

クラウド技術とオープンソース

2009/6/25

藤田昭人

大阪市立大学大学院 創造都市研究科

IIJ Innovation Initiative

自己紹介

- 社会人大学院(博士課程)の学生です
 - 大阪市立大学大学院 創造都市研究科に在籍しています
 - 研究テーマは“構造化オーバーレイの応用”です
- IIJ Innovation Institute の研究員です
 - 同社の第1回技術開発公募に合格しました
 - 2010年には自分の会社を設立しなければならない
- 私にとってクラウド・コンピューティングとは・・・
 - 起業のネタです
 - 構造化オーバーレイの Scale Out する特性はクラウドに役立ちます
 - クラウド技術は今多くの人の関心を集めるITテクノロジーの1つです

「クラウド・コンピューティング」という バズ・ワード

- ・ 未だ謎の多いバズ・ワードです
 - 2007年11月のコンファレンスでGoogleのCEOが語ったのが最初
 - 以前は Google の採用面接で問われる質問でした？
- ・ クラウド・コンピューティングと言えば、当初は・・・
 - Googleのサーチ・エンジンのアーキテクチャを多目的に利用するという意味？
 - Amazon の EC2/S3 まで含めると文字通り「雲を掴むような話」
→ この段階で十分混乱していたんだけど
- ・ ところが2009年になった途端
 - いろんな人がいろんな立場でいろんなことを言い始めた
 - もう、まったく訳がわからなくなってしまって・・・

クラウド・コンピューティングの現状

- 大規模Webサービスのためのインフラストラクチャ技術が大前提
 - 無料の情報サービスを安定的に提供するために必要だった
 - Google: 広告収入だけで維持できるコスト
 - Amazon: ECのカスタマ・サポートのコスト範囲で収まる
 - 低コストのためにスケール・メリットを究極まで追求
 - コモディティ・ハードウェアによる大規模クラスタ
 - 大規模分散システム・ソフトウェアの開発
 - 低コストを最優先にした自社データセンタの建設
 - 利用者に向けたプログラミング環境の提供
 - 何が目標なのか今ひとつハッキリしない
- ソリューションに対する明確なビジョンが定まらないままブーム到来
 - 各社がそれぞれの思惑で・・・
 - Windows Azure: Microsoftの.NETの普及活動の一環？
 - Open Cloud: Sunを中軸にその他大勢も名乗りをあげ？
- ビジョン不在で大混乱状態

クラウド・コンピューティングのビジョン

- 論文: **Above the Clouds: A Berkeley View of Cloud Computing**
 - <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>
 - UC Berkeley のコンピュータ・サイエンスの教授陣が書いた論文
 - RAD laboratory (<http://radlab.cs.berkeley.edu/>) での知見に基づく
 - クラウド・コンピューティングのビジョンに言及した数少ない文献
- その主張によれば「クラウド・コンピューティング」とは・・・
 - SaaS 形態で提供されるユーティリティ・コンピューティング
 - コンピュータ・リソースを社会的なリソースとみなし
水道、電気、ガス、電話と同じような社会的なサービスとして提供する
- ユーティリティ・コンピューティングとは・・・
 - 1961年にマッカーシが提示したアイデア
 - 社会的な需要も支えるインフラのコストも非現実的だったのでお蔵入り
 - Google や Amazon のインフラの事例から急に現実味を帯びてきた

ユーティリティ・コンピューティングの 3つのクラス(1)

- ユーティリティ・コンピューティングは計算リソースの再分配が基本
 - 必然的に計算リソースの仮想化の議論になる
- 既存のクラウド・コンピューティングの事例は3つのクラスに分類できる
 - Amazon EC2/S3
 - 物理ハードウェアを仮想化 → OSさえ選択可能
 - スケーラビリティやフェールオーバーなどの自動化はサポート困難
 - Google AppEngine/Force.com
 - 特定のアプリケーション領域に特化 → 汎用的な計算処理には向かない
 - スケーラビリティやフェールオーバーなどの自動化をサポート
 - Windows Azure
 - 両者の中間に位置する
 - 幾つかのアプリケーション領域を選択できるがOSまでは無理
 - スケーラビリティやフェールオーバーなどの自動化はある程度サポートされるがアプリケーション側での対応が必要になる
- どのクラスがクラウドの主流になるのか、まだよく分からない

