

ストレージ・ソフトウェア・ボトルネックを克服する手法

手法

長所

短所

問題に対してさらなるハードウェアを投入する

ソフトウェア・ボトルネックを隠すためにCPUやメモリをシステムに追加投入する

特定のストレージ機能を個別のコアに割り当てることでパフォーマンスを改善する
メモリはCPU-フラッシュ間のキャッシュとして働く

ハードウェアのコストが増大する:
CPUマルチスレディングを有効に
使えない: 本来揮発性であるメモリが
絡むと高価になり、複雑さが増す

重要視するものを変える

パフォーマンスの代わりに
機能にフォーカスする

ストレージ・システムが提供する
IOPSは大部分のデータセンターに
とって十分な性能:
管理を簡素化しQoSを改善できる

必要以上の容量を買わされる:
AI、機械学習、ディープラーニングなどの
ワークロードは、一般的なアレイが
提供する以上のパフォーマンスを
要求する

ストレージ・システム概念を捨てる ためにクラスター環境で ワークロードを稼働する

サーバークラスターを管理する
ソフトウェアは、基本的なストレージ
管理機能を持っている

ネットワークや
先進的ストレージソフトウェアは
レイテンシを増やさない

ストレージ・リソース使用率が
非効率的で且つ一般的に低い:
ストレージ管理はクラスター毎に
別々になる: 容量の消費が
大幅に増加する

コンポーザブル・クロスクラスター ストレージを使う

ユーザーが必要に応じて引き出せる
共有の仮想ストレージ
リソースプールを作る

ドライブが特定のサーバー用として
縛られていない:
必要に応じてドライブやサーバーに
割り当てたユースケースを
変更できる

ストレージ管理はクラスター毎に
別々に実行される:
機能とパフォーマンスは大部分
クラスター・ソフトウェアの
ストレージ機能に依存:
先進的ネットワーク
アーキテクチャーが必要

カスタマイズ可能な ハードウェアを使う

演算装置はストレージ・システム
専用に設計される

カスタムFPGA、ASIC、
ネットワーク・インターフェース・カード
これらは、タスクをオフロードして、
ストレージ・パフォーマンスの
高速化に貢献する

ストレージ・ソフトウェアの
オフロードによるメリットが不明確:
コストが高む難点:
Intelの開発サイクルを
活用できなくなる

ゼロからストレージ・ソフトウェアを 書き直す

ストレージ・ベンダーは、ストレージ
パフォーマンスを改善するために
アルゴリズムを再考する

ドライブからフルに
パフォーマンスを引き出し
ストレージ・ソフトウェアを
よりマルチスレッドに
対応させられる

時間がかかる作業:
ストレージ製品が市場に
出るのが遅くなる