IIJ Technical WEEK 2010

分散処理システムの動向とその運用

Internet Initiative Japan

2010年11月17日

株式会社インターネットイニシアティブ
前橋　孝広

Ongoing Innovation

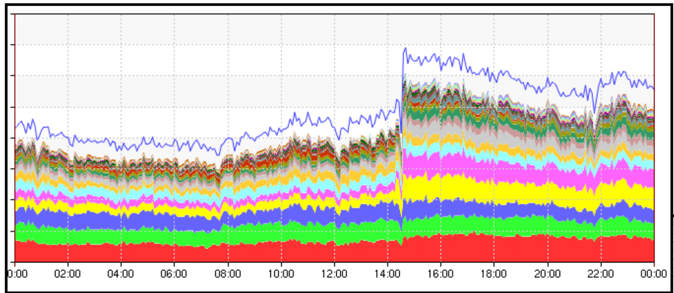
本日の内容

- 分散システムの概要と動向
- IIJにおけるBig Dataの扱い
- 運用

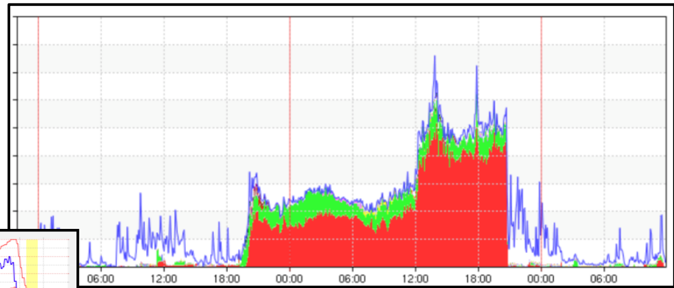
大量のデータ “Big Data” を扱いたい

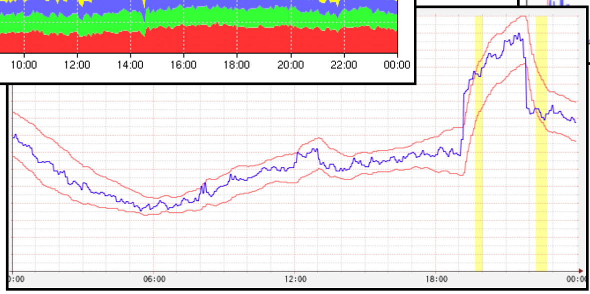
- **ISPにおいて、サービスの状態把握は必須**
  - 運用や設計を最適な状態に維持するため
  - 異常の検知・障害への対応のため
- **大量のトラフィックデータやログの分析が不可欠**

例: AS番号別トラフィック解析グラフ

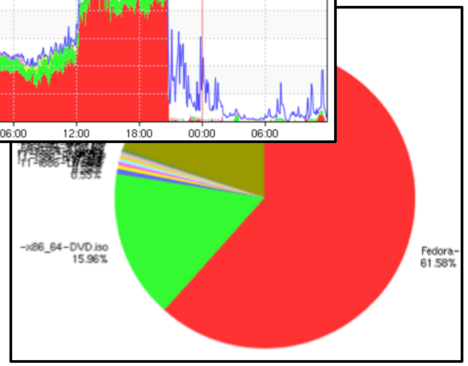


時系列アクセス数グラフ





異常トラフィック検出



アクセス数割合

Big Data を扱う上での問題点

データ量が膨大

機器故障による
データ消失

大規模
データ処理

大量のストレージ、
サーバの管理限界

検索や抽出に
時間がかかる

従来のデータベースからの脱却が必要

ISPに限らないBig Data

- **金融**
  - トランザクションの生データ
  - 会計バッチ処理
  - 値洗い
- **流通**
  - POSデータ
- **業務システムの大量のログ**
  - 内部統制
- ････

