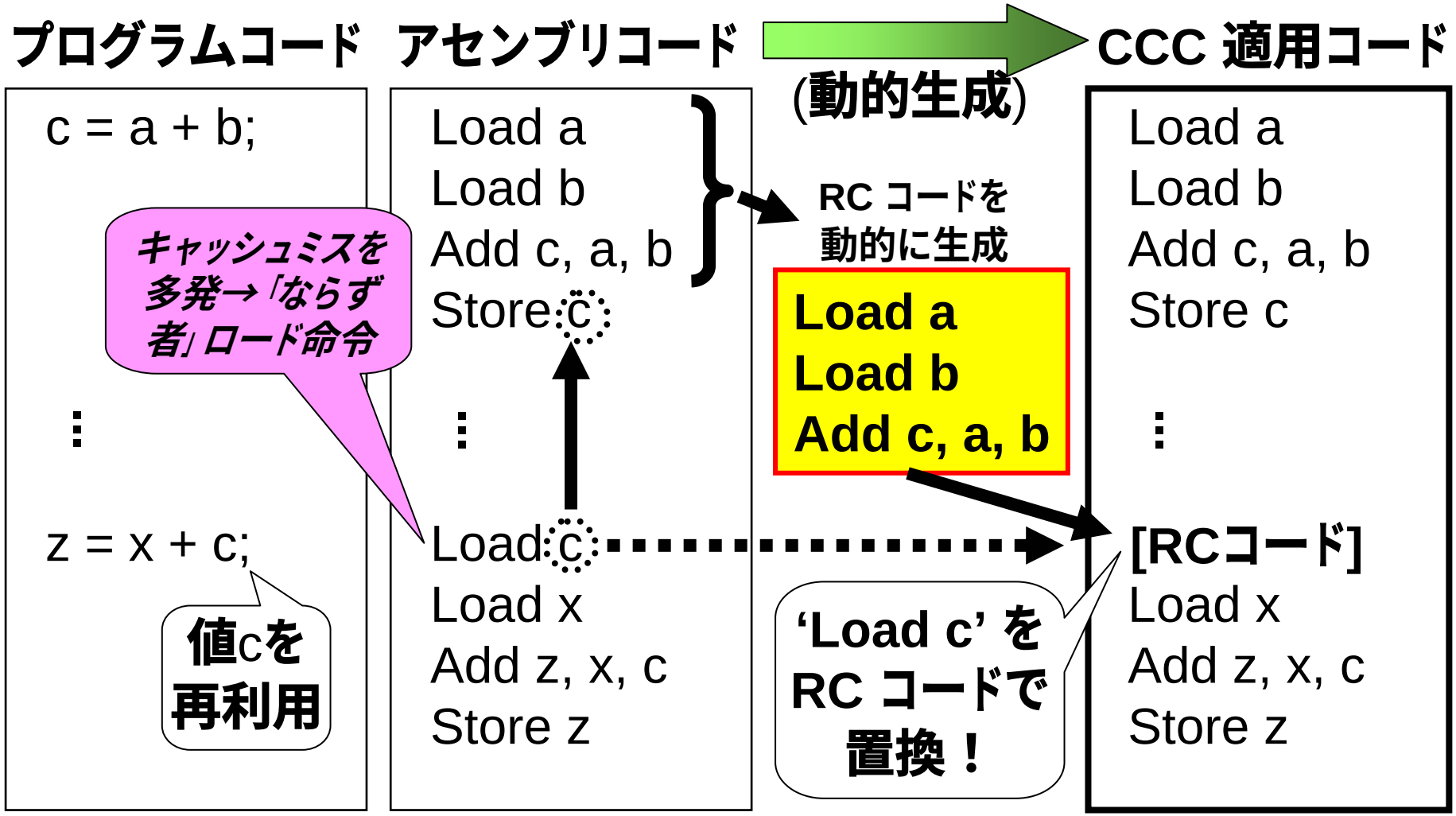


SPEC CPU 2000 benchmarks on Simplescalar 3.0d. (50M Forward, 10M Exec, L1=32k, L2=128k.)