ユーティリティ・コンピューティングの 3つのクラス(2)

	Amazon Web Services	Microsoft Windows Azure	Google AppEngine
計算処理モデル (VM)	<ul style="list-style-type: none"> ●Xen VMによる x86 インストラクション・セット アーキテクチャ (ISA) ●計算処理の柔軟性はスケーラビリティを許容するが、開発者がマシンを構築しなければならない あるいは RightScale などのサードパーティ VAR が提供するものを使用しなければならない 	<ul style="list-style-type: none"> ●Microsoft Common Language Runtime (CLR) VM; 管理された環境下で共通中間形式が実行される ●マシンは(例えば"roles" から複製された)宣言的な記述に基づいて供給される 自動的にロードバランシングが行われる 	<ul style="list-style-type: none"> ●予め定義されたアプリケーションの構造とフレームワーク プログラマによる Python で書かれた「ハンドラ」のすべての状態は永続的で(Python コードの外部にある)MegaStore に格納される ●計算処理とストレージは自動的にスケール・アップ(ダウン)される ネットワークとサーバーのフェールオーバーは3階層Webアプリ構造で一貫させる
ストレージ・モデル	<ul style="list-style-type: none"> ●モデルには EBS から SimpleDB まで幅がある ●使用しているモデルに依存して、自動スケールは共有のみ (EBS) から完全自動 (SimpleDB, S3) まで変化する ●一貫性の保証も使用しているモデルに依存して変化する ●API も標準的からプロプラエタリまで変化する 	<ul style="list-style-type: none"> ●SQL Data Services (制限付き SQL Server) ●Azure storage service 	<ul style="list-style-type: none"> ●MegaStore/BigTable
ネットワーキング モデル	<ul style="list-style-type: none"> ●IP レベル・トポロジの宣言を指定する配置の詳細は内部的に考慮される ●Security Groups がノードからの通信の制限を可能にする ●Availability zones が独立したネットワークの停止の抽象化を提供する ●Elastic IP addresses が永続的なルーティング可能な名前を提供する 	<ul style="list-style-type: none"> ●アプリケーション・コンポーネントでのプログラマの宣言記述に基づき自動化 	<ul style="list-style-type: none"> ●3階層Webアプリ構造に対応する固定的なトポロジ ●スケール・アップ、ダウンは自動的に行われ、プログラマには見えない

クラウド・システムとオープンソース

- アーキテクチャ的には2種類に大別できる
 - 特定のハードウェアを仮想化したシステム
 - コマーシャルベースの事例
 - Amazon EC2/S3
 - IBM Blue Cloud
 - オープンソース実装
 - Eucalyptus (<http://open.eucalyptus.com/>) Amazon EC2 のクローン
 - 抽象化されたエンジンによる仮想化システム
 - コマーシャルベースの事例
 - Google App Engine
 - Microsoft Windows Azure
 - オープンソース実装
 - Hadoop
 - CloudStore (Kosmos File System)
 - Hypertable

