

2010.07.12 T2Kシンポジウム

パネルディスカッション 5年後のスーパーコンピュータ像

“データセントリック” スーパーコンピュータ

NEC システムプラットフォーム研究所

菅 真樹

自己紹介

NEC システムプラットフォーム研究所

- 菅 真樹

入社以来, ストレージ関係の研究に従事

- 対災害(ストレージ)システム
- 情報検索システム
- 大規模データ処理システム

現在は, クラウド基盤向けの大量データの処理・分析システムとして, [データセントリックコンピューティング](#)というコンセプトを立ち上げ, 研究を開始しています.

- 菅, 小林, 鳥居, 小川, 板橋, 宮田, 山川, 長谷部: “スケーラビリティと高効率性を備えたクラウド基盤を実現するデータセントリック分散制御”, DEIMForum 2010: 第2回データ工学と情報マネジメントに関するフォーラム, pp. C2-2 (2010).
- 大野, 小林, 菅: “データインテンシブコンピューティングの省電力化に向けたGPUノードの活用”, SWoPP2010金沢(CPSY)発表予定

データセントリックコンピューティング

コンセプト:

“**計算**”中心のアーキテクチャ思考から“**データ**”中心のアーキテクチャ思考へ

近年の情報量増大の流れから、コンピュータシステムの性能ボトルネックの要因は、何らかのデータI/Oボトルネックが多くなるのではないかと？

- ディスクI/O
- ノード間インターコネクトによるデータ転送
- 記憶媒体間のレイテンシ差

これらのボトルネックを回避するために、ボトルネックを生み出さないための様々な技術革新が行われてきた。

- インターコネクト帯域増加, トポロジの工夫, 分散ストレージ...etc.

データ中心で考えてみると、データの配置戦略・データの構造制御を従来よりインテリジェンスに実現できれば、ボトルネックの幾つかを回避できるのではないかと？

データ配置戦略・構造制御をインテリジェンスに行うには？

データがアプリケーションの情報をより知っていなければならない

簡単な例：

- 適切な並列化データI/O性能を得るためには、データが適切に分散配置されている必要がある
 - 利用特性に応じたデータ配置戦略
- 必要データにインデックスが貼ってあれば、必要データに短時間でアクセスできる
 - 利用特性に応じたデータ構造制御戦略

データを適切に配置，データ構造を適切な形にするために制御していくためには、**ストレージがデータに対応するアプリケーションの情報をより知る(or推定)**できる必要がある

