

Platform グリッド協議会 第15回ワークショップ
- LSF進化形 -

2006年3月8日
プラットフォームコンピューティング株式会社
ビジネス開発担当
石川 克也

Platform 今日のお話の内容

- Platform LSF とは
- Platform LSF6.2 の特徴
- Platform Symphony とは
- Platform の今後の方向性

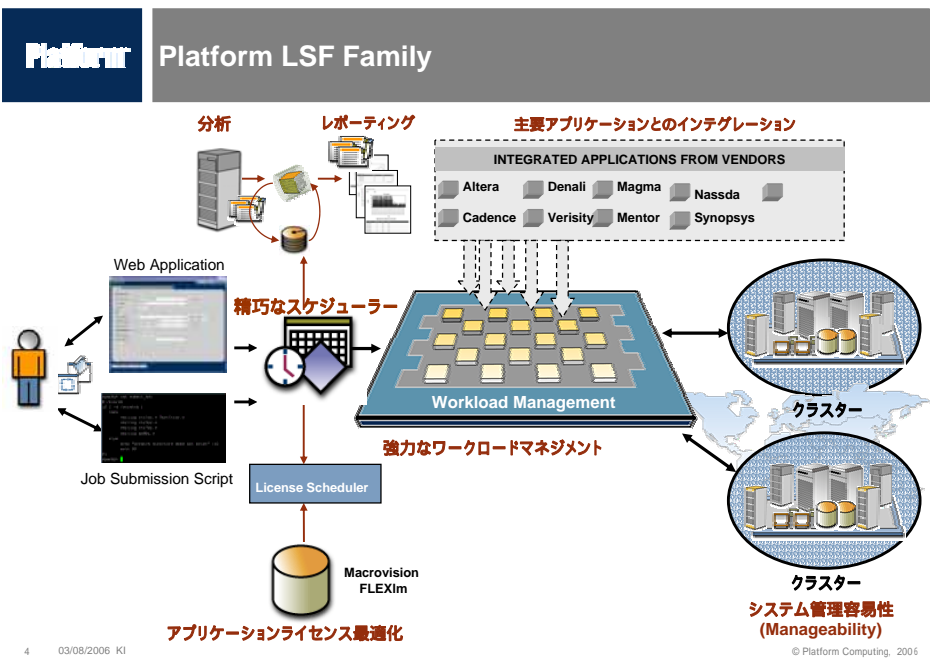
2 03/08/2006 KI © Platform Computing, 2006

Platform

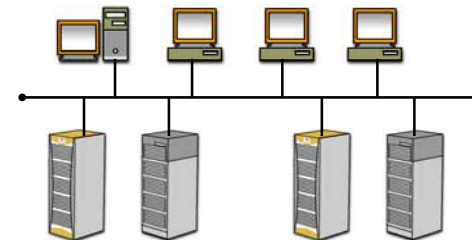
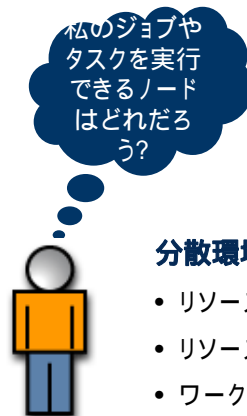
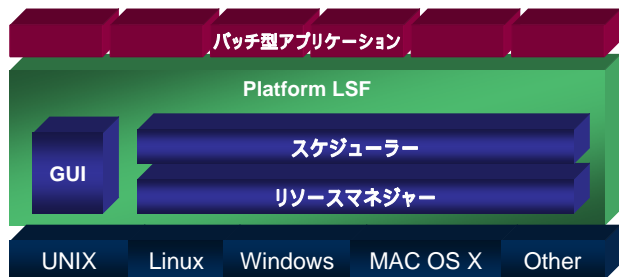
Platform LSFとは

Platform

3 03/08/2006 KI © Platform Computing, 2006



Platform LSF (Load Sharing Facility) とは、
バッチ型アプリケーションを対象にしたグリッドミドルウェア

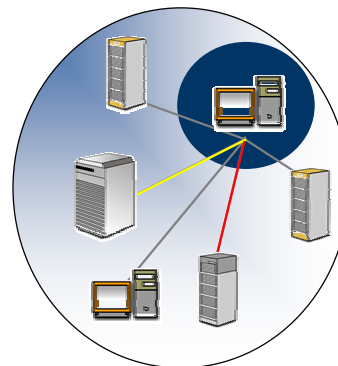


分散環境(数百のホスト)において

- リソース制御が集中管理されていない
- リソースモニタリングが集約されていない
- ワークロードが分散されていない
- リソース利用が不均衡
- ユーザはリソース不足に悩まされる