クラウド・システムのアーキテクチャ

アプリケーション: サービス利用者が作成するアプリケーション

分散並列処理

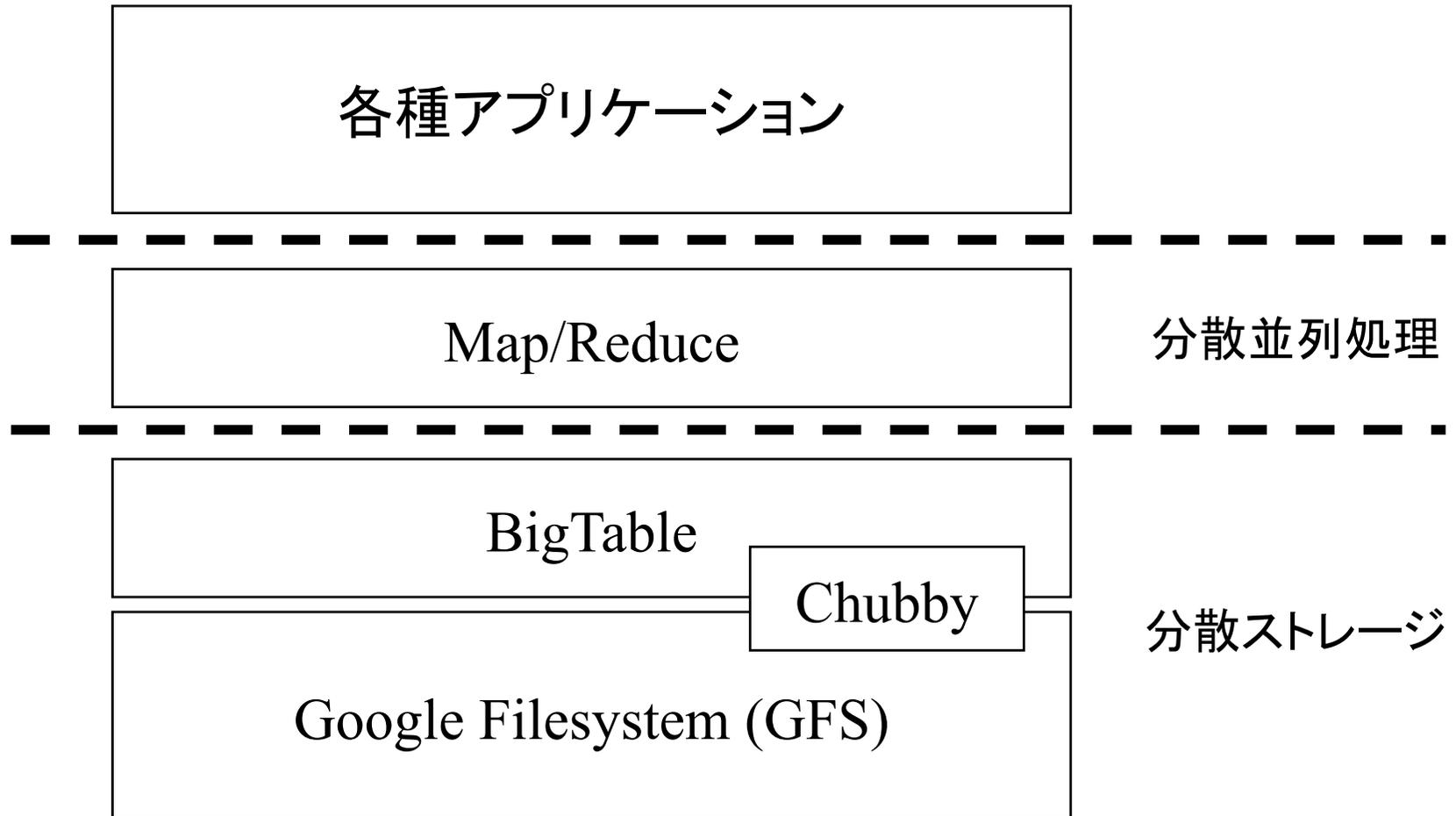
仮想エンジン: アプリケーションを実行する仮想化エンジン
クラウド・サービスの個性に反映される。既存のクラウド・ベンダーは各社各様。

分散ストレージ

データベース: アプリケーションが使用するデータベース
Web アプリケーションの場合は実行中のコンテキストも格納される。

データストレージ: ファイルを格納するデータ・ストレージ
DB のストレージとしても利用される場合もある。

Googleのサーチエンジンのアーキテクチャ



Apache Hadoop (1)