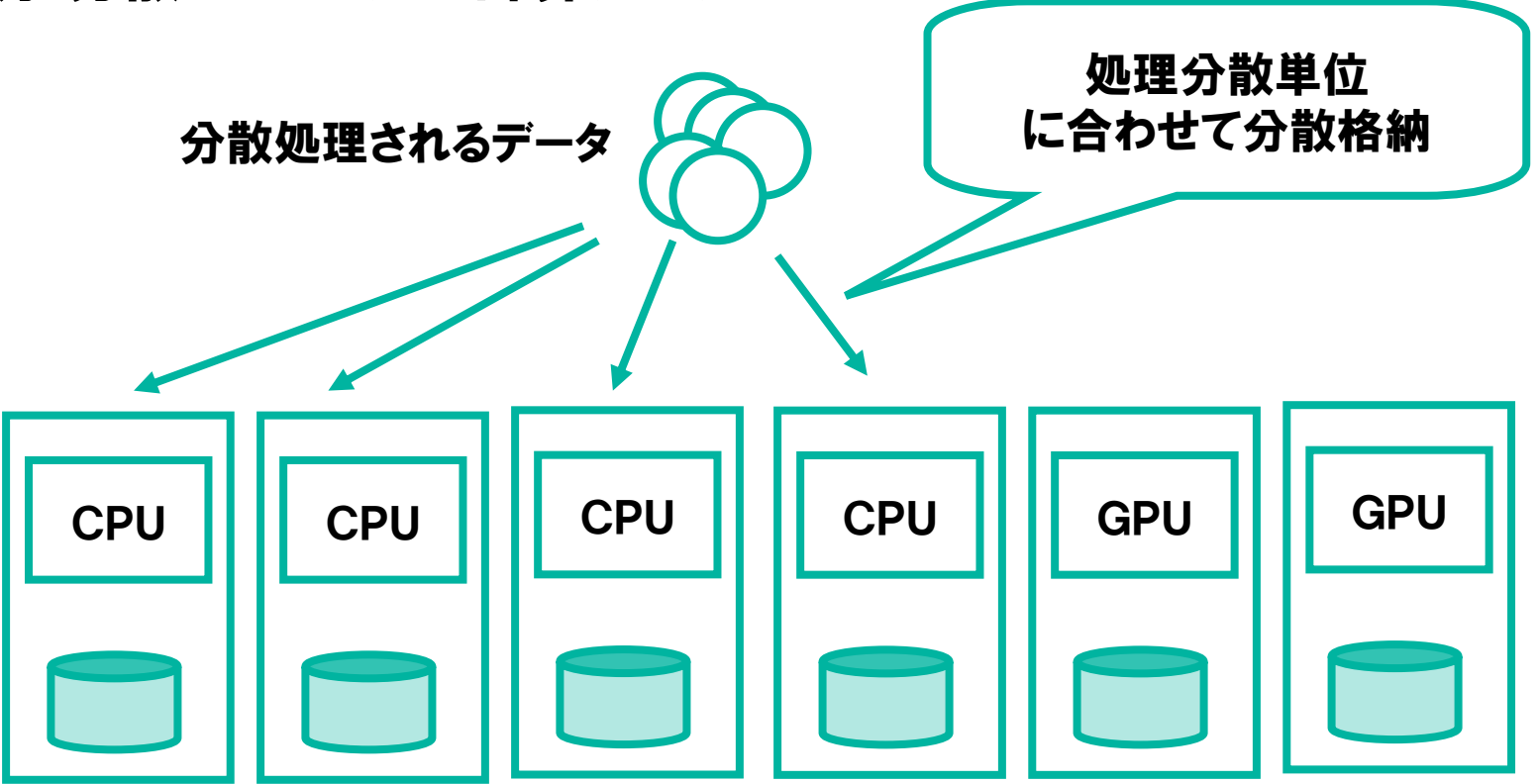
- データがいつ利用されるのか？
- どのようなアプリケーションに利用されるのか？
- そのアクセス特性や必要な性能は何か？

