今後のAsakusaの方向性

NAUTILUS

Agenda

- 1.現状認識
- 2.日本市場
- 3.Hadoop/Spark
- 4.新しいアーキテクチャ
- 5.Asakusaの方向性
- 6. New Engine

1.現状認識

- ビックデータ・IoT
  - なかなかビジネスになっていない
    - 少なくとも何兆円というマーケットに残念ながらなっていない
  - ビックデータ
    - 結局は従来からのCRMとあまり変わっていない
      - MLにしても、もともとある手法の延長ではある
    - よって、取れるビジネスメリットも変わらない
    - ほとんどの案件が既存DWHのリプレース案件
      - ベンダー的にはレッドオーシャン
      - 一足飛びにAIという話になっているが、そもそも機械学習とAIはかなり異なる
  - IoT
    - 最終イメージまでできていないとなかなか大規模な投資にはならない。
    - PoC案件が多い
    - まだまだ手探りの状態
    - Internetではなく、Intranetであることが多い

1.現状認識

- ビックデータは既存延長・IoTはPoCフェーズ
  - したがって、システムとしては大規模にはなりにくい
  - システム的な付加価値がそのままユーザの売上になりにくい

- ユーザーのスタンス
  - ユーザーサイドは大きな金額を突っ込める状況ではない。
  - 「小さく・中規模で」・「お金をかけずに」

  - むしろ既存システムの更新処理の方に時間と人をさかないといけない状況
  - 既存DWHリプレースにしても、線形でのデータ増加であり、級数的に増えているわけではない
    - 勿論例外はある

