フローセントリックコンピューティング

国立研究開発法人 産業技術総合研究所
情報技術研究部門
高野 了成

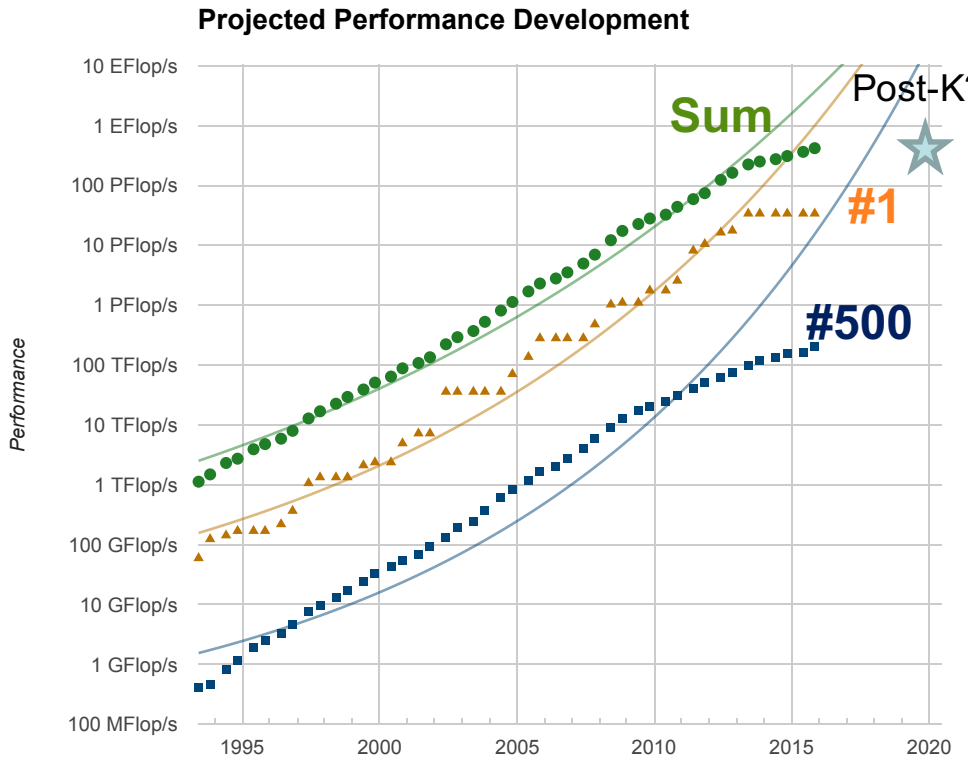
2016年6月20日　産総研STARシンポジウム@コクヨホール

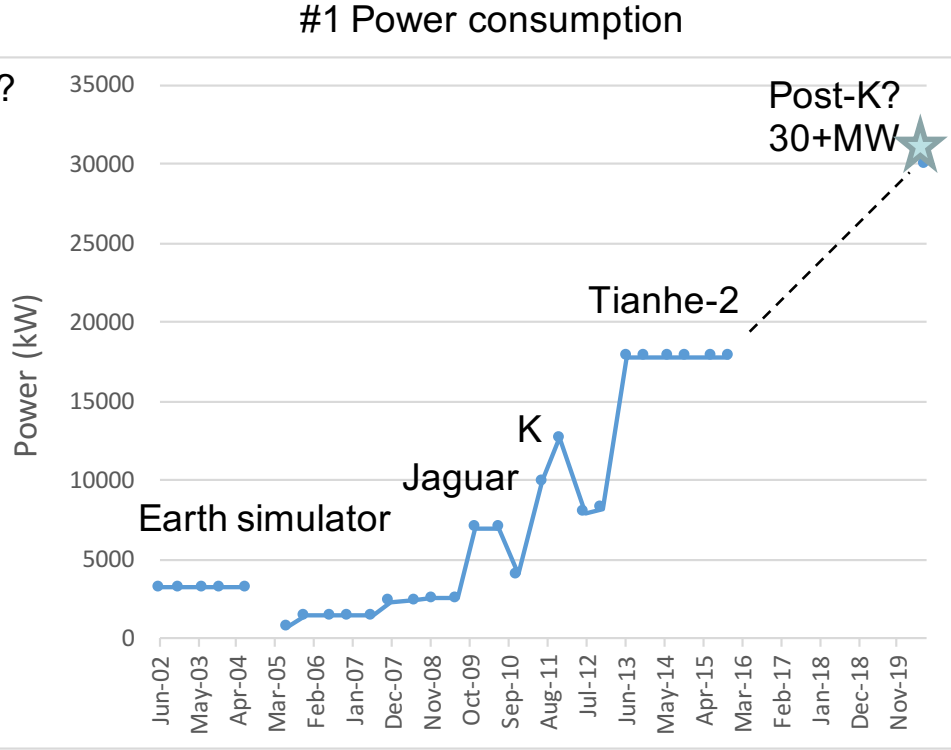
発表の流れ

- フローセントリックコンピューティング～ポストムーア時代に向けたデータセンターアーキテクチャ～
- 要素技術開発
  - アプリケーション中心型OSデプロイメントシステム
  - ハイパーバイザ型仮想化を用いたハイブリッドメモリシステム
- 実用化に向けた今後の取り組み

なぜ「高電力効率」が必要か？

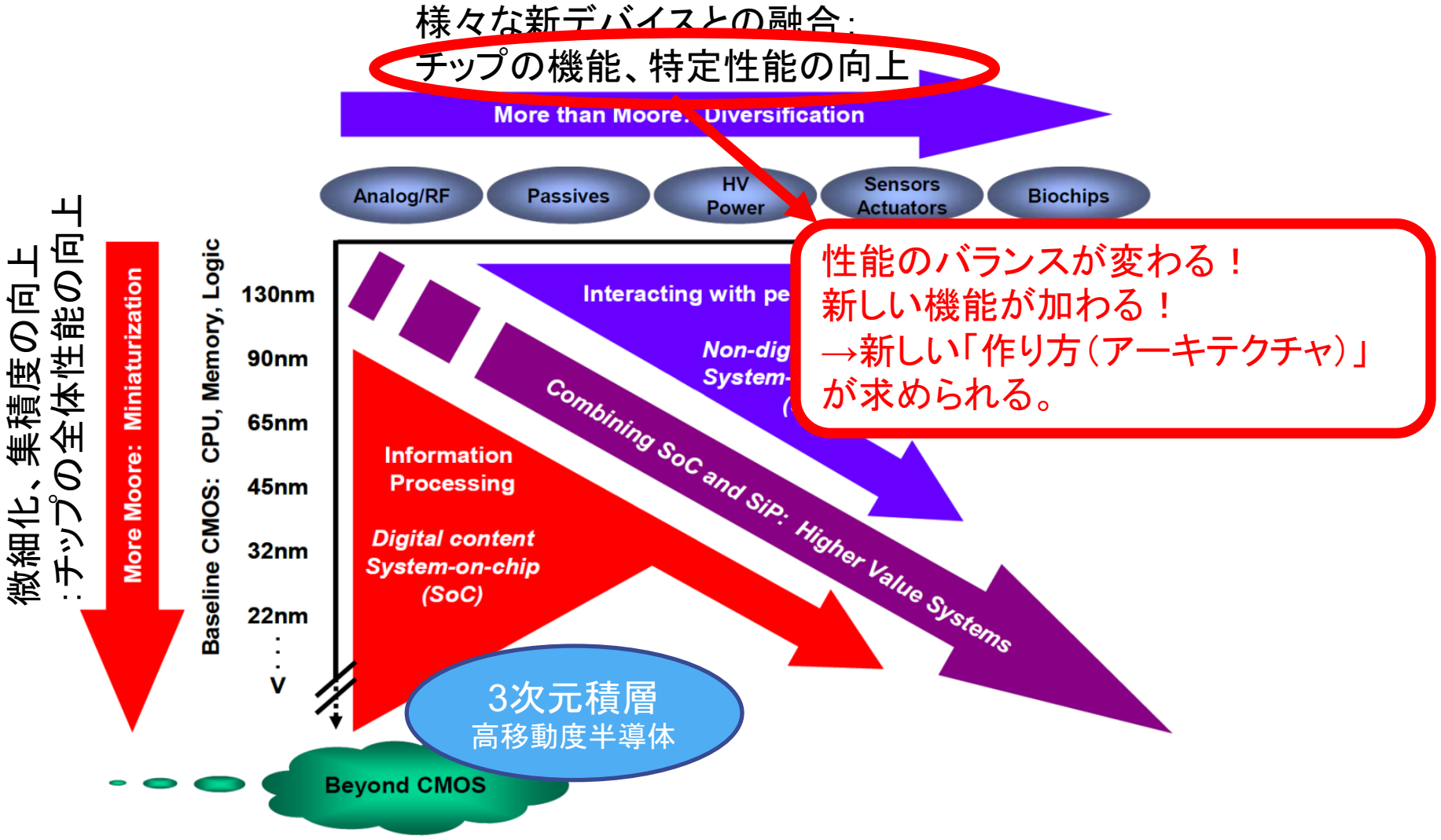
- 電力の壁（Power Wall）問題
  - 2～3倍の電力で数10～100倍の性能向上