本来，コンピュータプログラムにおいて，データ構造は処理内容と不可分であったが，管理や使い勝手の問題から，ストレージやデータベースは独立してきた。

- そのため，ストレージは独自に最適化の道を辿ってきた
- ストレージはアプリケーションを知らず，アプリケーションはストレージを知らなくなってきたのではないか？

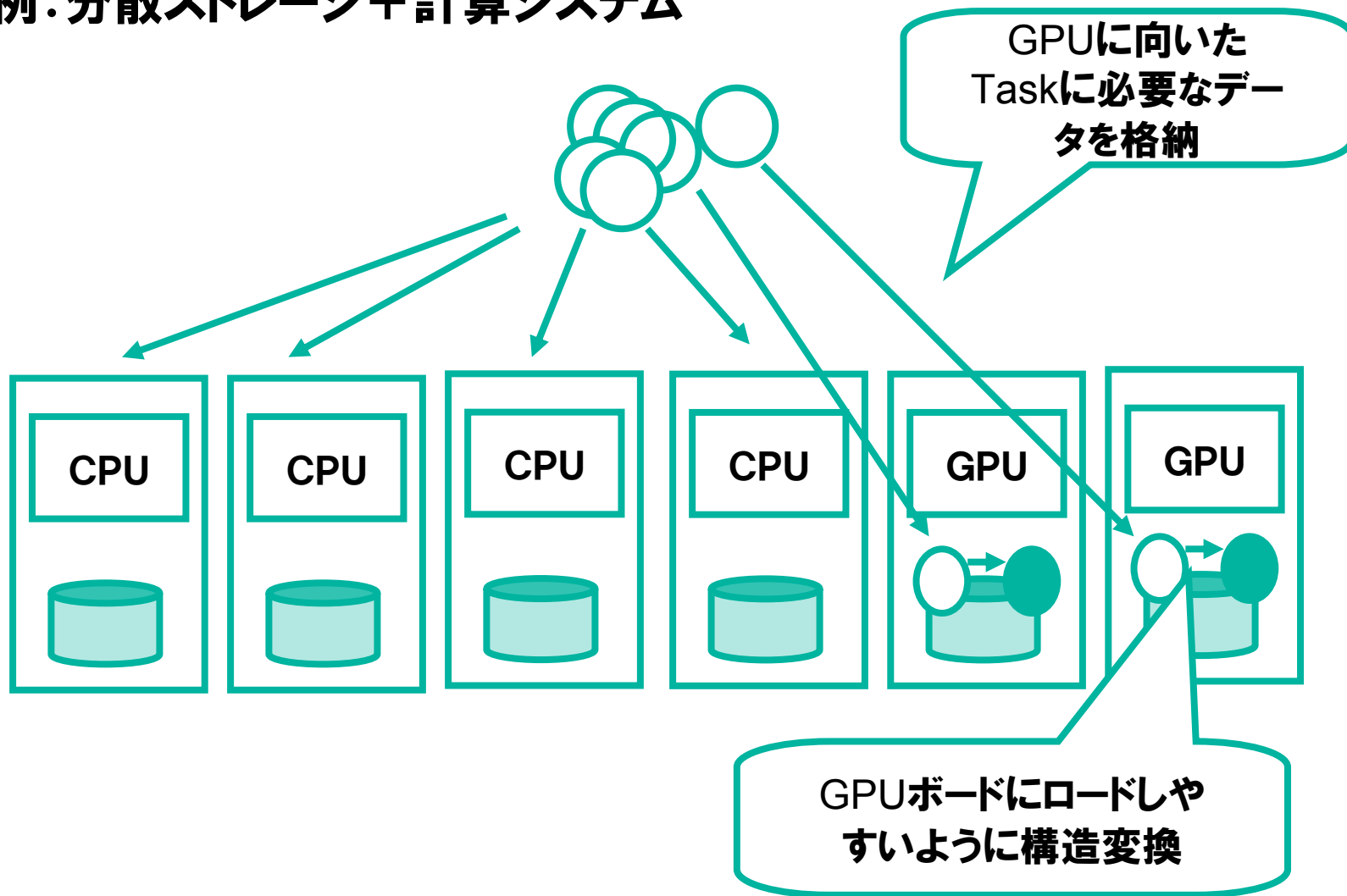
データ(ストレージ)が、よりアプリケーションの情報を知っていれば(1):