メモリ・アクセス・ヘルパー
～「ならず者」ロード命令を退治出来たら～



CCC (Computing Centric Computation) ～RC (Re-Computation)～

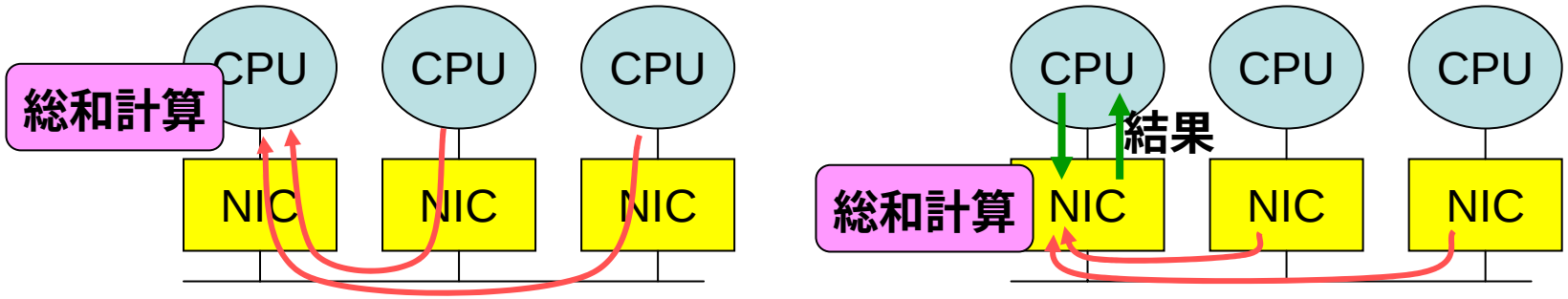


HPCアーキに対する要件に関して 2010年のHPCマシンはどう応えるか？

		要件	2004年のPCクラスタ	2010年のHPCマシン（私案）
プロセッサ	演算	• 浮動小数点演算中心 – （単精度はもちろん）倍精度，4倍精度 – （実数に加えて）複素数 – 四則演算に加えて初等関数などの特殊演算	• 高性能汎用マイクロプロセッサ –浮動小数点四則演算ユニット（倍精度サポート） –SIMD演算命令を追加して短ベクトル演算をサポート	• 高性能＆低消費電力組込みプロセッサ • 動的再構成可能データパス – 2次元ALUアレイで浮動小数点四則演算（倍精度，4倍精度，複素数）を実現 – ループボディのデータフローをデータパス（HW）として直接実装 – ストリーミング・データ供給 – 結果的に，命令＆データ供給ボトルネックを解消
	制御	• ループ構造	• 分岐予測，OOO実行による制御依存ペナルティの隠蔽	
メモリ		• 大規模配列アクセス – 連続アクセス – ストライド・アクセス – 間接参照（リスト・アクセス）	• 大容量キャッシュ・メモリ – ノンブロッキング – 多階層（L1～L3） • HW/SWプリフェッチ機能	• 大容量キャッシュ・メモリ • メモリ・アクセス・ヘルパー – プリフェッチ機能 – データ・ストリーミング機能 – 「ならず者」ロード対策
通信		• p-to-p通信（大量データ転送） • Collective通信 – バリア同期 – リダクション	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造	• Myrinet等の高スループット＆低レイテンシ網（ただし，閉塞網）による階層バス構造 • NIC／スイッチでcollective通信サポート
黄金律		1Flop : 4B/s : 4bps		• 1Flop:1B/s:1bps

MPI Collective通信のオフローディング

- NIC (Network I/F Card)／スイッチ上でcollective通信サポート
  - 通信コストの低減
  - CPUやメモリバスの有効利用



- 実装例
  - 汎用インターコネクト
    - Myrinet, InfiniBand, Gigabit Ether, QS-Net, ...
  - 専用インターコネクト
    - ES, BlueGene/L, ...