source: http://top500.org/

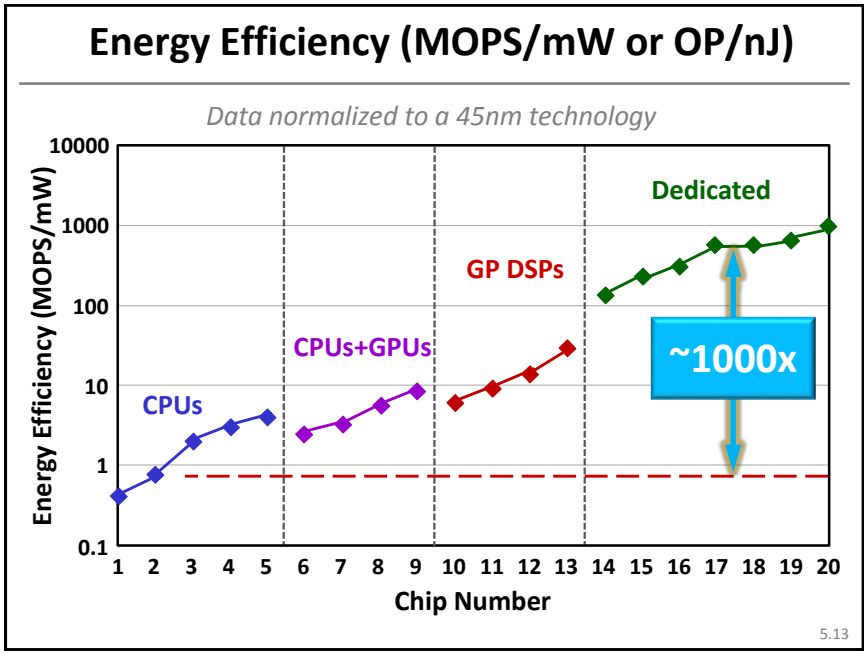
More MooreとMore than Moore



Source: Fig.3 of “More-than-Moore White Paper,” ITRS white paper, 2010

ポストムーア時代の計算機

- 専用回路が陳腐化しない
  - 深層学習用チップ
  - イジングマシン
  - ニューロチップ
- 2次元から3次元へ
  - CMOS積層、TSV
  - メモリの大容量化・広帯域化
  - 光通信によるIOの広帯域化