- 概要
 - クラウド関連では最もメジャーなオープンソース・ソフトウェア
 - 元々は Apache Lucene(検索エンジン)の一部として開発されていた
 - Google のような分散並列検索エンジンを実現するサブ・コンポーネント？
 - 現在は事実上 Yahoo の開発プロジェクト
 - Hadoopの主要開発メンバーはYahooに雇用されてるらしい？
 - Yahoo は50人体制でHadoopの開発を進めているとか？
 - Yahoo のデータセンタでは数千台規模で動作している？
 - Hadoop を利用した様々なアプリケーションが開発されている
 - デファクト・スタンダードに近づいている？
- ホームページ
 - <http://hadoop.apache.org/>

Apache Hadoop (2)

- コンポーネント
 - Hadoop Core
 - MapReduce と Hadoop DFS (HDFS) を含む基本コンポーネント
 - Google の MapReduce/Google File System (GFS) のクローン
 - HBase
 - Hadoop 版 BigTable
 - BigTable の基本的なアーキテクチャをほぼ踏襲
 - Hive
 - 「データウェアハウス」と銘打たれた、もう1つの分散データベース実装
 - SQLライクなデータベース
 - その他にもいろいろ
 - Chukwa MapReduce を使ったデータ収集・解析システム？
 - Pig 大規模データセット解析のための高水準言語プラットフォーム？
 - ZooKeeper Hadoop 上で動くアプリケーションの管理ツール？

Apache Hadoop (3)

- ドキュメント
 - 各コンポーネントのホームページ・・・結構しょぼい
 - Hadoop Wiki (<http://wiki.apache.org/hadoop/>)
 - 僕も UNIX Magazine 2009 Spring に記事を書いています
- 利用事例
 - Hadoop Wiki — PoweredBy (<http://wiki.apache.org/hadoop/PoweredBy>)
 - 50サイト以上の利用事例が簡単に紹介されています
- サポート
 - Cloudera (<http://www.cloudera.com/>)
 - 各種ディストリビューション向けの Hadoop パッケージを配布している

CloudStore (Kosmos File System)

- 概要
 - もう1つのオープンソース GFS クローン
 - アーキテクチャは GFS を踏襲
 - MetaServer のバックアップは未サポート？
 - C++による実装
 - Linux のシステムコールを使うので HDFS に対し性能的なアドバンテージ？
 - 実装コードは HDFS より読みやすい
 - Hadoop 0.18 のファイルシステム・インターフェースにより Hadoop と接続可
 - Hadoop 環境で HDFS からの置き換えが可能
 - Hadoop 付属のベンチマーク・プログラムで性能比較の作業中
- ホームページ
 - <http://kosmosfs.sourceforge.net/>