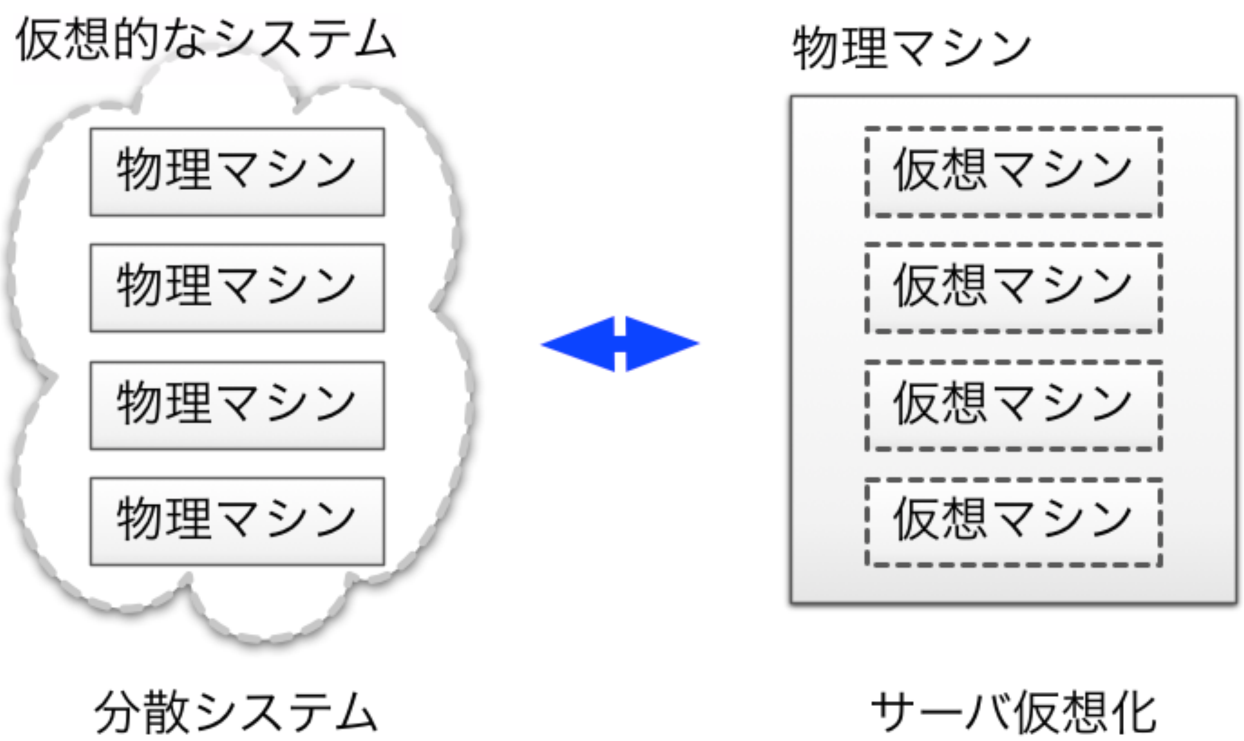
Big Data を扱うための手段: 分散システムとは?

- 利用者からは単一システムとして見える
- 多数のコンピュータノードの集合

- 一台で出来ないことをたくさんのノードを使ってやる

分散システムとサーバ仮想化

- サーバ仮想化 – 1台のマシンを複数に分割して利用率向上
- 分散システム – 複数のマシンを一つにまとめて強力なマシンを作り上げる



分散システムを使う理由

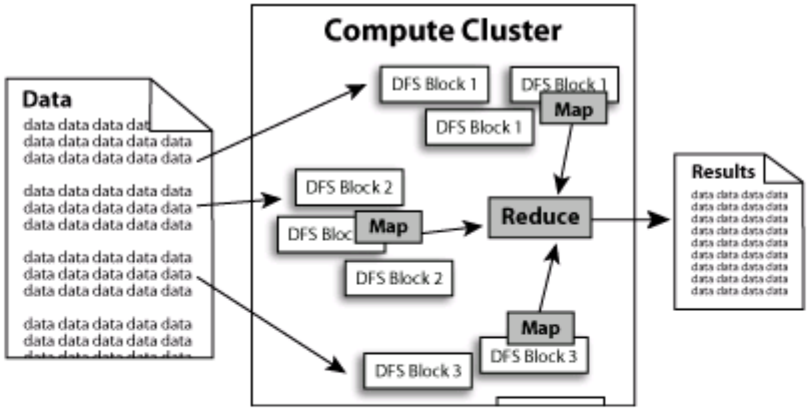
- **Performance**
  - 1台でできないことを複数台で分担することで性能向上
- **Availability**
  - 一部のマシンが壊れても全体としては動き続ける可用性
- **Elasticity**
  - マシンを追加するのが容易(動的追加・削除)
- **Cost**
  - 安価なマシンによるコスト削減