Markovic@UCLA

省電力を意識したデータ処理

今後の計算機はBYTE/FLOP比が一転増加するように進化（「FLOPSからBYTE中心へ」by松岡先生@東工大）

- データ移動のコスト大
  - ①データを極力動かさないで処理
- データ移動のコスト小
  - ②計算を再利用
  - ③専用エンジンにデータ移動して処理

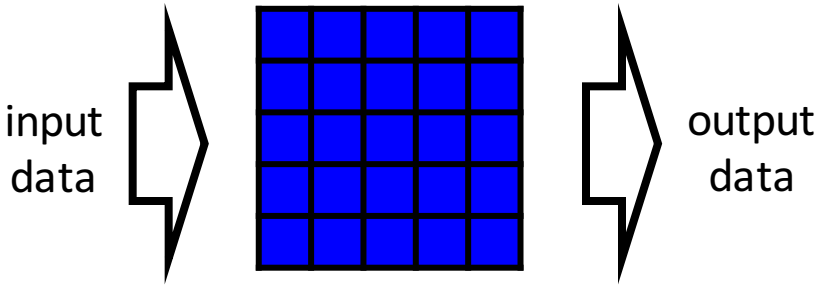
使い分けが必要

ポストムーア時代の計算パラダイム

- 省電力かつ高速データ移動が可能になれば、
  - 現在：データのある場所で計算
  - 提案：機能がある場所にデータを移動

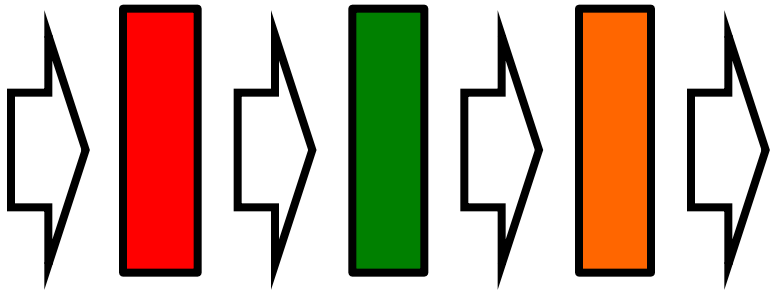
Data affinity Processing

moving computation to data