例:分散ストレージ+計算システム



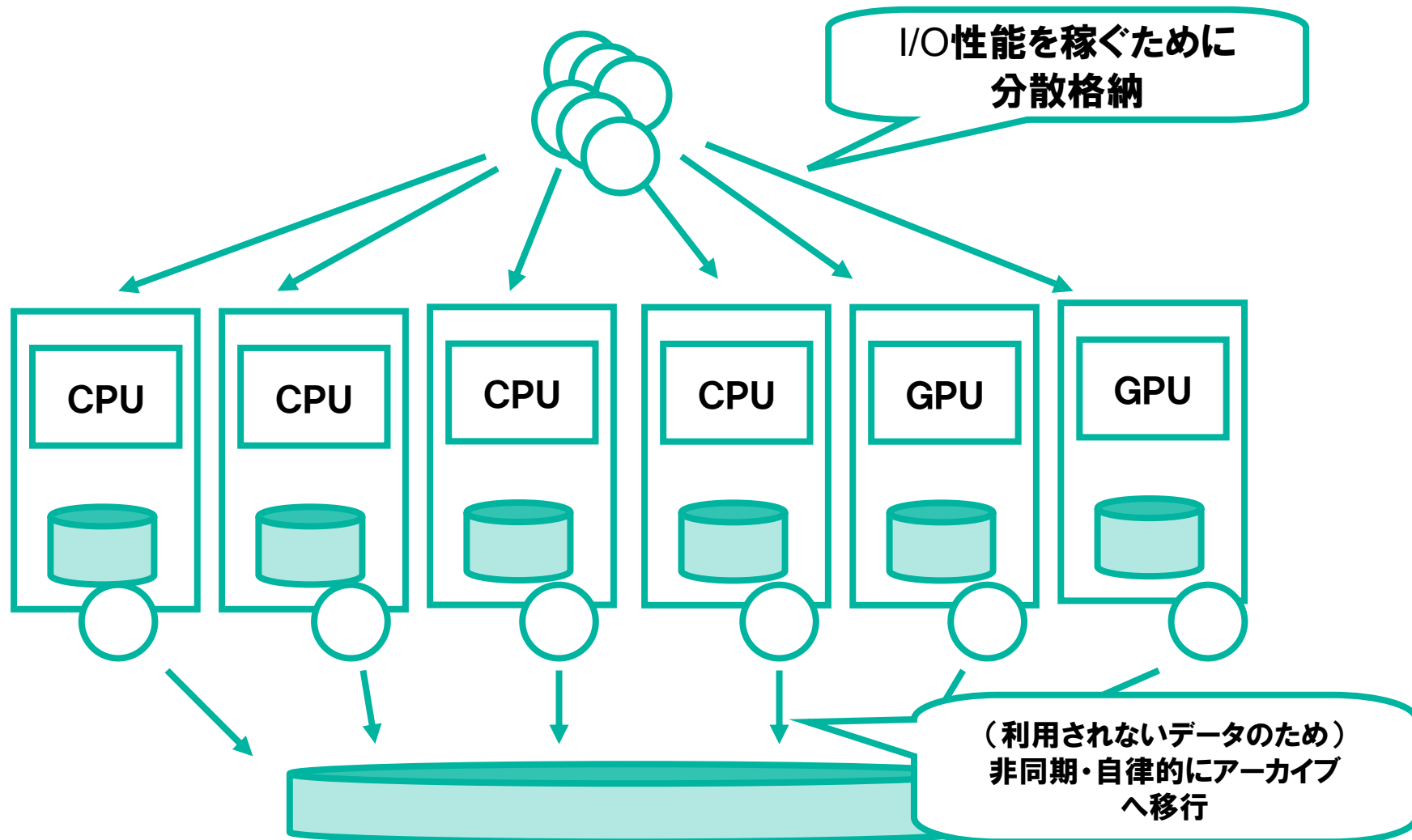
データ(ストレージ)が、よりアプリケーションの情報を知っていれば(2) :

例:分散ストレージ+計算システム



データ(ストレージ)が、よりアプリケーションの情報を知っていれば(3) :