分散システムの具体例

- **Google**
  - GFS, MapReduce, BigTable, Dremel, Percolator, …
- **Amazon**
  - Dynamo
- **Hadoop**
  - オープンソース、HDFS, MapReduce, Hbase
- **Cassandra**
  - オープンソース
- **ddd**
  - IIJで開発、分散ストレージ+分散データ処理
- **…**

Hadoop

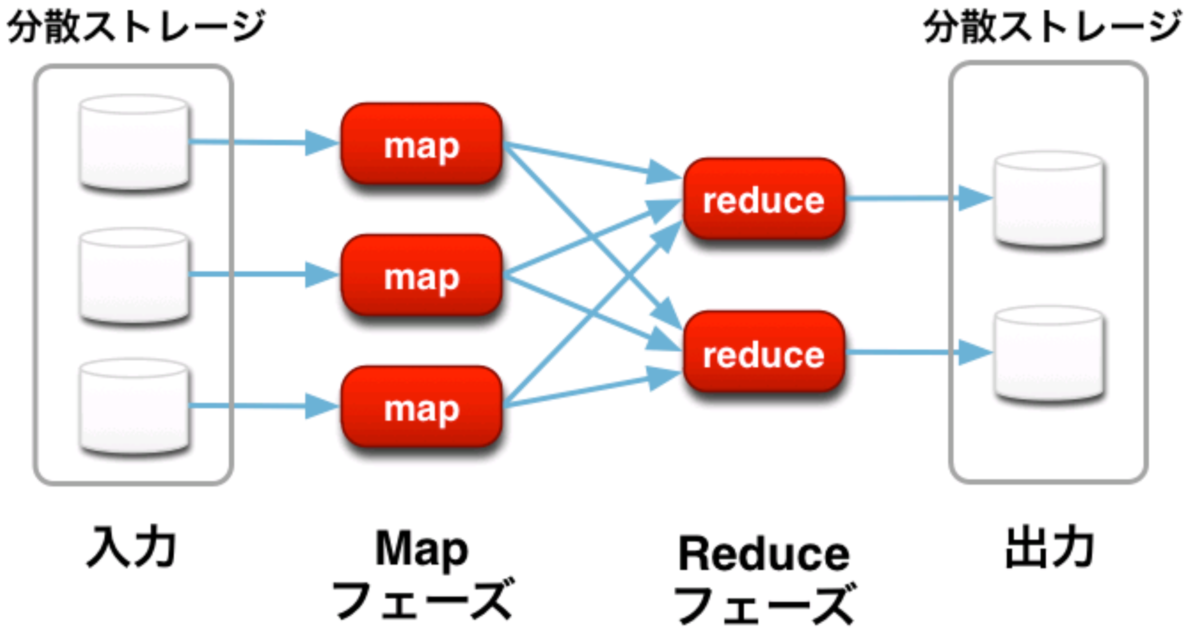
- **Hadoop DFS(HDFS), Hadoop MapReduce**
  - Google のGFS, MapReduceの論文を参照して実装



- **Hadoop Eco System**
  - HBase, Pig, Hive, ZooKeeper, ...

並列分散処理フレームワーク MapReduce

- mapとreduceの2段階にわけてデータ処理
  - ① map – 抽出・変換
  - ② reduce – 集約・集計
- 並列処理を抽象化



Hadoopの使用例

ebay

- Customer intimacy
- Precision targeting
- Driving top line growth

cloudera

Bank of America

- Risk analysis
- Customer insight
- Drive growth

So Much To See Today!

- Optimizing search — ORBITZ
- Advanced analytics in the Army — DIGITAL REASONING
- Using Flume &Hive for log data — mozilla Firefox
- Analyzing VOIP data with R — PURDUE UNIVERSITY