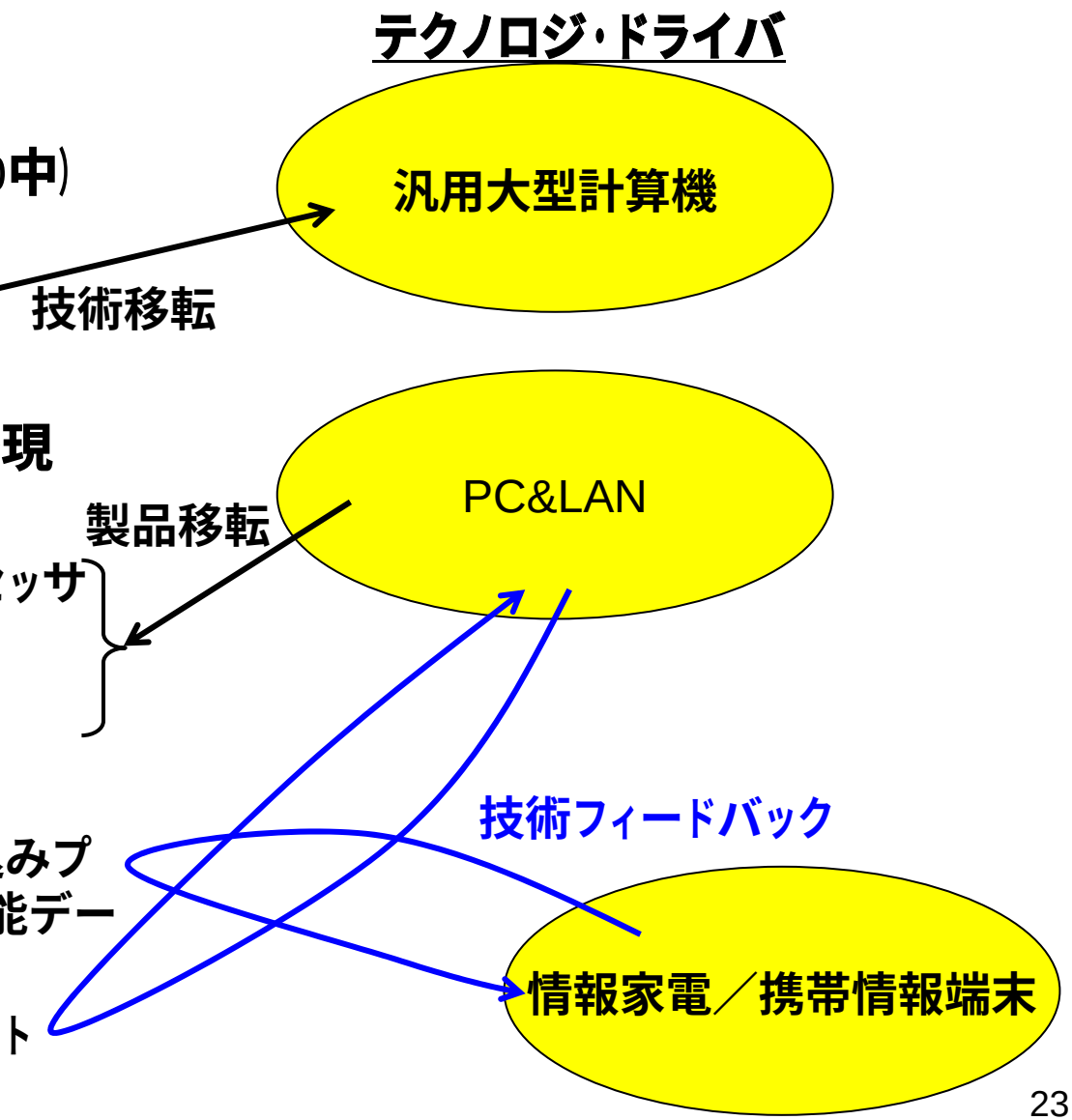
2010年のHPCアーキテクチャは？～検証～

- ビジネス的に成り立たなければならない
  - 何をテクノロジ・ドライバとするか？
- ソフトウェア開発者に対して，現在の「安定性／連続性」を引き続き保証しなければならない
  - MPPの時のような「フィーバー」はもう懲り懲り
- (45nmまで) CMOSのクロック速度の伸びは鈍化するものの，集積度の伸びはまだまだ堅調
  - クロック速度の伸びに頼るのではなく，トランジスタ・バジェットに頼る
    - トランジスタ・バジェットを如何に性能に還元するか？
  - 「Memory Wall」問題はさらに深刻化
    - メモリ・アクセスするより (無駄な計算でも) 計算した方が良い!
  - 消費電力＆発熱問題も同様に深刻化