例:分散ストレージ+計算システム



データセントリックコンピューティング:その他のコンセプト

ヘテロジニアスリソース(GPGPUなど)を有効に活用する

- GPUのハードウェアの特殊性
 - ・ バス間メモリ転送ネックにより, 処理によって性能に差がある
 - ・ プログラミングの特殊性:技術も必要

省電力性

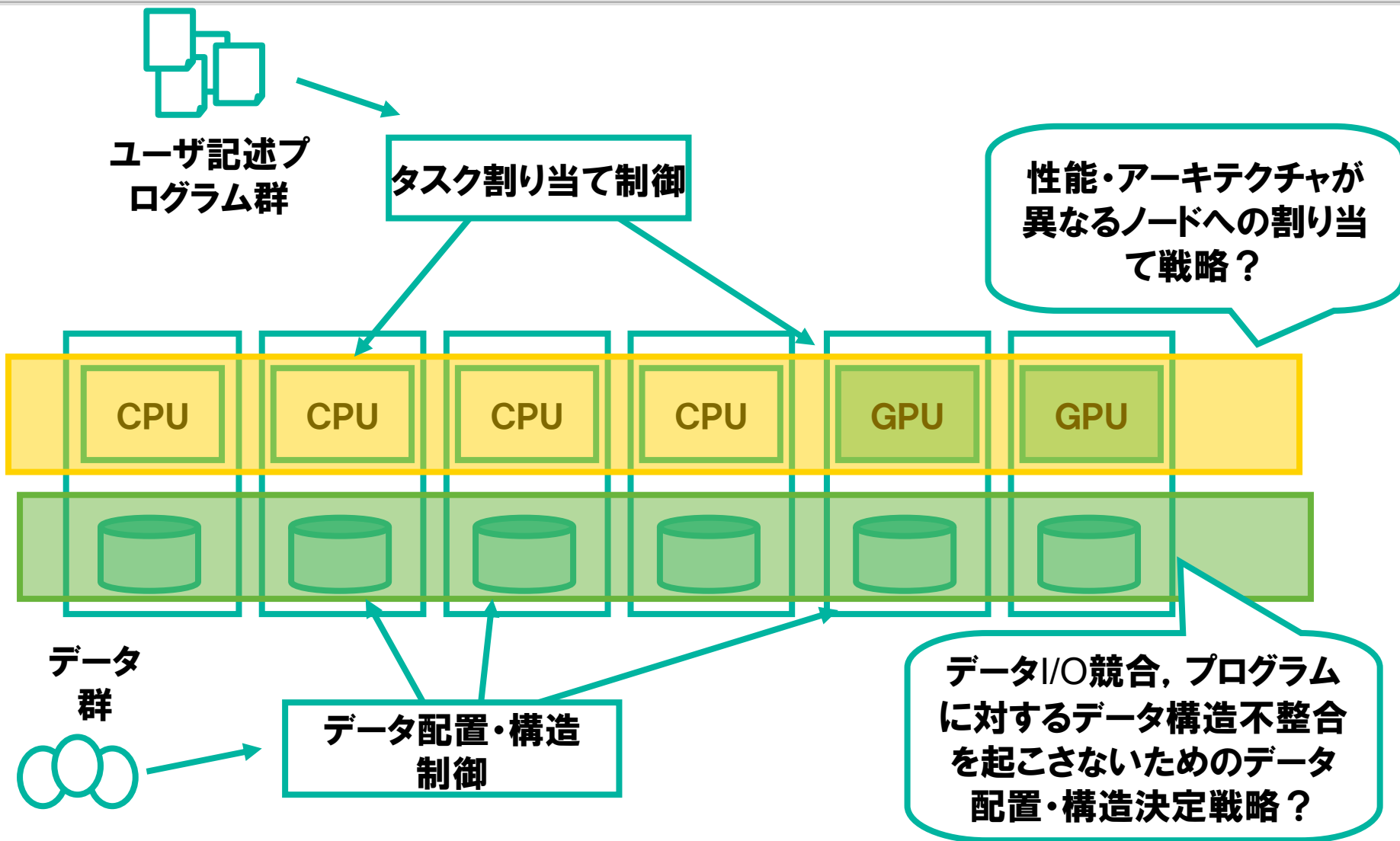
- データ配置や構造の制御, GPGPUの有効活用による性能向上による間接的寄与
- 不要ノードの停止可能時間を創出するデータ配置:
 - ・ ある時間帯にアクセスされないデータを集約してノードを停止
- 性能と電力消費量の間トレードオフ:
 - ・ 性能より電力消費量を重視するアプリケーションは電力消費量を削減するように制御