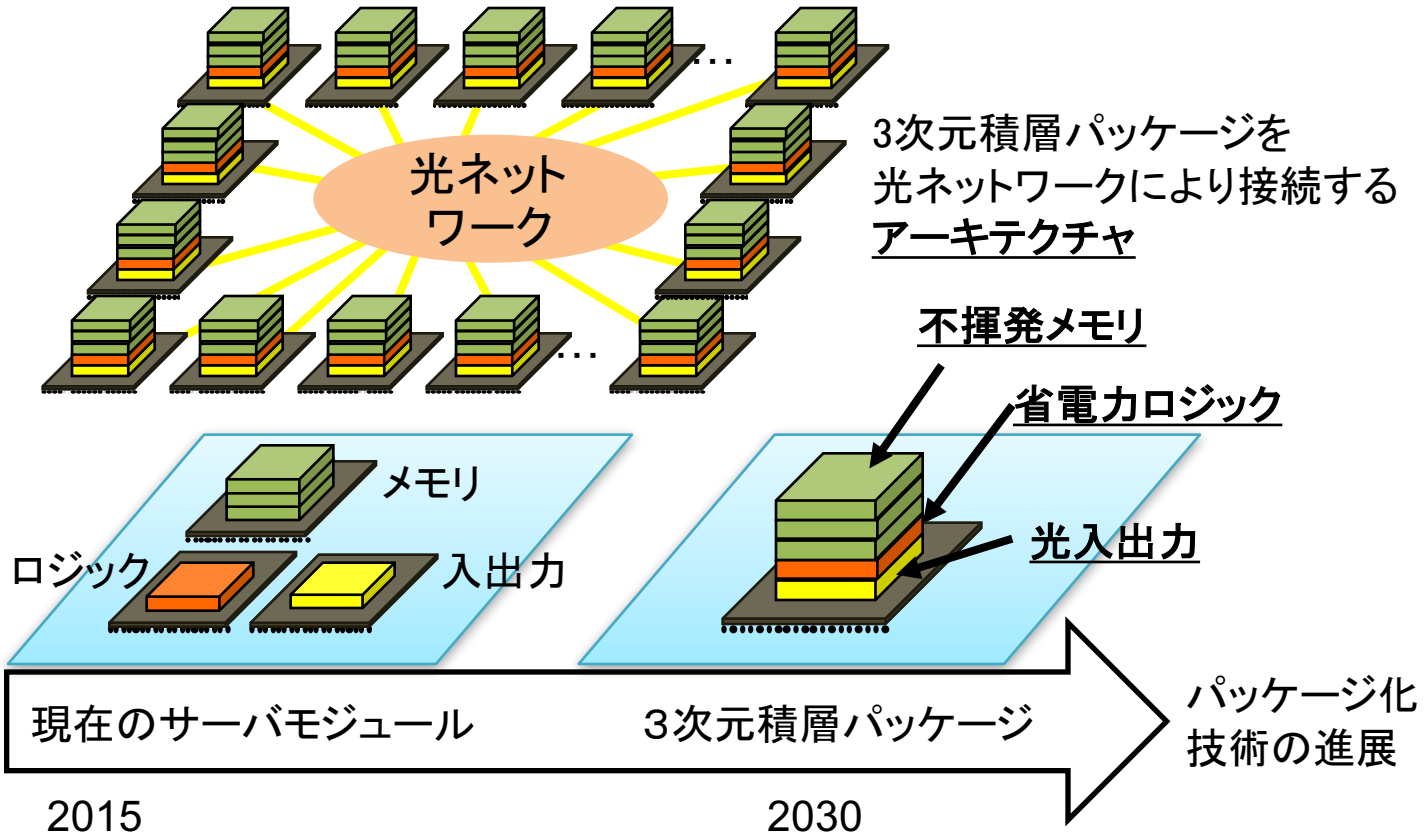
Function affinity Processing

moving data to computation

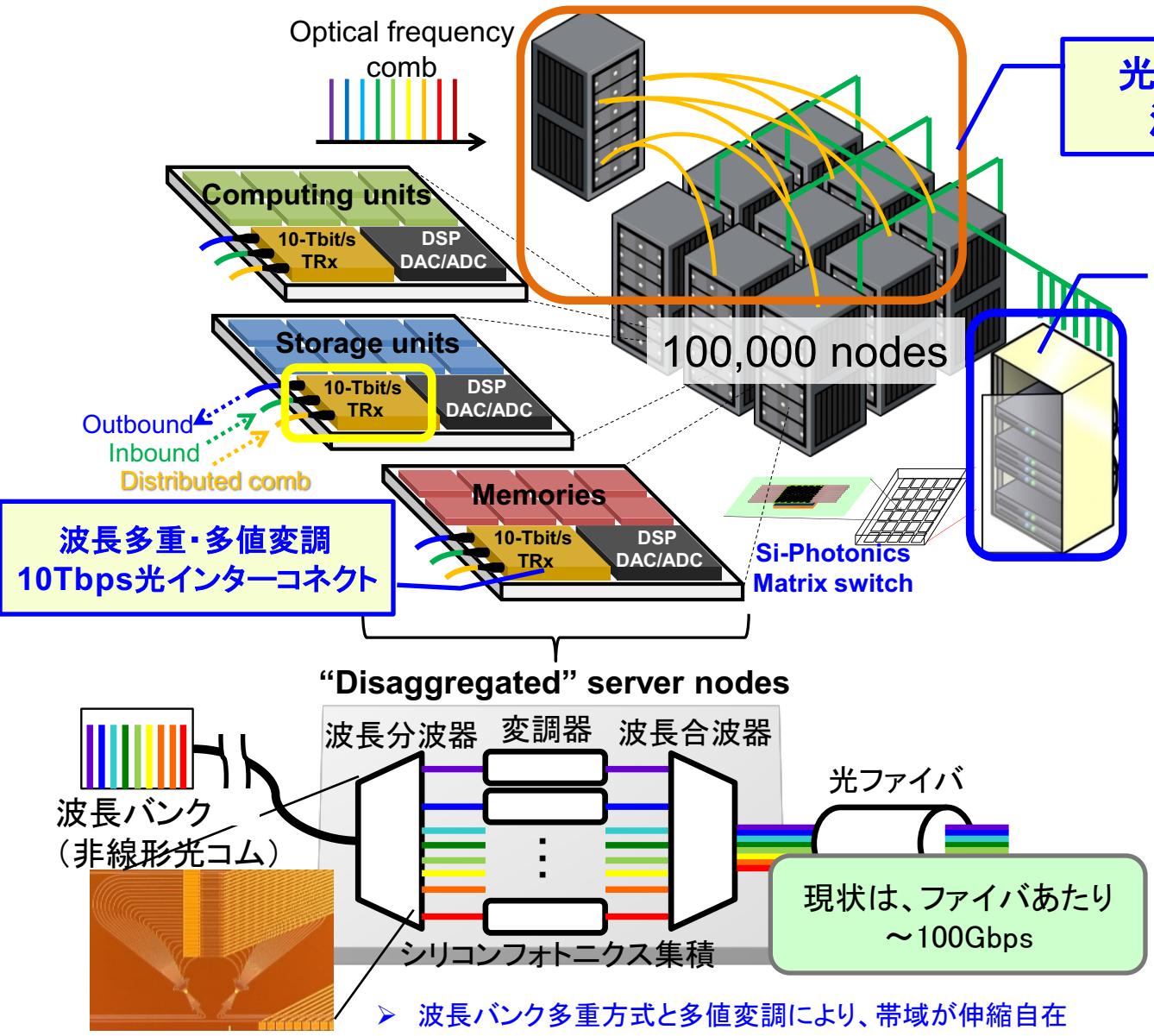


将来のデータセンタアーキテクチャ

専用プロセッサやメモリ、ストレージを光ネットワークで接続する包括的な計算機システムアーキテクチャが必要



参考：光ネットワーク技術の活用



現状電気スイッチ容量：
～1 Pbps → Ebps級へ

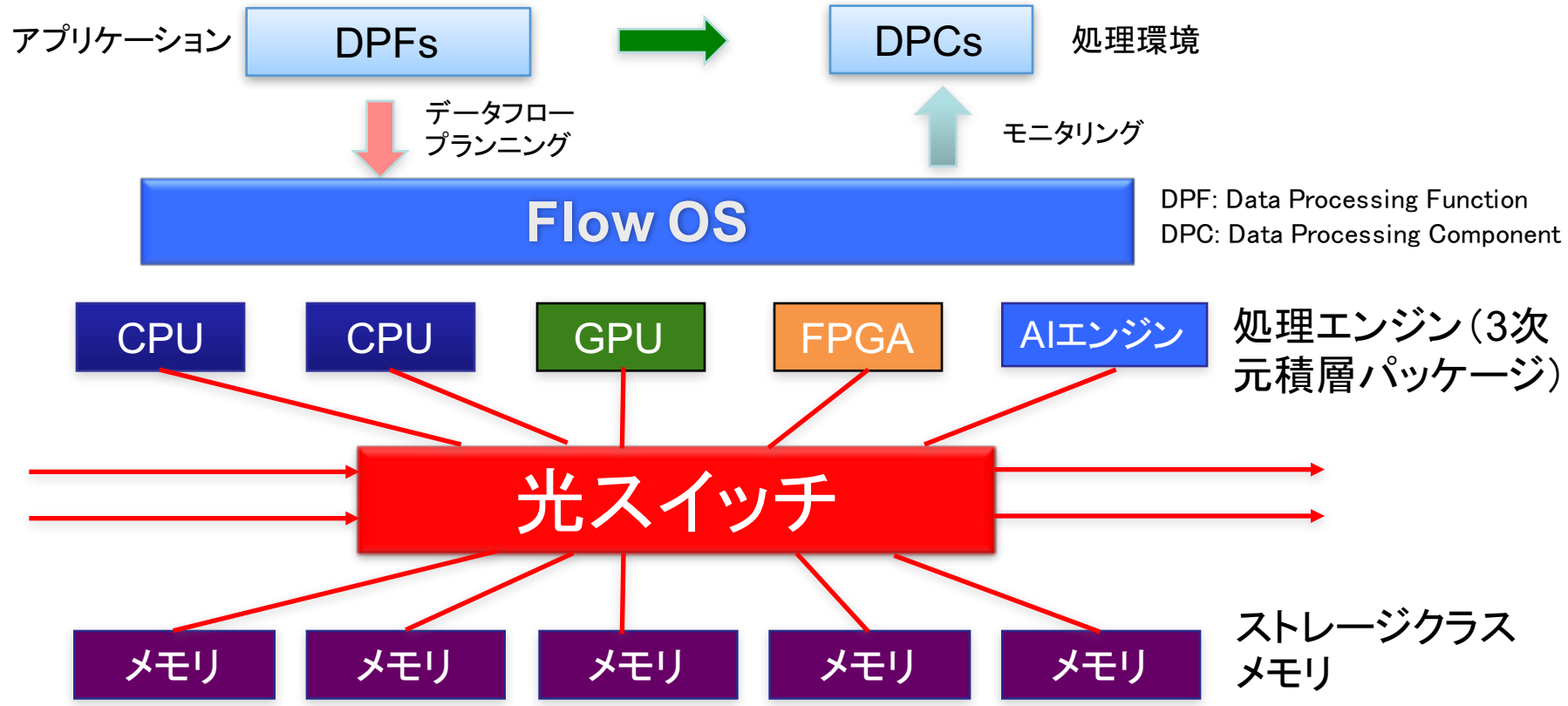
Baud rate	2偏波 QAM数	総帯域 Tbps
20 G	64	7.68
20 G	256	10.24
32 G	64	12.29
56 G	16	14