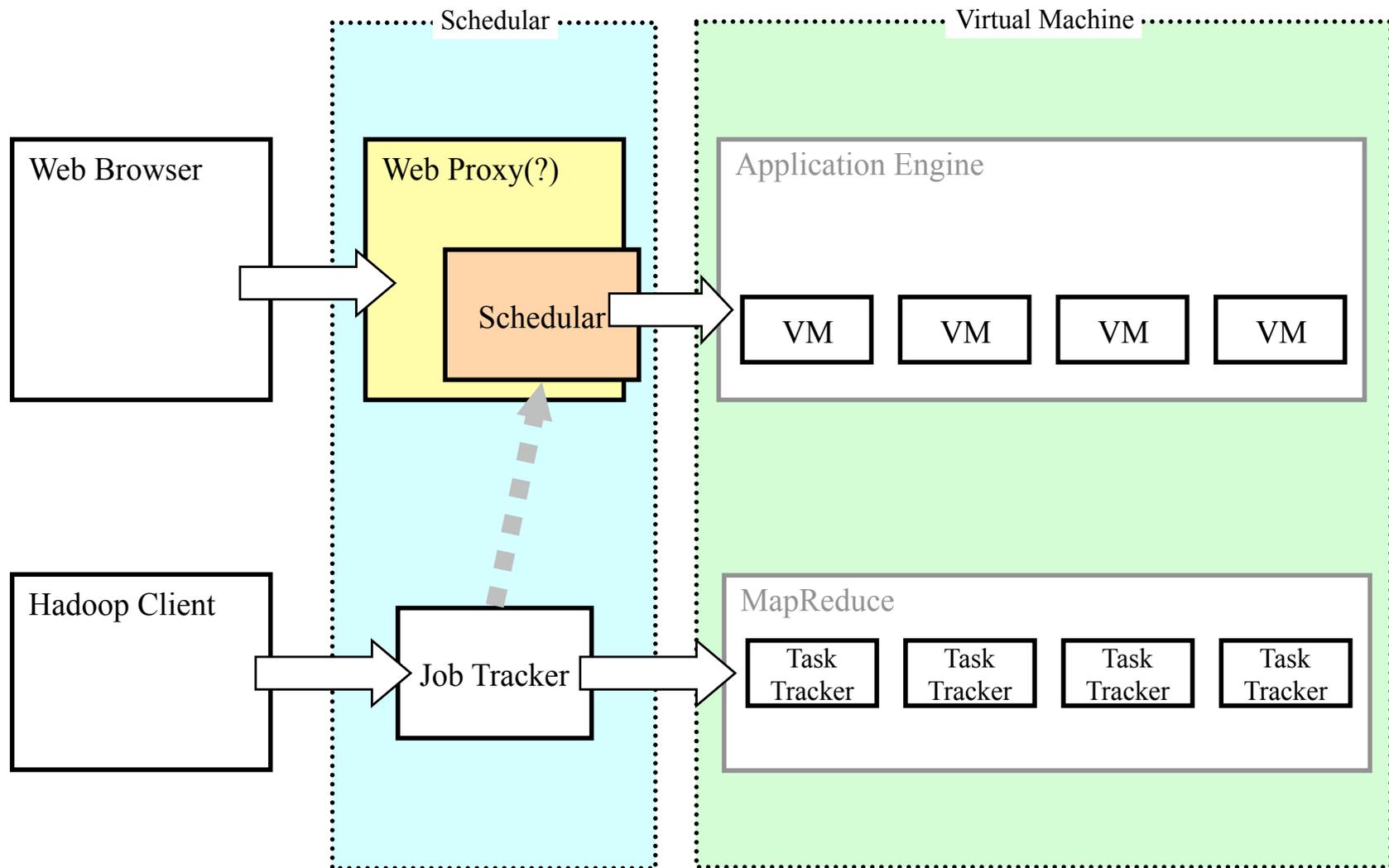
Hypertable

- 概要
 - もう1つのオープンソース BigTable クローン
 - HBase のライバル
 - 当初は1つの開発チームだったが設計方針を巡って喧嘩別れしたらしい
 - C++による実装
 - これが喧嘩別れの原因らしい
 - Hyperspace (Google の Chubby のクローン) が付属
 - 分散ロックマネージャですね
 - Thrift を使ったクライアントAPIのサポート
 - 様々な言語に対応・・・Ruby からの直接コールも可能
- ホームページ
 - <http://www.hypertable.org/>