cloudera

Cassandra

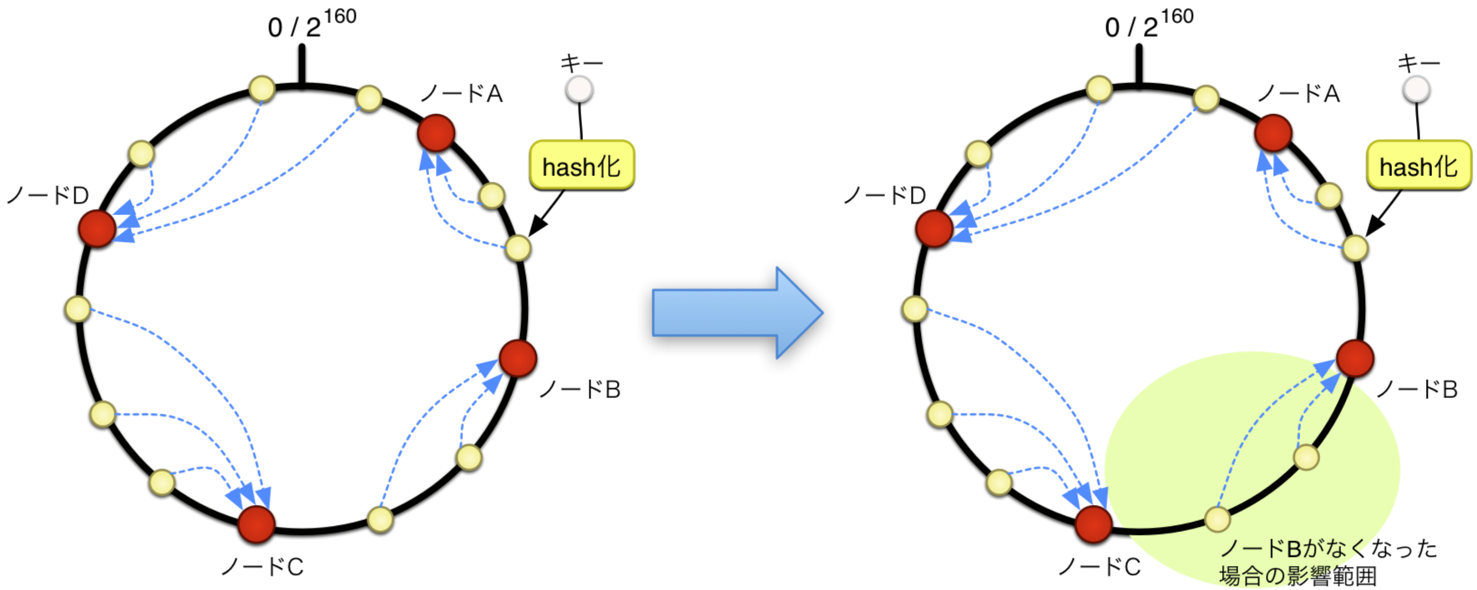
Cassandra = Dynamo + BigTable

- Dynamo – Amazon開発の分散キーバリューストア(KVS)
- BigTable – Googleが開発、構造化データを扱うための分散ストレージ

•高い耐障害性
•P2Pによる単一障害点がない構成
•結果整合性
•リッチなデータモデル
•Hadoopとの連携

コンシステントハッシュ法

- **マシンが増減しても影響を受けるkeyが少ないアルゴリズム**
  – 多くの分散KVSで採用されている
- **名前をキーにして複数ノードに分散**



NoSQL というムーブメント

- **NoSQL = Not Only SQL**
  - NoSQLの定義はない・・・
    - 分散システム、単純なデータモデル、トランザクションなし…
  - 適材適所
  - SQL(RDBMS)以外のデータストレージに注目しようという動きで、SQLを否定するものではない

- **例**
  - Key-Value Store: Tokyo Cabinet, Kyoto Cabinet
  - 分散KVS: Dynamo, Kai, kumofs, ddd
  - BigTable型: BigTable, HBase, Cassandra
  - ドキュメント指向: CouchDB, MongoDB
  - etc

適材適所

RDBMS
厳密な整合性
構造化・変化しないデータ構造
関係代数による集合演算

NoSQL
単純・柔軟なデータ構造
分散処理による大量のデータ処理

- **処理対象に適したシステムの選択**
- **すべてに万能なものはない**
  - トレードオフ
- **対象に合わせて細分化していく傾向?**

NoHadoop

- **Hadoop(MapReduce)では以下のような要求にこたえられない**
  - complex joins
  - ACID requirements
  - real-time requirements
  - supercomputing algorithms
  - graph computing
  - interactive analysis
  - the need for continuous incremental updates.