- 上記を客観的に見ていく必要がある
  - 近年はIT系マスコミと実際のビジネスの乖離が酷いので正確な状況がつかめない

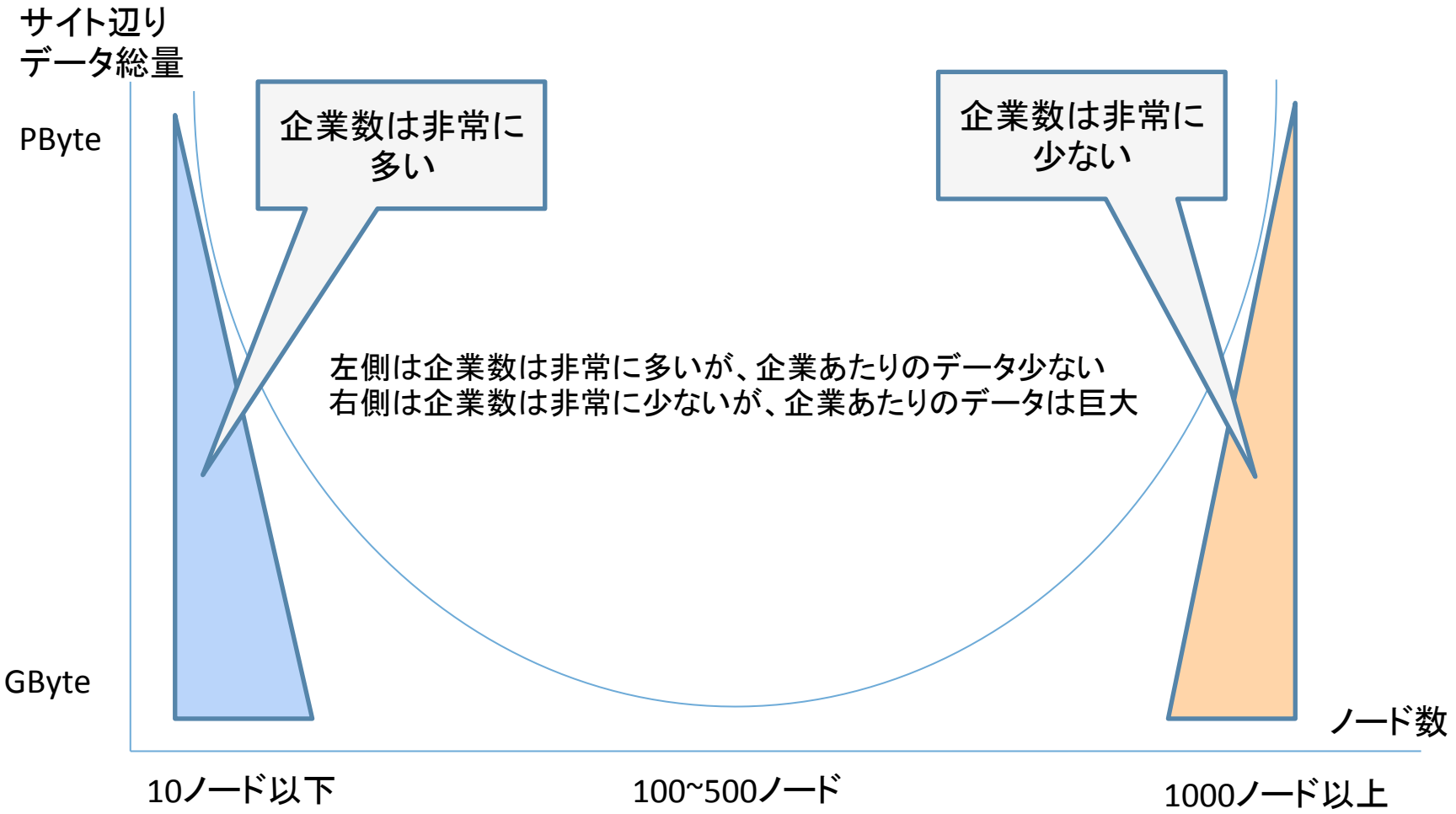
2.日本市場

- 小規模・中規模が圧倒的多数
  - 弊社の実績
  - 結果として、投資総額が低い
  - クラスターノード数も10-20ノードのサイズ件数として非常に多い
  - 特にエンタープライズ

  - 確かに大きなクラスターもあるが、数十Pクラスで、しかも運用的には大きなクラスターにして、割となんでも入れてしまうゴミ箱状態。
    - そもそも業務別にある程度クラスターをクラスター化した方がいいぐらい

- つまり、日本市場では「金額を押さえつつ＝ノードを大規模せずに、ITでのレバレッジをどう効かすか？」を考えないといけない
  - 実際のユーザニーズや実態はこのポイントにある

2.日本市場：どういうことかというと



日本は圧倒的に左側で、例外的にごく一部が右側

3.Hadoop/Spark

- Hadoop/Spark
  - まず、分散並列処理基盤としてきわめて優秀
  - 汎用機・大規模SQLバッチよりは、非常に高速に動くことがはっきりした

- データ集積の基盤としてはどうか？
  - データ集積という視点では、そもそも想定しているユースケースが、日本市場に合わない。
  - というよりもそもそも見ているマーケットサイズ・ユーザー規模が違う。
    - データ集積という意味では、結果として日本のマーケットでは、効率の悪い仕組みになっている

- そもそも多数のノードをどう統括するか？という視点が強い。少数・中規模のノードをどう効率化するか？という視点は薄い。

- 他に代替がないので使っているという感じであり、ベストではない

3.Hadoop/Spark

NAUTILUS

- 課題
  - 100ノード以上のクラスターが主要ターゲット
  - 何が問題なのか？
  - 故障・障害の実際発生率の違い

- 大規模クラスター
  - 1000ノードの場合、0.1%の障害発生は、常に1ノードは壊れていることを想定しないといけない
  - サーバが壊れるのは普通の状態
    - 「常にサーバが一つは壊れている」
  - 障害対策と復旧に特段の仕組みが必要
    - 自動化しないととても無理

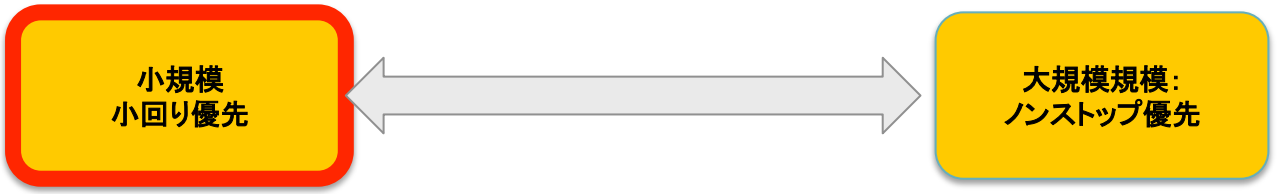
- 中規模・小規模クラスター
  - 10ノードのケースでは、0.1%の障害発生は、顕在化しない
  - サーバが壊れるのは異常事態
    - 障害対策はとりあえず止めて、復旧は手作業
    - 手動で間に合う（・・・まぁセンスはないが、現実はコレで十分）