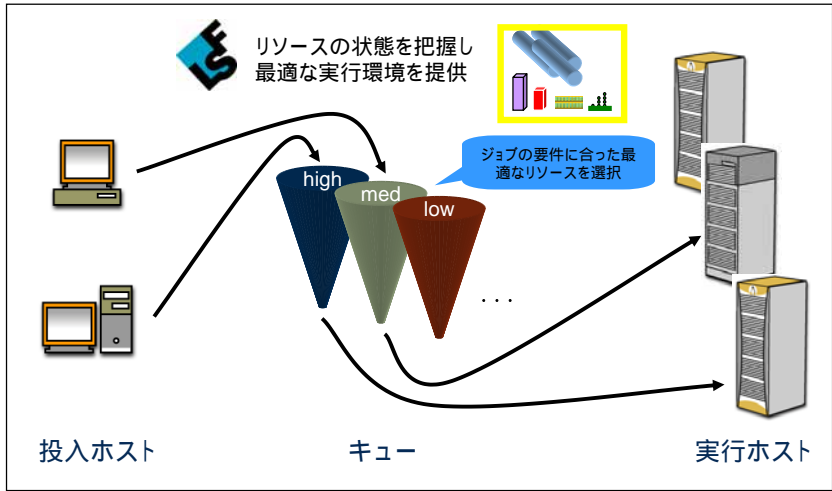
**Platform LSFにより
仮想化された計算機**



- 各ノードは、Platform LSFが管理する「クラスタ」にグループ化される
- クラスタ内のサーバホストのうち、1ホストが「マスタ」として選ばれる
- クラスタ内の各スレーブサーバホストは、自分の“稼動状況”を定期的に収集し、それをマスタに報告する

- ユーザがジョブやタスクをLSFに投入すると、マスタは、収集された稼動状況に基づいて適切な実行ホストであるサーバホストを決定する

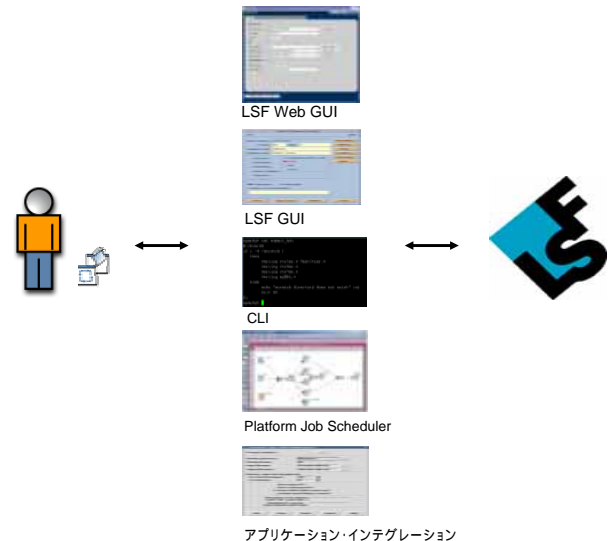
Platform キューによるスケジューリング



Platform スケジューリングポリシー

- FCFS
First Come First Serve
- PREEMPTION
優先度に応じてジョブの割り込み
- FAIRSHARE
現在の実行状態と過去のCPU使用実績をもとに、ユーザあるいはユーザグループ間で公平に配分
- BACKFILL
パラレルジョブ投入時、指定したCPU数が確保できるまでの待ち時間を他の短いジョブの実行に利用する
- EXCLUSIVE
ホストの独占使用