※波長数：32に対する

- 波長バンク多重方式と多値変調により、帯域が伸縮自在

フローセントリックコンピューティングの提案

- レゴブロックのようにアプリケーション毎に必要な部品（処理エンジン、ストレージクラスメモリ）を適材適所で組み合わせたシステム構成を動的に構築
- データセンタ全体を一つのオペレーティングシステム(OS)で効率的に運用・管理



関連プロジェクト

- 従来型アーキテクチャ
  - All-in-oneサーバの集合→性能面、管理コスト面でスケールしない
- ディスアグリゲーテッド・アーキテクチャ
  - 各機能ブロックを分離（ディスアグリゲート）し、それぞれを広帯域光インタコネクトで接続
  - 用途に応じて構成を動的に再構成
  - コンセプト先行で、技術は未開発

Intel Rack Scale Architecture

HP The Machine

UCB FireBox

発表の流れ

- フローセントリックコンピューティング～ポストムーア時代に向けたデータセンターアーキテクチャ～
- 要素技術開発
  - アプリケーション中心型OSデプロイメントシステム
  - ハイパーバイザ型仮想化を用いたハイブリッドメモリシステム
- 実用化に向けた今後の取り組み

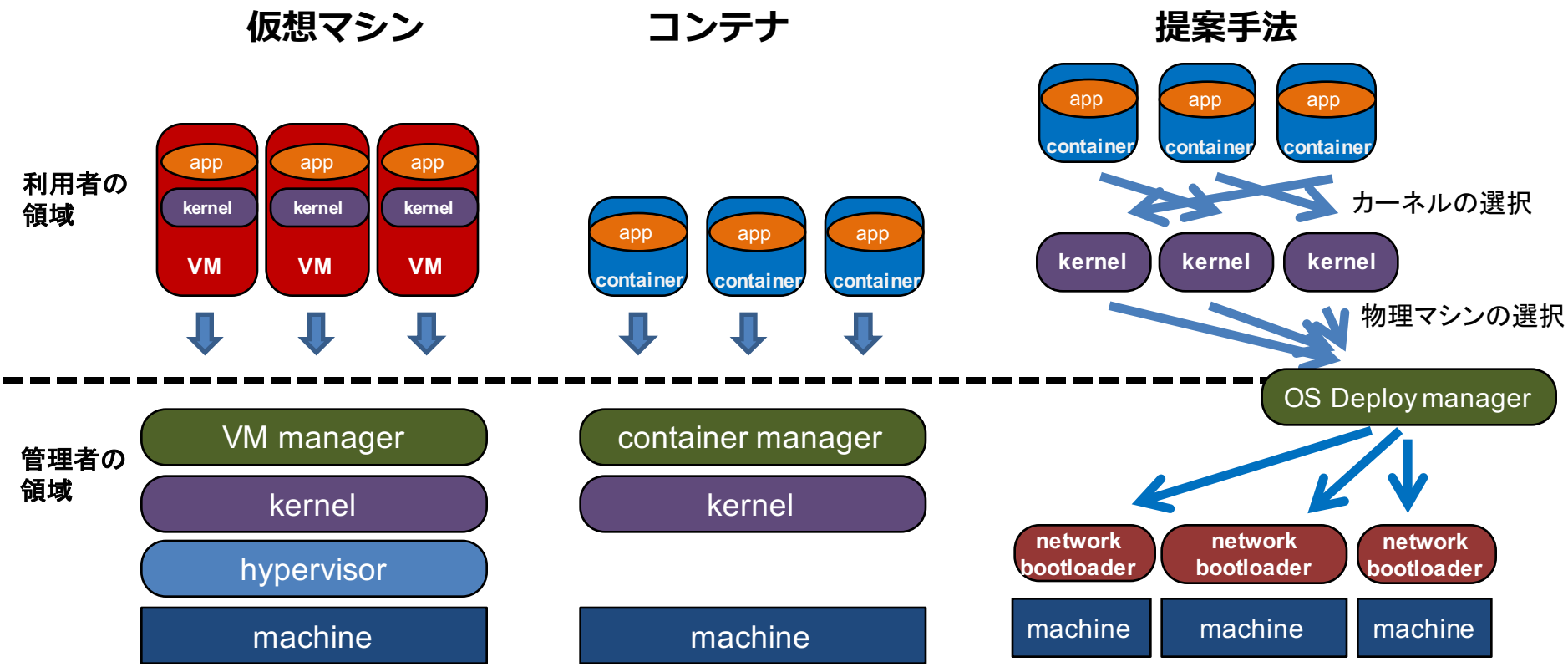
アプリケーション中心のシステム

- 現在のクラウド環境は、個々のアプリに対して最適な構成になっているわけではない。
- カーネル最適化の例
  - CPU: 数値計算ではHyper Threadingの無効化が一般的
  - Memory: HadoopではTransparent Huge Pages (THP)の無効化を推奨*
  - Profile-Guided Optimization (PGO):アプリ実行履歴からカーネルを再コンパイルしてアプリ毎に最適化

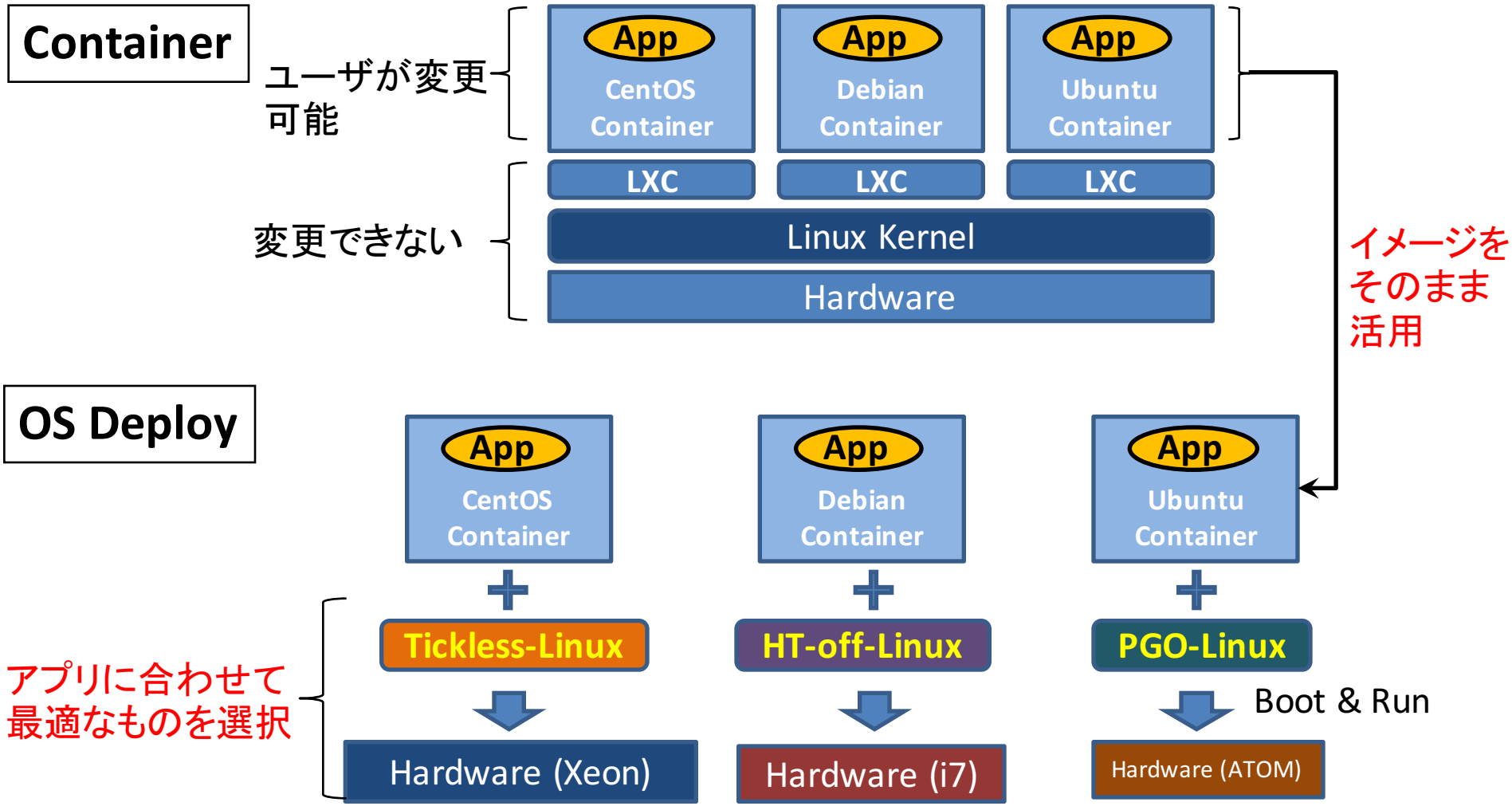
* http://structureddata.org/2012/06/18/linux-6-transparent-huge-pages-and-hadoop-workloads/