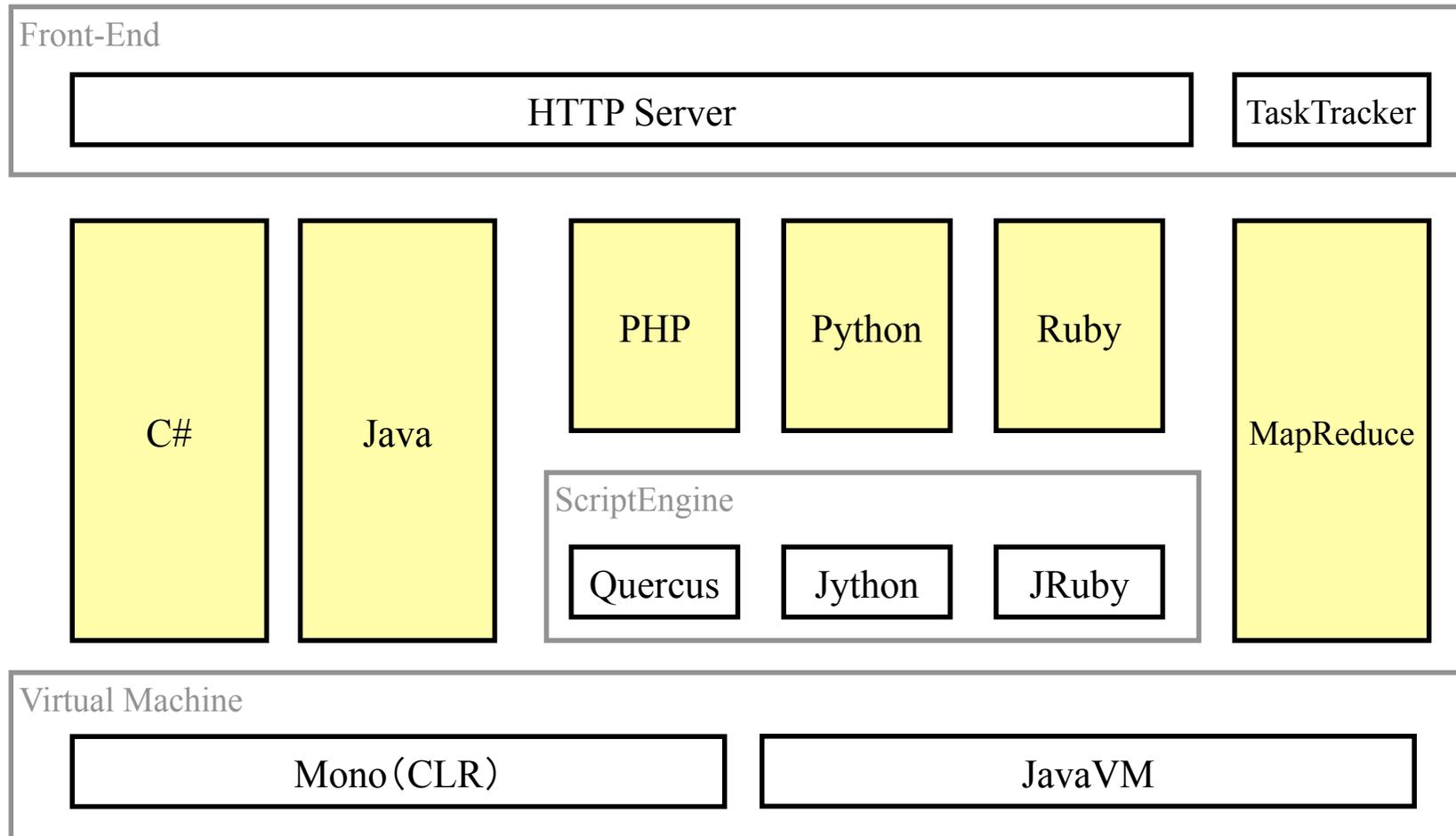
クラウド系オープンソースの活用(1)

- プロプライエタリであったクラウド技術も今では開かれた状態
 - Google の4システムはオープンソース・クローンで再現可能
 - MapReduce → Hadoop MapReduce
 - BigTable → HBase/Hypertable
 - Chubby → Hyperspace
 - GFS → Hadoop DFS/Kosmos File System
 - これをベースにサーチエンジンの構築？
 - Apache には Lucene(全文検索エンジンのためのツールキット)もあります
 - Google AppEngine や Windows Azure モドキも実現可能？
 - 仮想化エンジンは JVM や Mono を利用するとして…
 - スクリプト・エンジンも載せれば Python や Ruby もOK
 - あとスケジューラさえあれば…
 - Hadoop MapReduce の JobTracker を参考にすれば作れる？

クラウド系オープンソースの活用(2)



クラウド系オープンソースの活用(3)



オープンソース・クローンによるクラウドシステム

- 実際にどんな需要があるのだろうか？
 - 個人であれば・・・
 - オープンソース・プロジェクトの開発ネタとしては面白い
 - 自己学習の題材や教材にもなるかも？
 - 大学や研究機関なら・・・
 - それ自体が研究の対象になる
 - 研究用のプラットフォームとしても使える
 - それなりに信頼性のある低コスト情報サーバーとしても使える
- それだけ？
 - そうでもなさそうな雲行きに見えるんだけど・・・個人的には
 - 「プライベート・クラウド」だとか・・・
 - 「クラウドのインターオペラビリティ」だとか・・・