Beyond Hadoop: Next-Generation Big Data Architectures
http://gigaom.com/cloud/beyond-hadoop-next-generation-big-data-architectures/

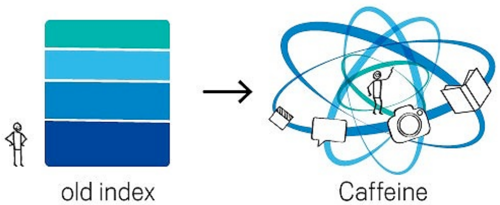
NoHadoop: 次世代Big Dataアーキテクチャ

- SQL
- Cloudscale
- MPI and BSP
- Pregel
- Dremel
- Percolator(Caffeine)

Beyond Hadoop: Next-Generation Big Data Architectures
http://gigaom.com/cloud/beyond-hadoop-next-generation-big-data-architectures/

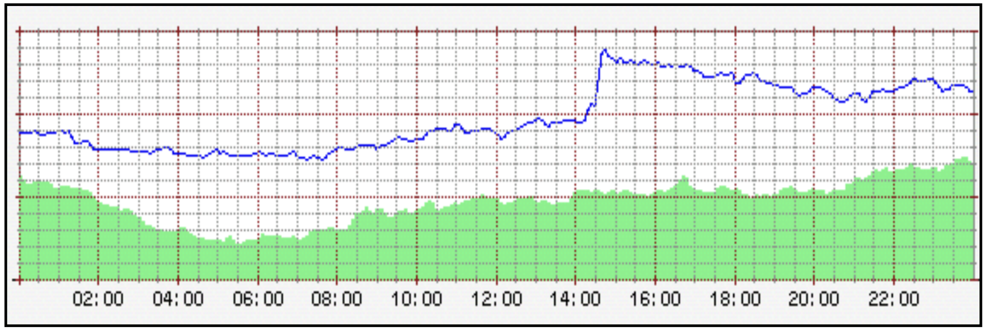
Dremel / Percolator

- **Dremel**
  - Webスケールデータを対話的に分析
  - 兆単位のレコードを数秒で集計
  - multi-level execution trees, columnar data layout
  - 数千のCPU、ペタバイト級のデータ
- **Percolator**
  - 新検索エンジン“Caffeine”で使用
  - incremental processing
  - ACID transaction