ストレージスケーラビリティ

- ストレージレイヤを分散KVSで実現

現在想定している研究課題

例:分散ストレージノードが計算ノードも兼ねる形のシステム,CPU/GPU混載クラスター



5年後のスーパーコンピュータを作るとしたら・・・

もっとデータ(ストレージ)に親和性の高いスーパーコンピュータ

データセントリックコンピューティングコンセプトを取り入れつつ、もっとデータI/Oに気を遣ったスーパーコンピュータ

- データの配置・格納方式の制御
- ヘテロジニアスリソースを有効活用

ストレージ

- データI/Oに対する利便性向上は、様々な価値を提供できます

省電力化

- 消費電力の観点での制御

ユーザに対する利便性向上

- タスク割り当て、データ配置・構造制御により、より多くのユーザに使いやすい形に

スーパーコンピュータにおけるデータとストレージ

データ管理, データI/Oは永遠の課題

- データ処理, プログラム実行時のデータI/Oボトルネック
- 研究データの管理

膨大なデータI/Oスループット性能

- CPUコアの増大→CPUコアへのデータ投入
 - システム全体で100万コア
 - コアあたり100MB/secのデータを転送するとなると100TB/secのスループットが必要
- チェックポイントなどの書き出し
 - 高いスループット性能を持つストレージは, プログラム実行の対故障性にも貢献

ストレージレイヤでの演算

- Diskの近くに付与される計算ユニットで一部の計算を肩代わり
 - FPGAなどを利用(ex. Netezza)
 - フィルタ系処理のみを実行(ex. Oracle Exadata)

利便性

- データI/O性能を必要な時に必要なだけ. ステージング不要.

ストレージとの結合度を高めたスーパーコンピュータ

スーパーコンピュータにおける省電力化

電力コストの増大

- 電力コストだけで年間1x億円(Jaguar),
- 次世代スーパーコンピュータ年間xx億円？

HWの進化による省電力化は求められ、進むだろう

システム研究者として出来る省電力化アプローチ

- 遊休ノードの創出によるノード停止
 - データ配置
 - タスクスケジューリング
- ストレージノードの停止
 - スループット性能が必要な際にはActiveにしつつ、性能不要時にノード停止可能時間を創出する

データの特性を把握した制御により省電力効果上昇を目指す

ユーザに対する利便性向上

スパコンのユーザ層
は今のままでよいの
か？

大量データ処理におけるトレンド

- RDBMSの複雑化, スケールしない, SQLが使いにくい
- MapReduceのような, Simple Program Multiple DataStream (SPMD) アルゴリズムでロジックを記述する形が流行
 - 並列化実装をユーザに対して隠蔽