3.Hadoop/Spark

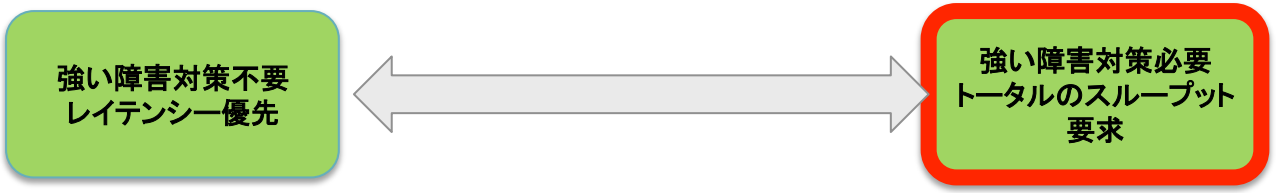
- 障害対策・復旧のアーキテクチャがそもそも論
  - Hadoop/Sparkに限らず、一般に分散クラスターでは、障害・復旧の仕組みは非常に難易度が高い
  - 常に何かあるので、その前提でノンストップにしなければならない
    - 何かあったら系全体がストップでは、事実上まったく動かない
  - 結果、往々にして以下のトレードオフが発生する
  - 1.アーキテクチャに制約が加わる
    - 複雑な処理は復旧が困難なんでなるべく簡単にする
    - 例）バリアーいれまくり
  - 2.オーバーヘッドを許容する
    - 余分にデータを持つ
    - 例）データ書きまくり・レプケーションとりまくり
  - 1+2→要するに「処理が遅くなる」

3.Hadoop/Spark

- 結局、何が問題なのか？
  - クラスター規模に合わせた、障害対策・復旧のアーキテクチャを選択できない
  - 本来はトレードオフなので、合わせて調整したい



日本市場はどっちかというと左より



規模に合わせて調整したいが・・・・できない
実際は選択の余地がなく、右よりになっている

4.新しいアーキテクチャ

- トレンド:「ムーアの法則」の終了
  - ハードウェア開発はムーアの法則が限界に来ている
  - メニーコア化
    - コア自体は微細化は限界ので、コア数を増やす
  - 周辺の強化
    - 特にメモリー周り
    - コアと使い切るようにメモリーバスを強化
    - IOスピードを上げる
    - ストレージのIOコストを減らす
- これは、結果として複数のサーバノードを集めてひとつのノードにする、という方向に倒れつつある。
  - インターコネクトの強化やメッシュ状のコア連携といった方向
- 「サーバクラスターの凝縮化（condensation）」
  - RSA(Intel)　TheMachine(HP)　Firebox(AMPLab)

4.新しいアーキテクチャ

- RSA
  - Intelの次世代アーキテクチャ
  - ラックスケール・アーキテクチャ
    - ラックのバックプレーンを共有化
    - 高速のインターコネクトを利用
    - RDMAをSoCで実装
    - NVMをDIMMソケットで連結
  - 1000コアが普通に透過的に利用可能
  - メモリーは基本的に共有化されて、普通に10数Tbyte
  - たかだか10~20台のサーバで構成される

4.新しいアーキテクチャ

- RSA
  - マーケット：日本市場に「合う」
    - データサイズが、RSAにはまる大きさ
  - すでに事実上の先行ベンチマークが行われている
  - 「日本市場でのOracleEXAデータのバカ売れ」
    - 非常に高価にもかかわらず、日本「だけ」で馬鹿売れ
    - RSAとほぼおなじ構成かつサイズ・規模がまったく同じ
  - 超高速で、OracleEXAの100分1の価格で、同じようなものが出たらどうしますか？
    - 普通に買うと思います
    - HadoopとかSparkとか買います？
  - なので、Asakusaはこの辺を狙っていきます。