Platform ジョブの実行手段



Platform

Platform LSF 6.2の特徴

Platform

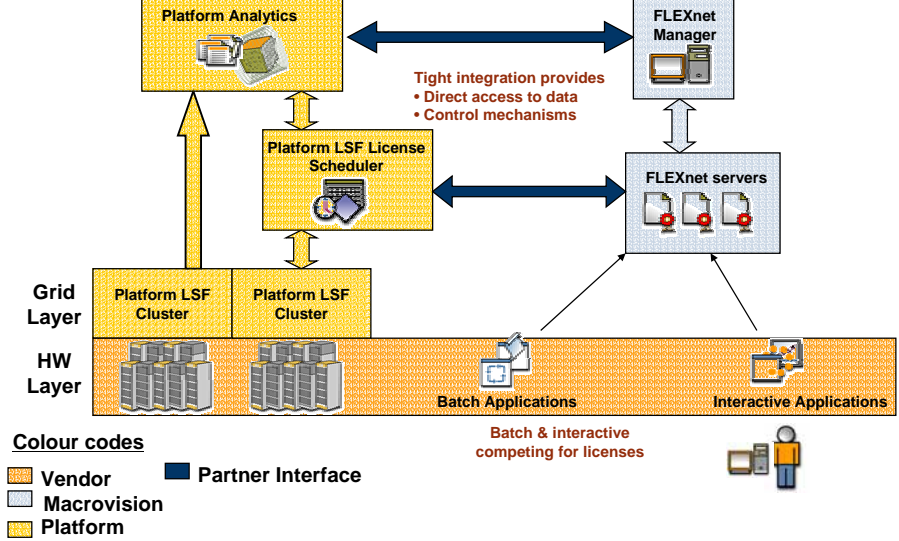
Platform What's New in Platform LSF 6.2

- Performance and Scaleability Enhancements
 - Project scheduling
 - Bulk job termination
 - Improved job event logging
 - Improved I/O performance in the job file info directory
 - Improved job forwarding for remote Platform LSF MultiCluster resources
 - Support for dual-core CPUs
- Platform LSF on Windows Enhancements
 - IBM Rational ClearCase Support
 - Silent installation
- Usability
 - Open advance reservation
 - Condensed host group configuration
 - Accounting and notification of Platform LSF product license overuse

Platform LSF ライセンススケジューラー 6.2



Platform LSF ライセンススケジューラー V6.2 アーキテクチャー



Platform ライセンス最適化の主要な価値

- リアルタイムでのポリシーベースのライセンス最適化
 - リアルタイムでスケーラブルなライセンス情報インターフェース
- Shares – 優先順位式スケジューリング
 - 各プロジェクトにライセンスを「公平に割り当て(フェアシェア)」る
- Limits – 乱用の管理
 - ライセンスの乱用が起こらないようにする(追跡または拒否)
 - 優先順位の低い作業によるライセンス使用コスト(使用料)の超過を防止する
- Ownership – 動的な優先順位変更
 - 優先順位の高いプロジェクトが、必要ときにライセンスにアクセスできるようにする
 - ライセンスを予備として「確保」する必要性をなくす(プリエンプティブな割り当てが必要)
- 包括的なレポート
 - ハードウェアリソースの消費と組み合わせたライセンス使用状況履歴
 - 企業全体にわたる視認性