スーパーコンピュータにおける, ユーザに対する利便性を追求する方向とは？

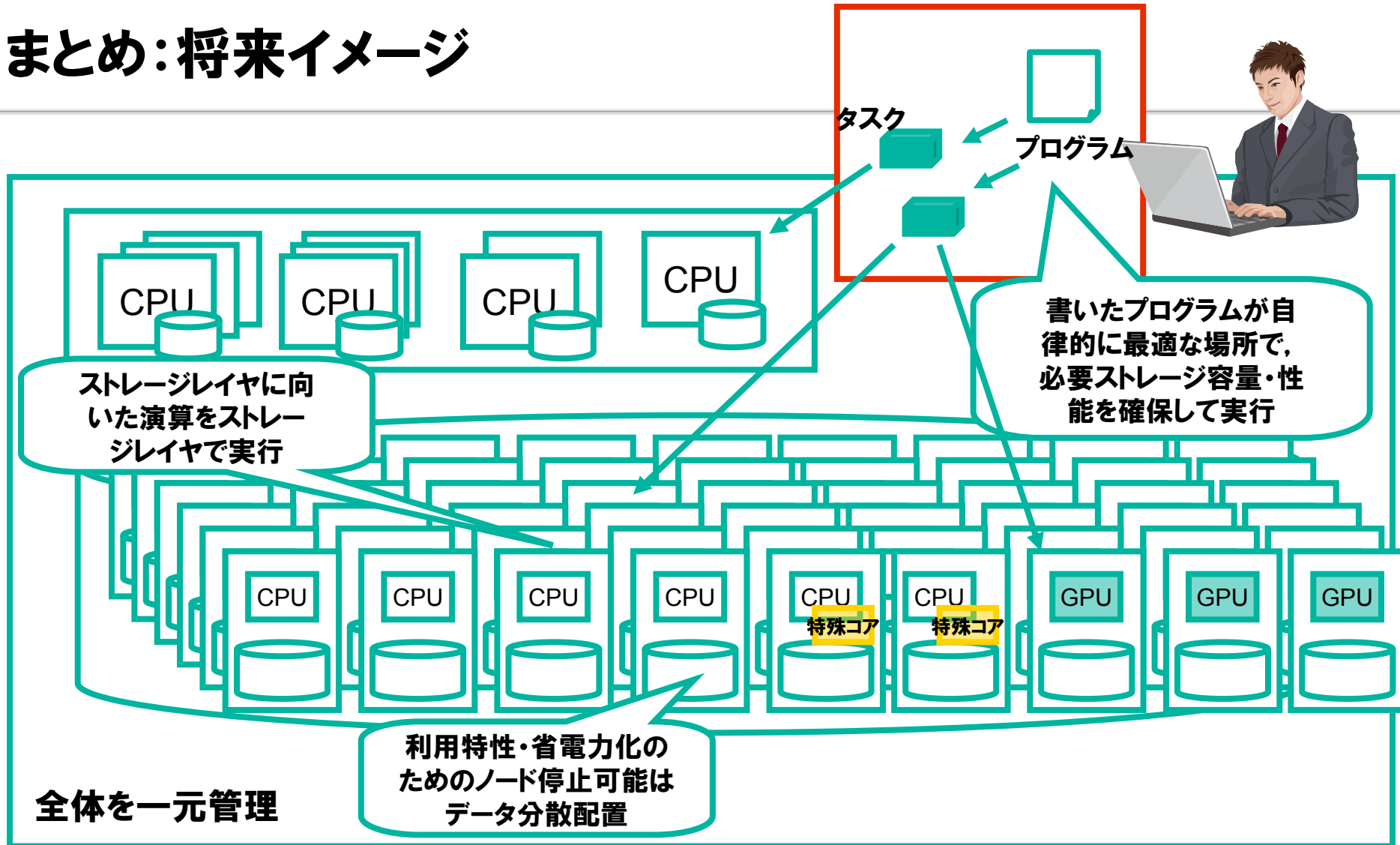
- ユーザ/アプリケーションの2極化
 - チューニングを重ね, ベストパフォーマンスを追求するタイプのユーザ/アプリケーション
 - 比較的簡単にプログラムを記述し, ほどほど満足できる速度を要求するユーザ/アプリケーション

ユーザのプログラムから, 部分的なタスクを適切なリソースに割り当て:

- Ex.)例えば, GPUが得意な部分タスクを優先的にGPUに割り当て
- プログラムからタスクの分割をどうやるか？
 - コンパイラ
 - 記述フレームワーク(Sawzall)
- データはプログラムによる利用内容に応じて, データ構造を変化させていく

アプリケーション・データ特性の把握により
ユーザに対する利便性向上を目指す

まとめ: 将来イメージ



ストレージをよりインテリジェンスにすることにより、より利便性の高いスーパーコンピュータを目指したい