パブリック・クラウドとプライベート・クラウド

- パブリック・クラウド
 - Google や Amazon などが提供するクラウド・サービス
 - データのロック・インやデータ転送のオーバーヘッドが懸念される
→ 企業のクラウド・コンピューティングへの関心を抑制する
- プライベート・クラウド
 - プライベート空間(企業内ネットワーク)でクラウド・システムを構築し
管理・運営を行うコンピューティング・ネットワーク
 - データのロック・インの解消策
- サージ・コンピューティング (Surge Computing)
 - 通常はプライベート・クラウドのみで処理を行い、
一時的な過負荷の際にのみパブリック・クラウドを利用する
- パブリック-プライベート連携
 - クラウド・サービスの最大の弱点はネットワークなので
 - 様々なシチュエーションで「パブリック-プライベート連携」の課題は出てくる

クラウド間のインターオペラビリティ

- 背景にはクラウド・サービスの「データ・ロックイン」問題があります
 - 「現在は相互運用性がないので最初の選択に引きずられる」が建前
 - この領域で先行するメジャー・プレイヤーの優位性を封殺する意図？
- OpenCloud
 - Sun が中心のクラウドAPIの標準化の試み
 - Google、Amazon、Microsoft はいずれもそっぽを向いている
 - 例によってIBM はここでもいい顔をしている
- 日本国内では・・・
 - 「クラウド」は景気対策のための財政出動の口実になっているので
 - その成果というか落とし所の1つとして当然クローズアップされる？
- しかし・・・
 - 「これってどこかで見たような構図でないかい？」って思うのは僕だけ？

「まとめ」・・・の代わりに

- まとまらないのは、今そういう状況にあるから
 - 今後混迷の度合いはさらに深まる・・・少なくとも今年度は・・・ってことにしておいてください
- 個人的には後悔しています・・・かなり
 - だって去年の今頃はこんな展開になるとは思ってなかったし
 - 今年になった途端、急に風向きが変わった・・・というのが実感
 - そもそも動機が不純すぎたかな？
 - P2Pをマスクングできるパスワードを探したのが間違い？
 - 査読論文のことをチラッとでも考えたのが間違い？
 - 開き直って「金になるなら、なんでもやるでえ～～」・・・って立場です今は
- 結局・・・しばらく騒ぎは続きます

クラウド・コンピューティングの課題(1)

- **Above the Clouds** では次の10の課題と対策を述べている(つづく)
 1. サービスの可用性
 - 複数のクラウド事業者の利用
 - 柔軟性を活用した DDOS 攻撃からの防御
 2. データのロック・イン(囲い込み)
 - API の標準化
 - サージ・コンピューティングを可能にする互換ソフトウェアの実現
 3. データの秘匿性と監査可能性
 - 暗号化、VLAN、ファイヤーウォールの展開
 - 地理的条件を考慮したデータ格納による各国の法律への適応
 4. データ転送のボトルネック
 - ディスクを FedEx で送る
 - データのバックアップとアーカイブ化
 - 広域ルーターの低コスト化
 - 高帯域LANスイッチ

クラウド・コンピューティングの課題(2)

- **Above the Clouds** では次の10の課題と対策を述べている(つづき)
 5. 予測不能なパフォーマンス 仮想マシン・サポートの改善
 - フラッシュ・メモリ
 - VMへのギャング・スケジューリングの導入(HPCアプリケーション用)
 7. スケーラブルなストレージ
 - スケーラブルなストレージの開発
 8. 大規模分散システムのバグ
 - 分散VMのデバッガの開発
 9. スケーリング設定に関わる問題
 - マシン・ラーニングによる自動スケーラーの開発
 - クラウド・コンピューティングによる自然保護主義を奨励するためのスナップショット
 10. 悪評判の共有
 - 電子メールと同じような評判を守るサービス
 11. ソフトウェア・ライセンス
 - 従量課金型のライセンス
 - バルク利用向け販売