ビジネス的にどう成り立たせるか？

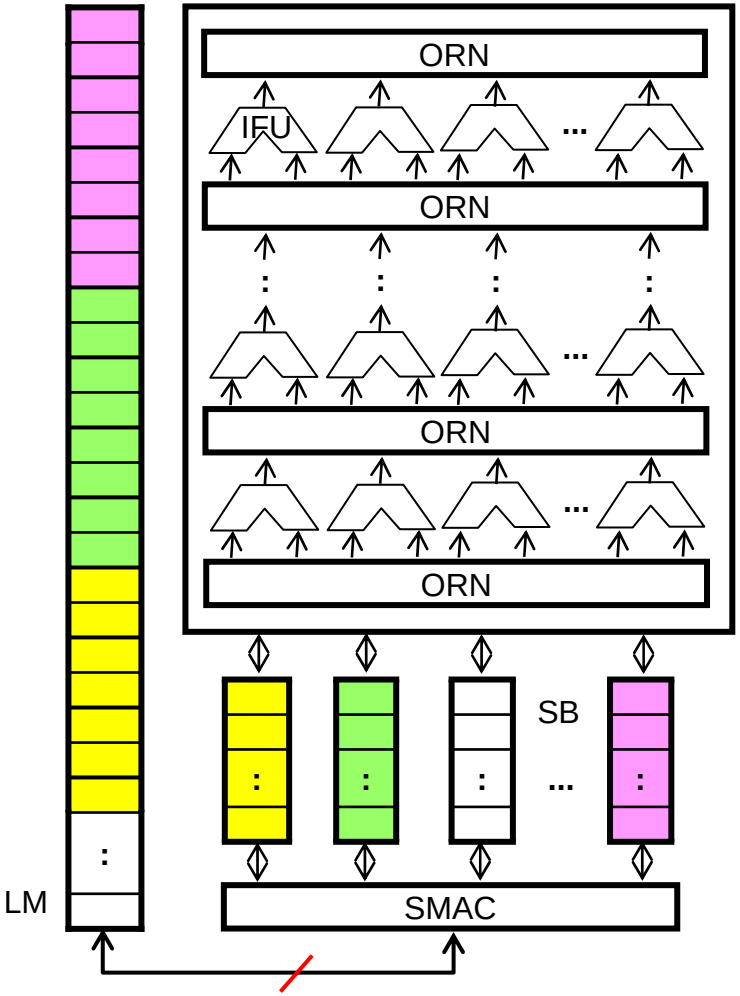
HPCマシン

- **スパコン時代**（1980～1990**中**）
  - 集積回路技術
  - 実装技術
  - コンパイラ技術
- PC**クラスタ時代**（1990**中～現在**）
  - 高性能汎用マイクロプロセッサ
  - 高集積メモリ
  - 高速インターコネクト
- 2010**年の**HPC**マシン**
  - 高性能＆低消費電力組込みプロセッサ＋動的再構成可能データパス
  - 高速＆知的インターコネクト