OSデプロイ機構の提案

- アプリケーションに対して最適なカーネル・ハードウェアを選択・実行する。

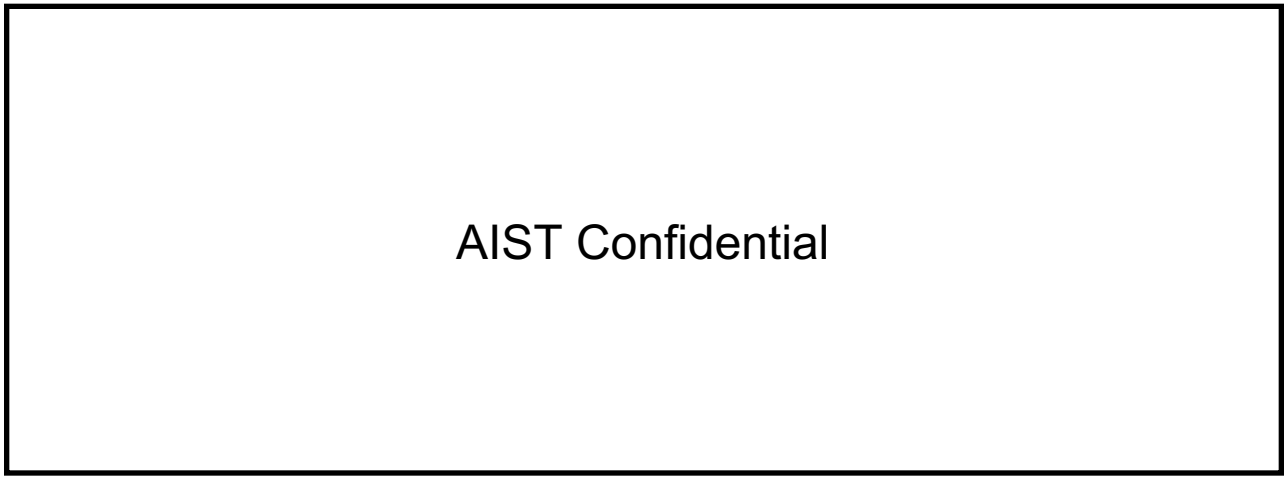


OSデプロイ機構の実装



ユースケース（KPGO）

- Kernel Profile-Guided Optimization
  - Apacheで20%、Redisで10%高速化
  - 実行時間が十分長ければ、ブートオーバヘッドは実効時間・消費電力共に回収できる



今後の展開

- アプリに特化したOSカーネルの研究
  - HPCの軽量OS、ライブラリOS
- システムソフトのオートクチューン化
  - アプリに合わせたシステムソフトが自動的に作成される仕組みの構築
  - 最適化オプションの機械学習
    - 最近のLinuxはカーネルオプションが豊富で、職人芸となっている最適化を自動化する。
    - Kernel Profile-Guided Optimization (PGO)

発表の流れ

- フローセントリックコンピューティング～ポストムーア時代に向けたデータセンターアーキテクチャ～
- 要素技術開発
  - アプリケーション中心型OSデプロイメントシステム
  - ハイパーバイザ型仮想化を用いたハイブリッドメモリシステム
- 実用化に向けた今後の取り組み

STT-MRAMとDRAMの比較

International Technology Roadmap for Semiconductors (ITRS) 2013 Edition, Table ERD3. Current Baseline and Prototypical Memory Technologies

		2013	2026
Read Time	DRAM	<10 (ns)	<10
	STT-MRAM	35	<10
W/E Time	DRAM	<10	<10
	STT-MRAM	35	<1
Single Cell Write Energy	DRAM	4E-15 (J/bit)	2E-15
	STT-MRAM	2.5E-12	1.5E-13

STT-MRAMはDRAMとほぼ遜色ない読み書き速度を提供しうるものの、
書き込み消費電力がDRAMよりも大きいというデメリットが存在する。

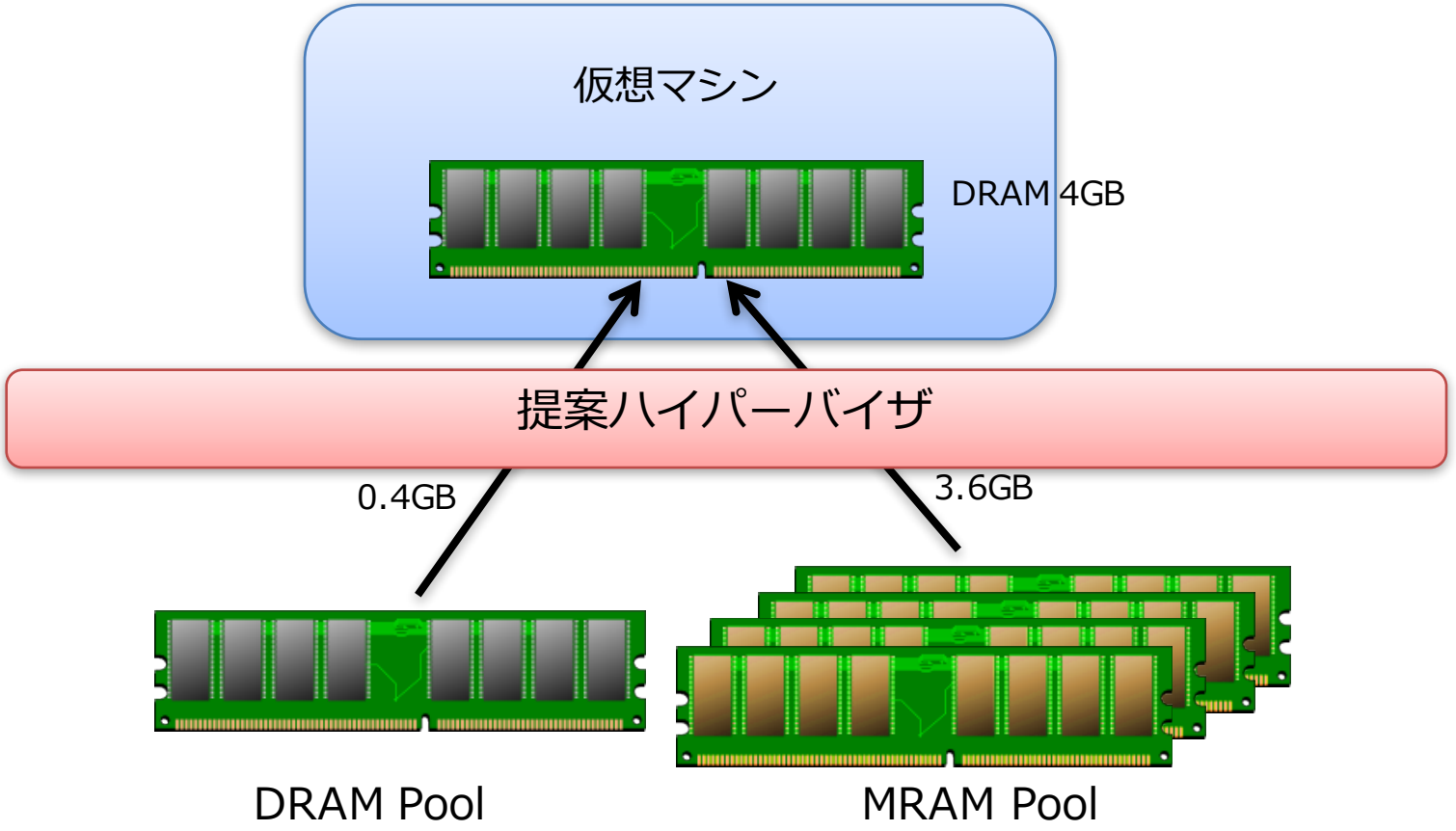
そこで…
　計算機システムのメインメモリとしてDRAMとMRAM両方を併用
　両者を上手に使い分けて省エネで大容量メモリを実現