Platform

Platform Symphonyとは

Platform

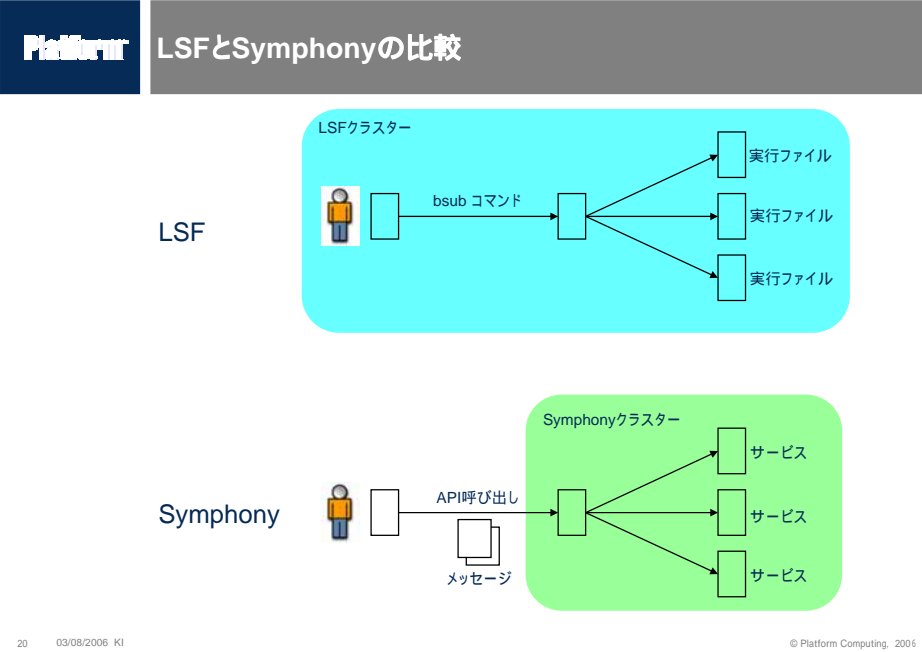
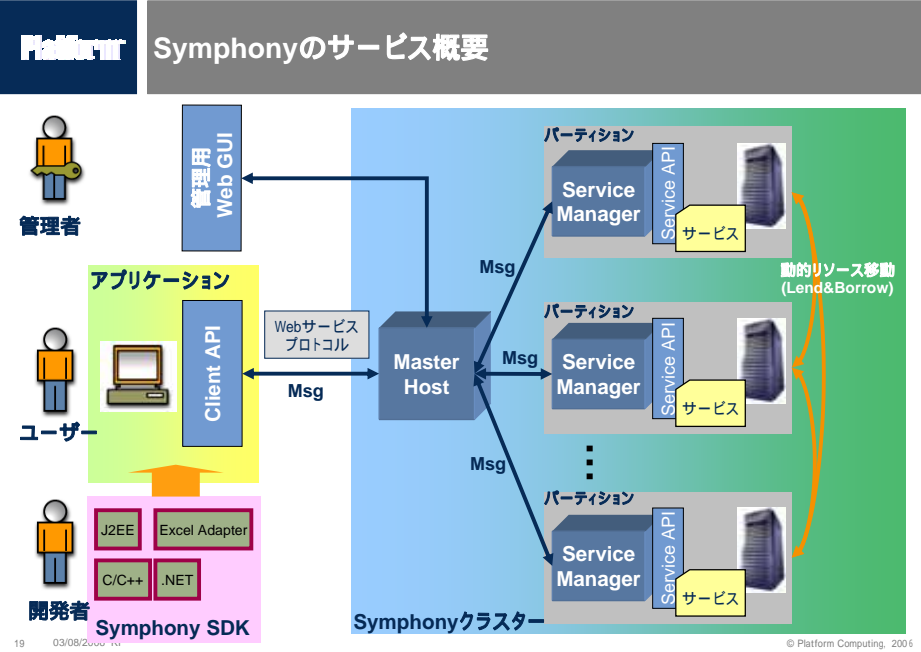
17 03/08/2006 KI © Platform Computing, 2006

Platform Service-Oriented Infrastructure
– Platform Symphony –

10年以上の実績のあるLSFをSOA拡張 – より広範囲のビジネスに

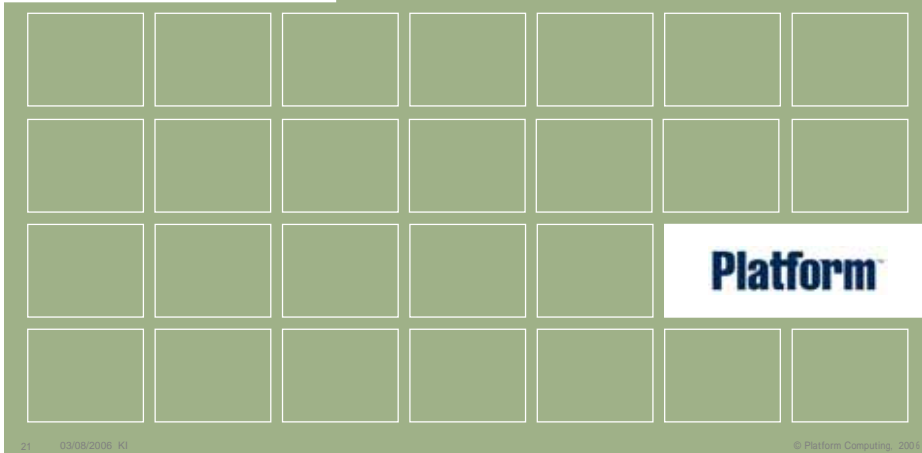
- サービスをグリッドで実行
- 組織を超えた動的リソース配置・移動 (パーティション間のLend&Borrow)
- 標準技術に基づく疎結合アーキテクチャ
- サービスレベルの確保・維持
 - 高速かつ高機能なワークロードマネジメント(スケジューリング)
 - リソースの仮想化
 - サービスの動的かつ最適なプロビジョニング
 - 自己修復性による高可用性
- 金融アプリケーションとの連携を中心に, ワールドワイドで導入中