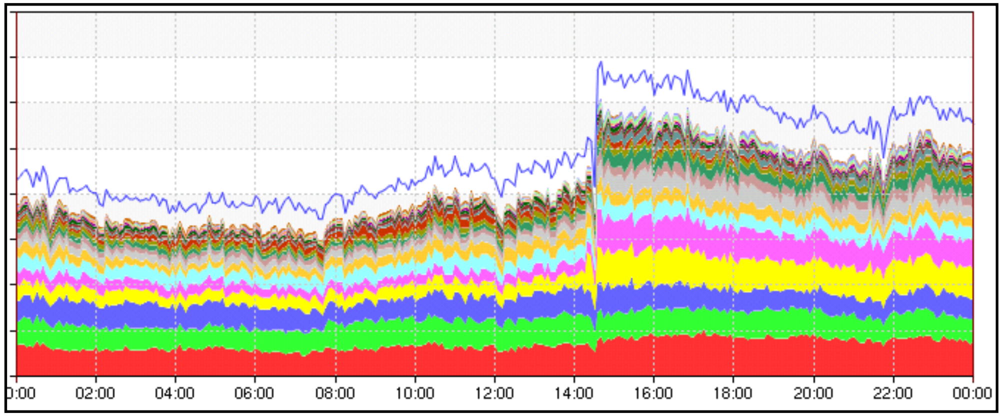


IIJにおけるBig Data

- トラフィック計測データの例



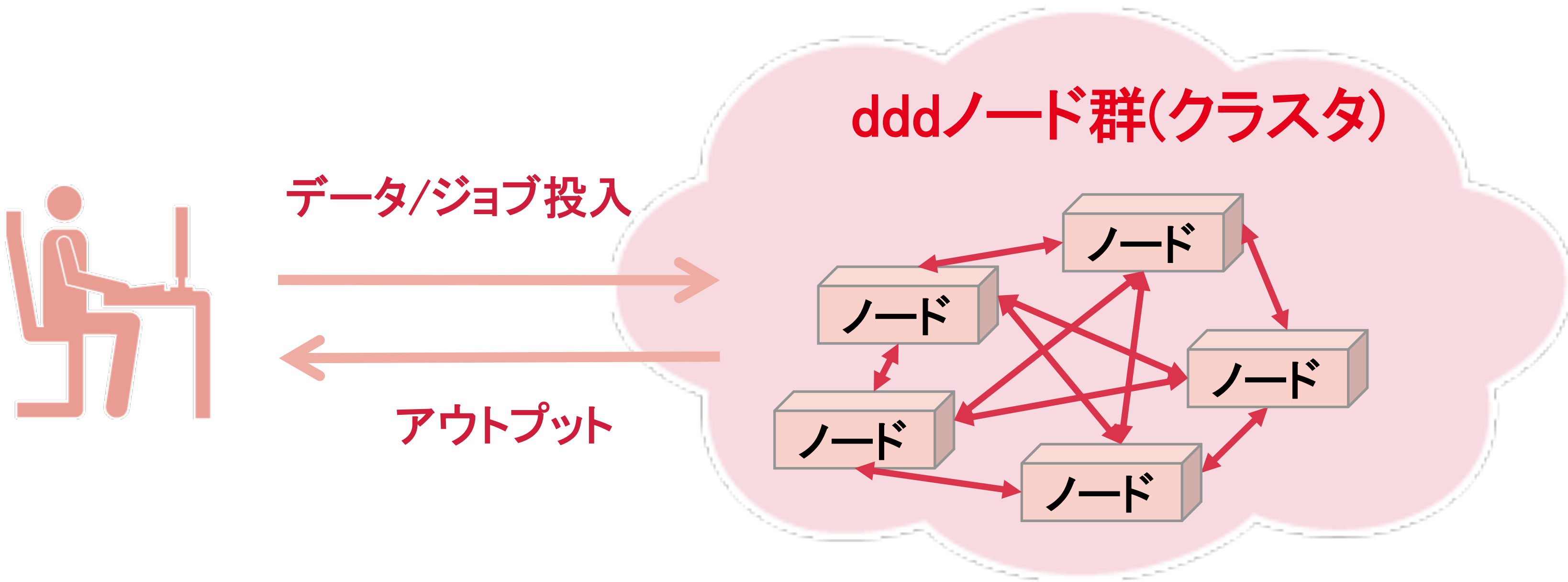
SNMP
1ヶ月分 1Mバイト程度



NetFlow
5分で数十Mバイト程度

ddd とは?

- IIJが独自開発した分散システムソフトウェア
  - 分散ストレージ + 分散データ処理機能(MapReduce)
- IIJ内部で使われている技術要素のひとつ



イメージ図

dddの特徴

- 独自開発の分散システム
- スケーラブル
  – 一台では扱えないような量のデータを高速に処理
- 高可用性
  – 一部のノードが故障してもシステムとしては動きつづける
- 柔軟性
  – 動的にノードの追加・削除可能

ddd開発の経緯

- **ddd以前(〜2007)**
  - リレーショナルデータベースを使ってデータを格納
  - 数千万レコード程度が限界、スケールしない
- **要求事項**
  - 大量のデータを保存できること(数百億レコード超)
  - 任意の条件で高速にデータを抽出・集計できること
  - そのかわりupdateもJOINもトランザクションも必要ない

要求にぴったりあうデータベースは存在しない

↓

独自開発

例: フロー統計情報のデータ構造

- 列数は少なく行数が多い

約30項目

約
1000億
行

time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	…
1288253400	172.16.100.1	1433	1	6	2497	9999	80	4321	192.168.0.1	192.168.9.8	
1288253400	172.16.0.10	987	1	6	1234	1000	25	2224	10.168.20.1	10.168.20.1	
1288253400	172.16.100.2	555	1	6	9999	2497	2345	80	192.168.0.7	10.18.80.1	
1288253400	10.16.10.1	12	1	1	1111	2222	22	22345	10.168.1.1	10.168.30.1	
1288253400	172.16.0.1	1490	2	17	2222	1234	22345	24	172.16.9.7	10.68.20.1	
1288253401	172.16.1.10	1309	1	6	3333	5555	8080	8765	192.168.0.1	172.16.9.79	
1288253401	192.168.10.1	538	1	6	1111	2497	5787	80	10.218.0.1	192.168.9.1	
1288253401	172.16.99.8	333	1	1	9999	1000	70	15588	10.68.0.1	172.16.9.72	
1288253401	10.16.98.7	981	1	17	4444	2222	80	11557	192.168.0.1	172.16.90.3	
1288253401	172.16.10.1	1433	1	6	1234	4321	7078	12345	172.16.0.1	172.16.4.71	
…											