本研究の目的

- クラウド環境を想定して、ハイブリッドメモリシステム向けのハイパーバイザを開発
  - 書き込みが頻出するメモリページをDRAM上に自動的に再配置して消費電力を削減
  - ハイパーバイザ上でメモリの使い分けを実現し、ゲストOSに対する変更は一切不要

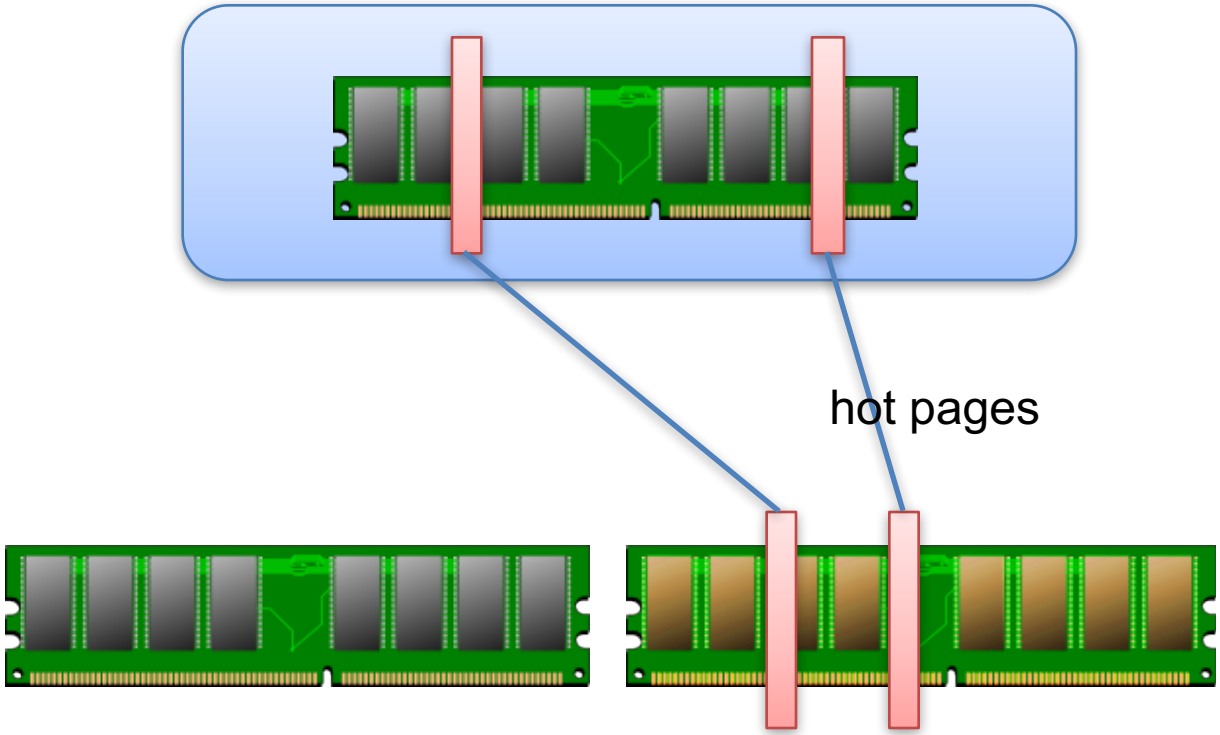
提案ハイパーバイザの概要（１）

- 仮想マシン（VM）のRAMをDRAMおよびMRAMから割り当てる
- ゲストOSにはDRAM単体で構成されているように見せる



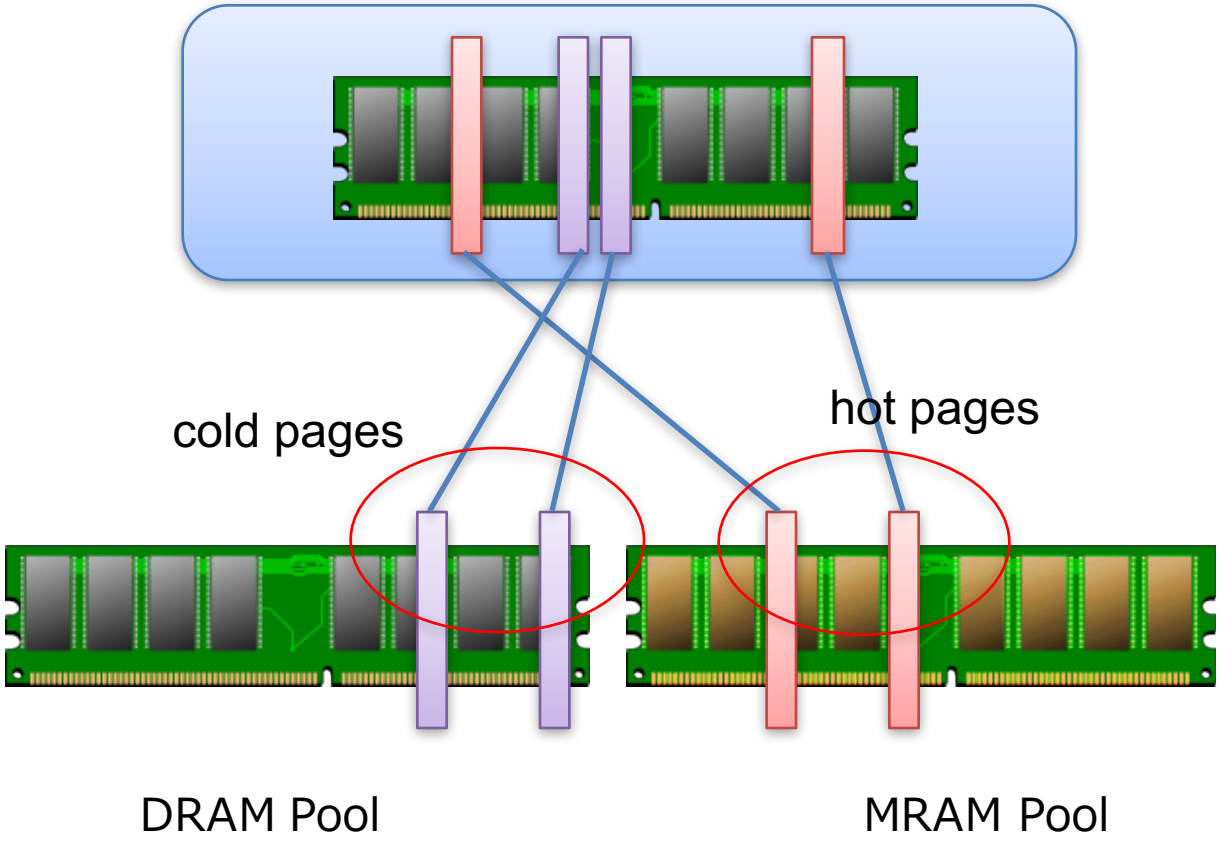
提案ハイパーバイザの概要（２）

1. MRAM上に存在するにもかかわらず、
   書き込みが頻出するゲスト物理ページを把握（消費電力増加の原因）



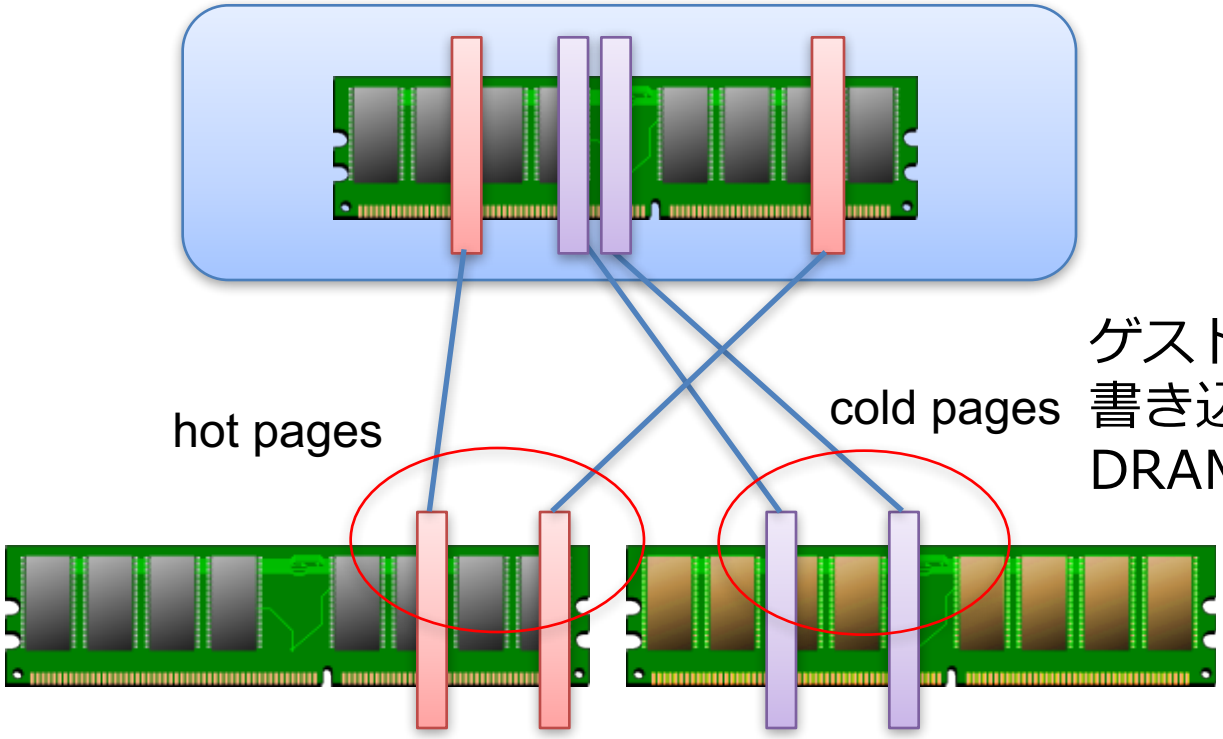
提案ハイパーバイザの概要（３）

2. DRAM上に存在するにもかかわらず
   書き込みがほとんど発生しないゲスト物理ページも把握



提案ハイパーバイザの概要（４）

3. MRAM上にあるのに書き込みが頻出するページと
   DRAM上にあるのに書き込みがほとんど起きないページを交換



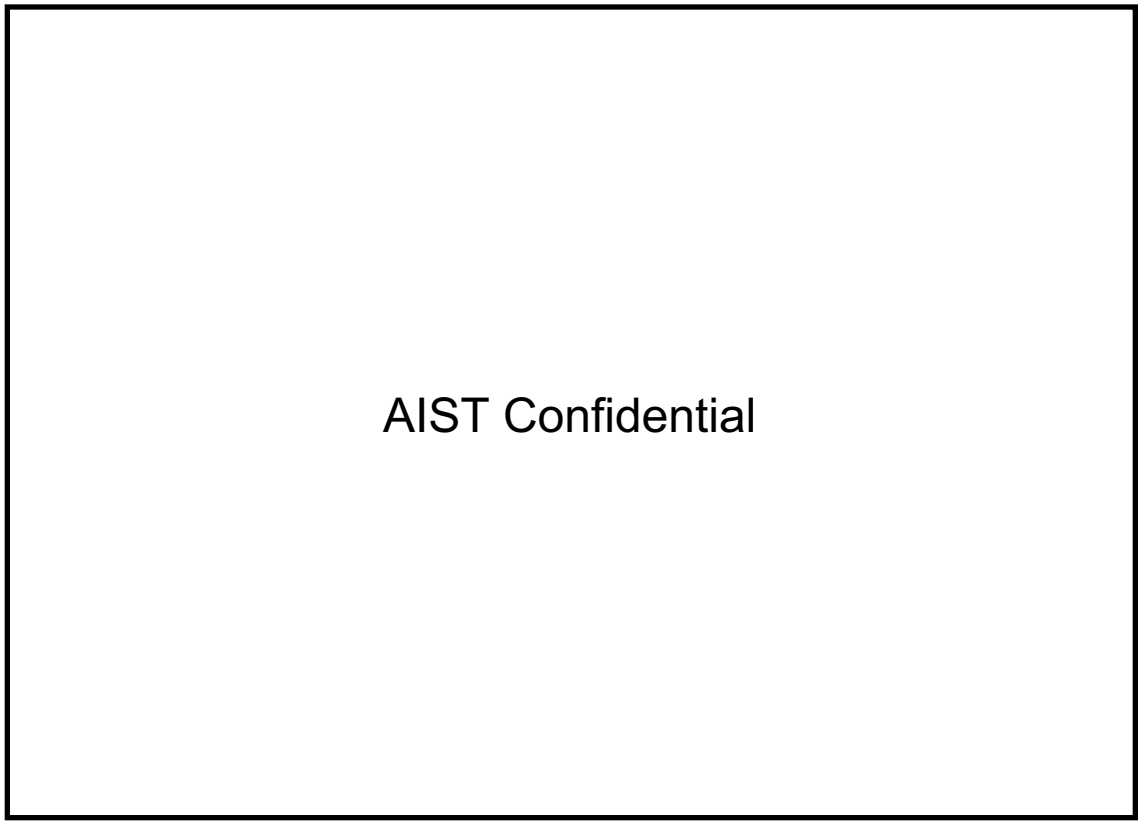
ゲストOSに気づかれないまま、
書き込みが多いメモリページを
DRAM側へ幅寄せ！

動的なページ再配置の結果

- DRAM容量が1%程度と仮定（DRAM 40MB, MRAM 3960MB）
- ゲストOS上のワークロードとしてLinuxのカーネルコンパイルを実行
- 半分程度の書き込みトラフィックをDRAM側へ集中できた