18 03/08/2006 KI © Platform Computing, 2006

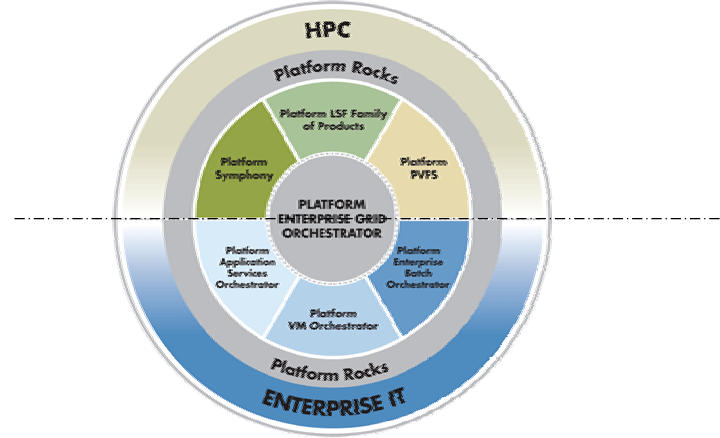


Platform

今後の方向性



Platform エンタープライズITに対するPlatformのグリッドパラダイム

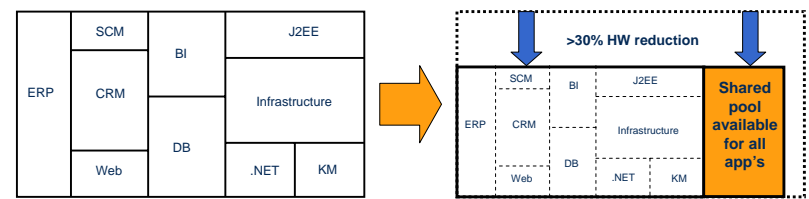


Enterprise Grid: 全てのエンタープライズアプリケーションが、単一の仮想計算機上で動作する環境

Platform エンタープライズグリッドの構想

サイロ化されたデータセンター エンタープライズグリッド

データセンターの“オーバーヘッド”の構成



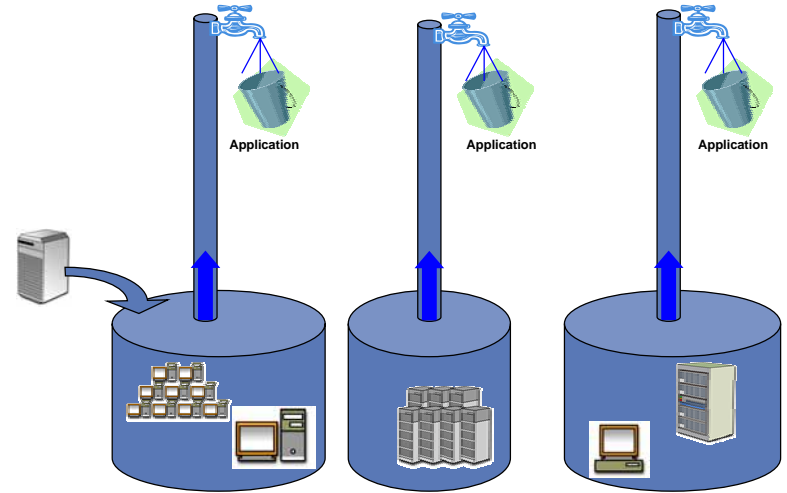
専用のサイロ

- 専用で高価なハードウェアのサイロ
- 個々のアプリケーションのピーク要求に合わせたプロビジョニング
- スケールアップ
- 限定されたインフラストラクチャ使用状況データ

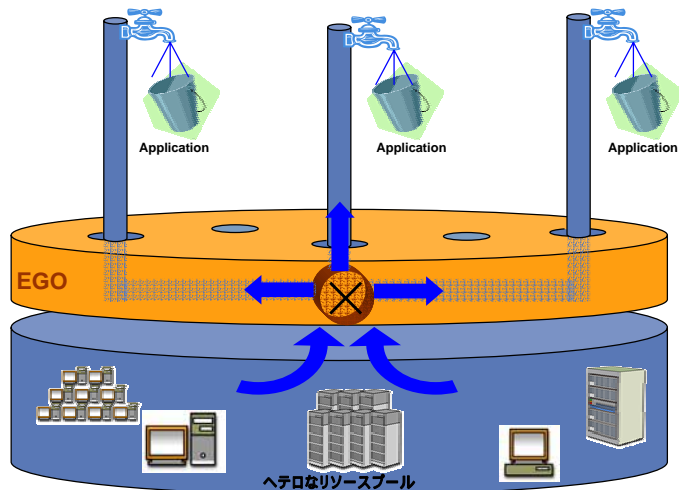
グリッド化された環境

- すべてのアプリケーションに対する共有リソース
- ビジネス優先度に基づいた、アプリケーションのリソース使用量の動的な調整
- 全体的なピーク要求、SLAに合わせたプロビジョニング
- コモディティインフラストラクチャを使ったスケールアウト
- ソフトウェアライセンス管理を含む、包括的なインフラストラクチャ使用状況データ