5.Asakusaの方向性

- 現状を踏まえる
- 「金額を押さえつつ＝ノードを大規模せずに、ITでのレバレッジをどう効かすか？」に注力していく

- 先の対応はRSAになるが、まずは最初に1node startをベースにして、新しいアーキテクチャに対応して行く。

- すなわち、メニーコアに特化した仕組みを出していく。
- これは高いスループットと低遅延要求を満たす方向を目指す。
- Hadoop・Sparkが背負っている「100ノード前提」を排除して、身軽にすることでスピードを上げる
- **かつAsakusaDSLの互換性は100%維持**

- α版をすでにFixstars社と協同で開発中

6.New engine

- Asakusaの新しい実行エンジン
  - C++実装
  - パフォーマンス重視
    - Spark・Hadoopの余計な部分は全部除去
    - 純粋に速さに特化
    - コア性能・メモリー性能を「物理的に使い切る」ことが目標
    - 理論的には世界最速

- 試験的なパフォーマンスでSparkの5-10倍の速度
  - 現行のバッチ処理で言うと・・・
    - ヘビーSQL：8時間
    - Hadoop：20分
    - Spark:5分
    - C++：1分以下
  - 8時間のバッチ処理が1分を切るレンジまで来ている
    - 別に驚くことでなくて、メニーコアとバスを限界まで使えば本来、この位のアウトパフォーマンスは出るはず

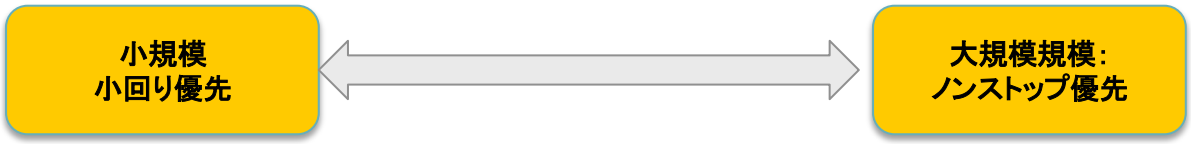
6.New engine

- まずは1-nodeスタート
  - したがってデータサイズはメモリーに乗るレベル
  - 近い将来RSAに対応
    - 結果として広いメモリー空間が利用可能
    - 日本のエンプラで十分な規模

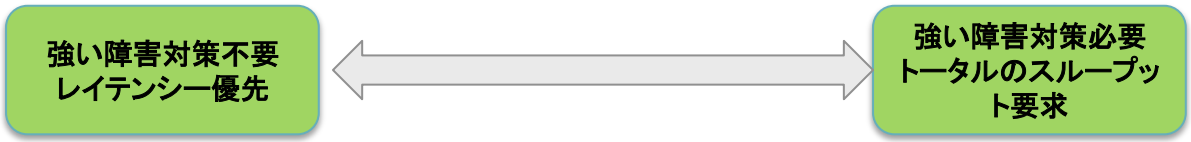
- 1-nodeのメリット
  - 導入が非常に簡単。一台でよい。(ただしメニーコアが前提)
  - スタートは最少投資でよい。
  - 各エッジでのプリプロセッサーとして利用可能
  - 工場・拠点・各センターで処理をしてから、IDCに送る
  - IoT的なことをやるときにも、通信費・IDCのコスト削減することができる。

6.New engine

- 投資可搬性の確保
  - 実行エンジンが複数選択可能
  - その上で、Asakusaで書いたアプリケーションはすべて実行可能
  - データがPクラスまで来て、サーバ台数が必要になった場合
    - そのままHadoop・Sparkに移行可能
    - アプリケーションは変更なし
    - テストもそのまま持って行ける



規模はさまざまになる



数ある分散環境のなかで、Asakusa「だけ」が選択が可能

6.New engine

- リリーススケジュール
  - 12月内部α
  - 2016/4　OSSとしてリリース（予定）
  - 詳細は担当営業に聞いてください！