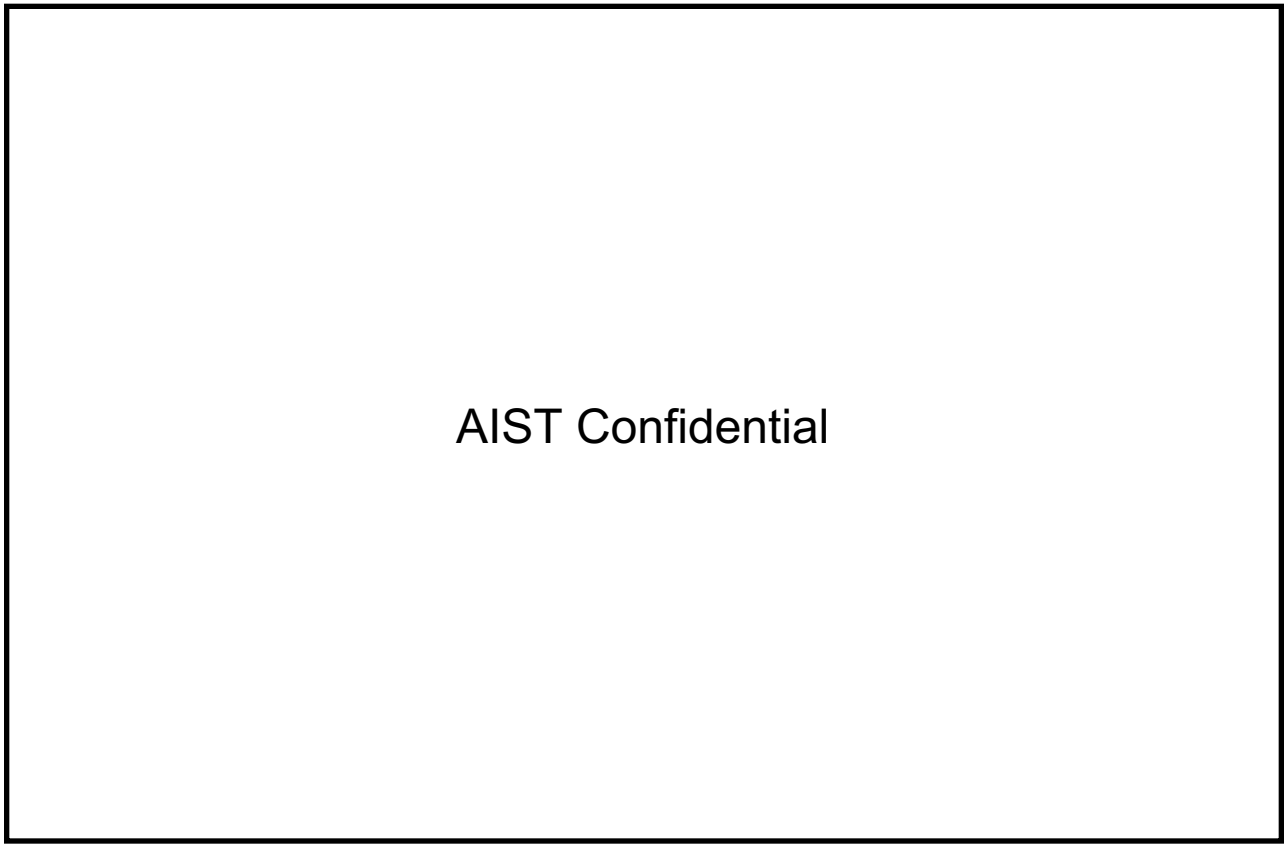
DRAMおよびMRAMへの
書き込みスループット

ページマイグレーション
実行数



省電力効果の見積

- Linuxのカーネルコンパイルを想定
- MRAMの書き込み電力はDRAMの100倍と仮定



今後の展開

- 現在は非常に単純なモデルを用いた消費電力の見積を採用

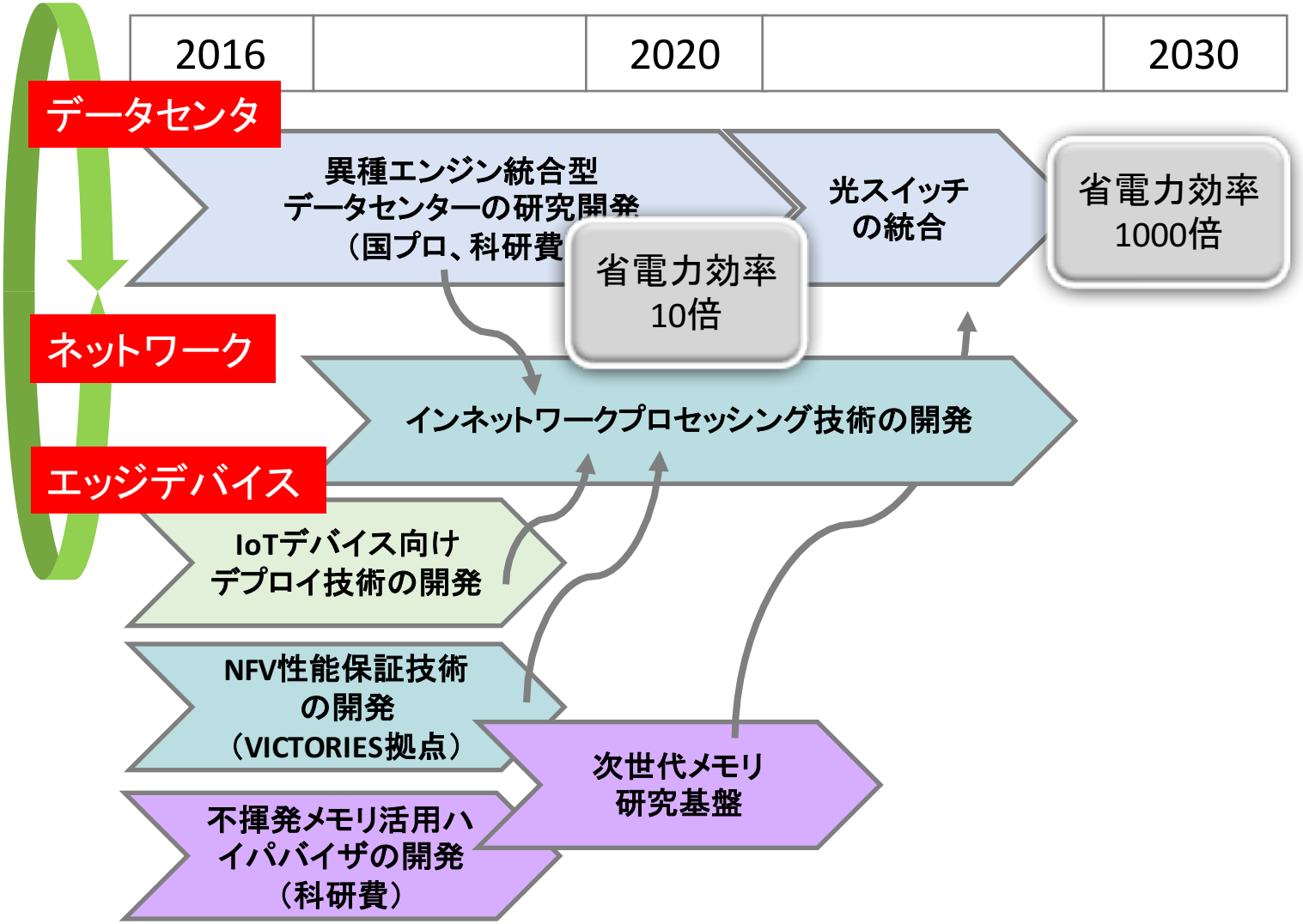
↓

- エミュレーション技術の確立（科研費）
  - 現状手に入らない将来のメモリデバイスをソフトウェアの力で仮想的かつ軽量に再現する技術
  - 将来のメモリデバイスを想定したメモリ管理手法を研究する上で不可欠

発表の流れ

- フローセントリックコンピューティング～ポストムーア時代に向けたデータセンターアーキテクチャ～
- 要素技術開発
  - アプリケーション中心型OSデプロイメントシステム
  - ハイパーバイザ型仮想化を用いたハイブリッドメモリシステム
- 実用化に向けた今後の取り組み

フローセントリックコンピューティング基盤の実現に向けたロードマップ



アウトカム

エクサビット級のデータを現在の1000倍のエネルギー効率で処理可能なフローセントリックコンピューティング基盤の実現

まとめ

- ポストムーア時代のデータセンターアーキテクチャであるフローセントリックコンピューティングを提案した
  - 用途に特化したエンジンの適材適所
  - データ移動のコスト軽減とその活用
- 処理内容や技術の成熟度に合わせて、エッジ、ネットワーク、クラウドを使い分けることが重要になる

将来の大規模データ処理基盤のあり方について一緒に議論する仲間を募集中！

「アーキテクチャ」主要研究者

- H27FYメンバー
  - 田中良夫（全体総括）
  - 池上努
  - 須崎有康
  - 高野了成
  - 竹房あつ子（現NII）
  - 田中哲
  - Jason Haga
  - 広渕崇宏
- 元メンバー
  - 工藤知宏（現東大）
  - 小川宏高
  - 谷村勇輔
  - 中田秀基