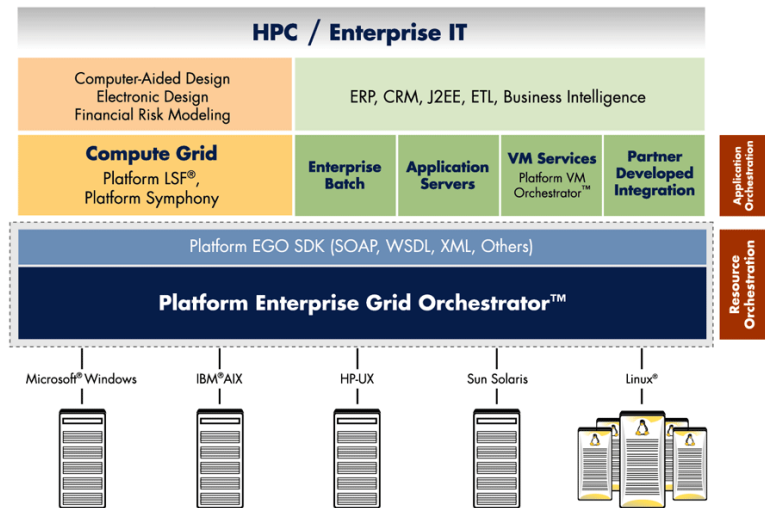
Platform サイロ型リソースマネジメント



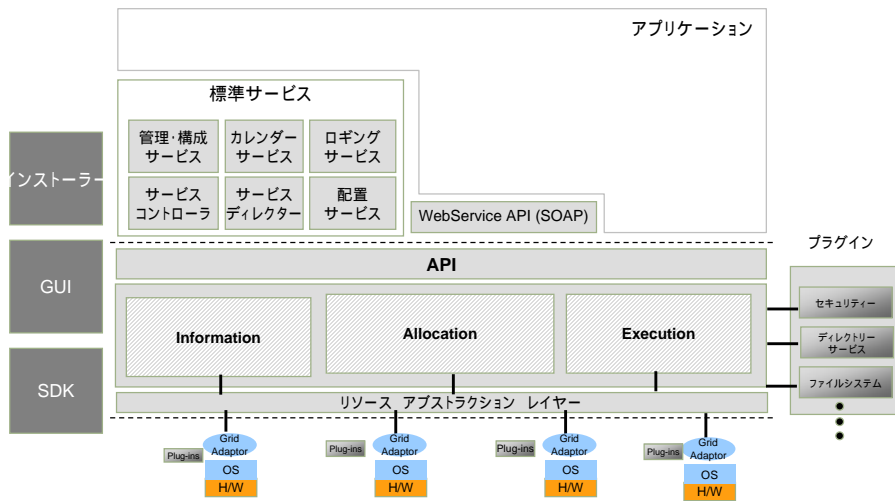
EGOによるユーティリティ型リソースマネジメント



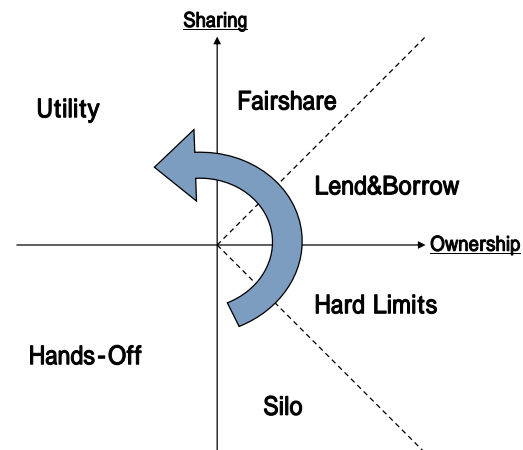
Platform ソリューションスタック



EGO アーキテクチャー



リソース管理形態の変遷 - Ownership vs Sharing -



- HPCバッチ向けスケジューラーとして特化
 - 新しいバージョンのOS, CPUタイプのタイムリーなサポート
 - スケジューリングオプションの充実
 - スケーラビリティ向上
 - スループット向上
- Platform他製品とのポリシーモデル統一化