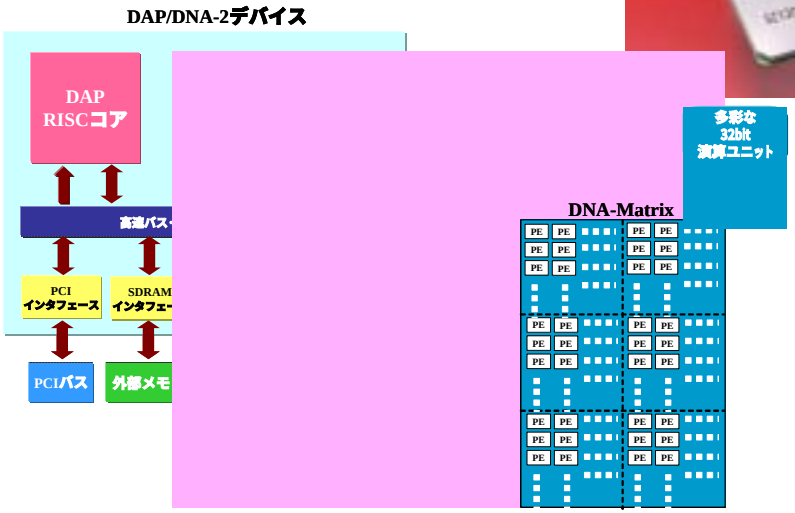


動的再構成可能データパス
～ビジネス的に成り立つのか？～

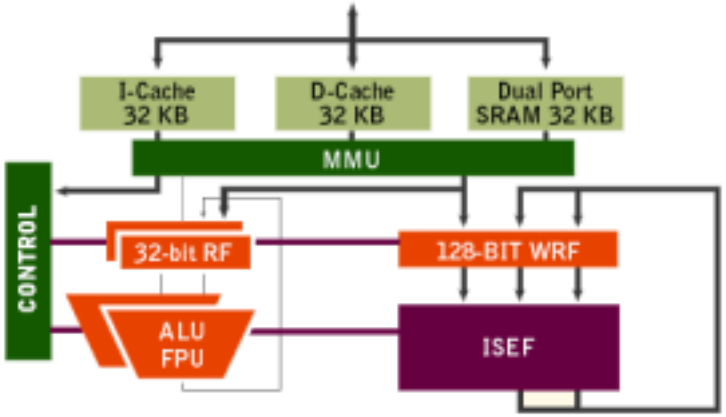
- 動的再構成可能データパス



- IP Flex DAP/DNA

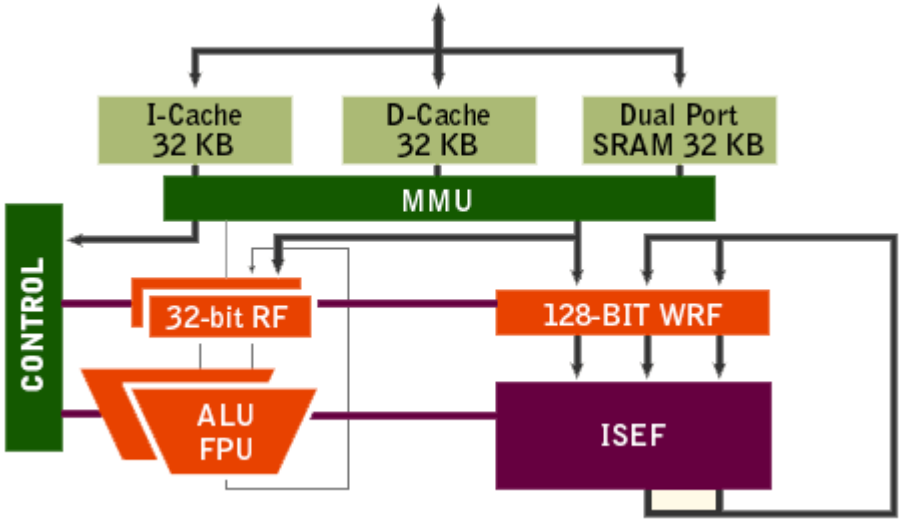


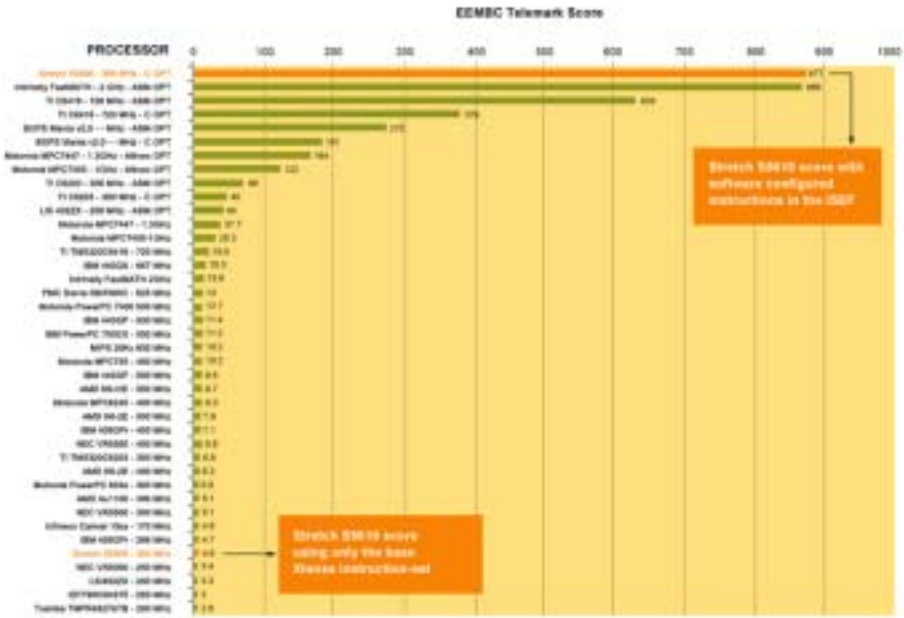
- Stretch



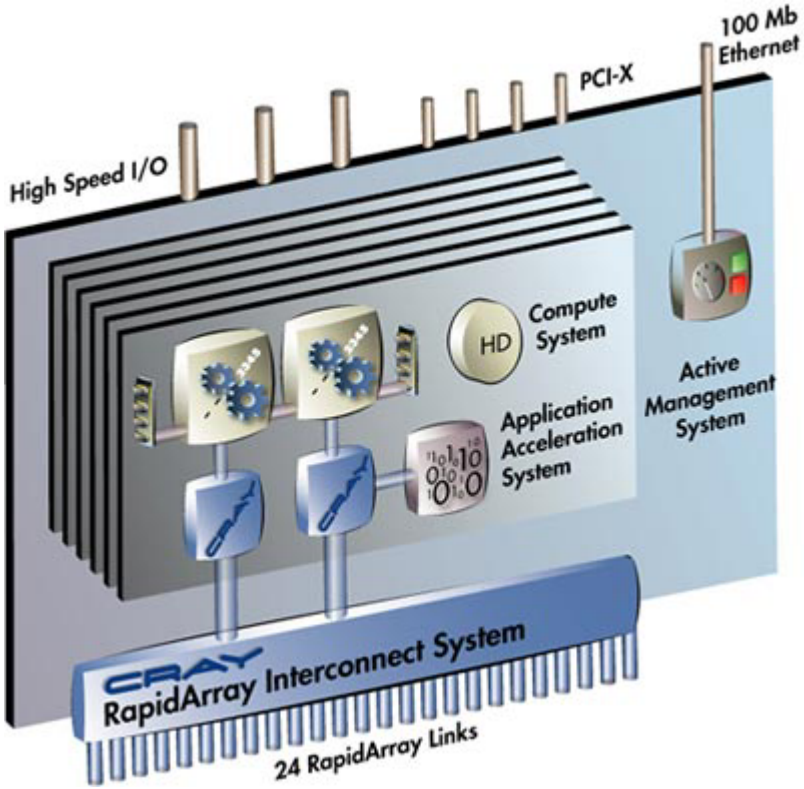
Stretch

- 300 MHz, 32-bit Xtensa-based processor
- 16- and 24-bit instructions
- FPU
- MMU with TLB
- Stretch Instruction Set Extension Fabric
  - Aligned load and store
    - 8, 16, 32, 64, and 128 bit
  - Unaligned load and store
    - Up to 16 bytes variable byte streaming I/O
    - Up to 32 bits variable bit streaming I/O
- User-defined extensions to the core ISA
  - Defined in C/C++
  - Fully pipelined and interlocked
- Low power consumption
- Support for standard operating systems





CRAY XD1



	Chassis	Each Rack
Compute Processors	12	144
Performance	53 GFlop/s	633 GFlop/s
Aggregate Switching Capacity	96 GB/s	1152 GB/s
Interprocessor Latency	1.6 us	1.8 us
Aggregate Memory Bandwidth	77 GB/s	922 GB/s
Maximum Memory	96 GB	1152 GB
Maximum Disk Storage	296 GB	