注: 保存期間によって量は変動する

Big Data の分割

- 単位時間・ルータ単位で区切る

例: 5分

time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	…
1288253400	172.16.100.1	1433	1	6	2497	9999	80	4321	192.168.0.1	192.168.9.8	
1288253400	172.16.100.1	987	1	6	1234	1000	25	2224	10.168.20.1	10.168.20.1	
1288253400	172.16.100.1	555	1	6	9999	2497	2345	80	192.168.0.7	10.18.80.1	
1288253400	172.16.100.1	12	1	1	1111	2222	22	22345	10.168.1.1	10.168.30.1	
…											

5分

time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	…
1288253700	172.16.100.1	1309	1	6	3333	5555	8080	8765	192.168.0.1	172.16.9.79	
1288253700	172.16.100.1	538	1	6	1111	2497	5787	80	10.218.0.1	192.168.9.1	
1288253700	172.16.100.1	333	1	1	9999	1000	70	15588	10.68.0.1	172.16.9.72	
1288253700	172.16.100.1	981	1	17	4444	2222	80	11557	192.168.0.1	172.16.90.3	
1288253700	172.16.100.1	1433	1	6	1234	4321	7078	12345	172.16.0.1	172.16.4.71	
…											

5分

time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	…
1288253400	172.16.200.2	1433	1	6	2497	9999	80	4321	192.168.0.1	192.168.9.8	
1288253400	172.16.200.2	987	1	6	1234	1000	25	2224	10.168.20.1	10.168.20.1	
…											

分散配置

- ルータ名・時刻を元に名前を付ける
- 名前をキーに、コンシステントハッシュ法により分散配置
- 異なる3つのノードに複製

r01/2010/00001

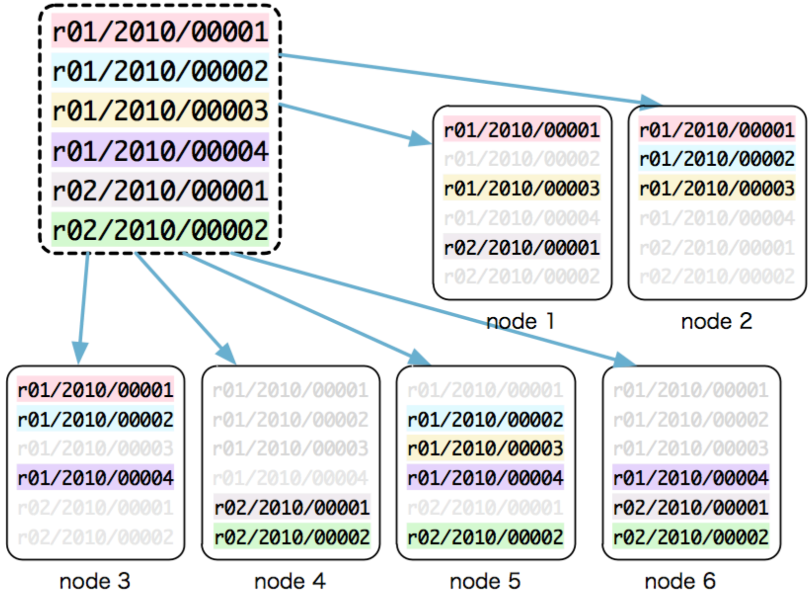
time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	...
1288253400	172.16.100.1	1433	1	6	2497	9999	80	4321	192.168.0.1	192.168.9.8	
1288253400	172.16.100.1	987	1	6	1234	1000	25	2224	10.168.20.1	10.168.20.1	
1288253400	172.16.100.1	555	1	6	9999	2497	2345	80	192.168.0.7	10.18.80.1	
1288253400	172.16.100.1	12	1	1	1111	2222	22	22345	10.168.1.1	10.168.30.1	
...											

r01/2010/00002

time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	...
1288253700	172.16.100.1	1309	1	6	3333	5555	8080	8765	192.168.0.1	172.16.9.79	
1288253700	172.16.100.1	538	1	6	1111	2497	5787	80	10.218.0.1	192.168.9.1	
1288253700	172.16.100.1	333	1	1	9999	1000	70	15588	10.68.0.1	172.16.9.72	
1288253700	172.16.100.1	981	1	17	4444	2222	80	11557	192.168.0.1	172.16.90.3	
1288253700	172.16.100.1	1433	1	6	1234	4321	7078	12345	172.16.0.1	172.16.4.71	
...											

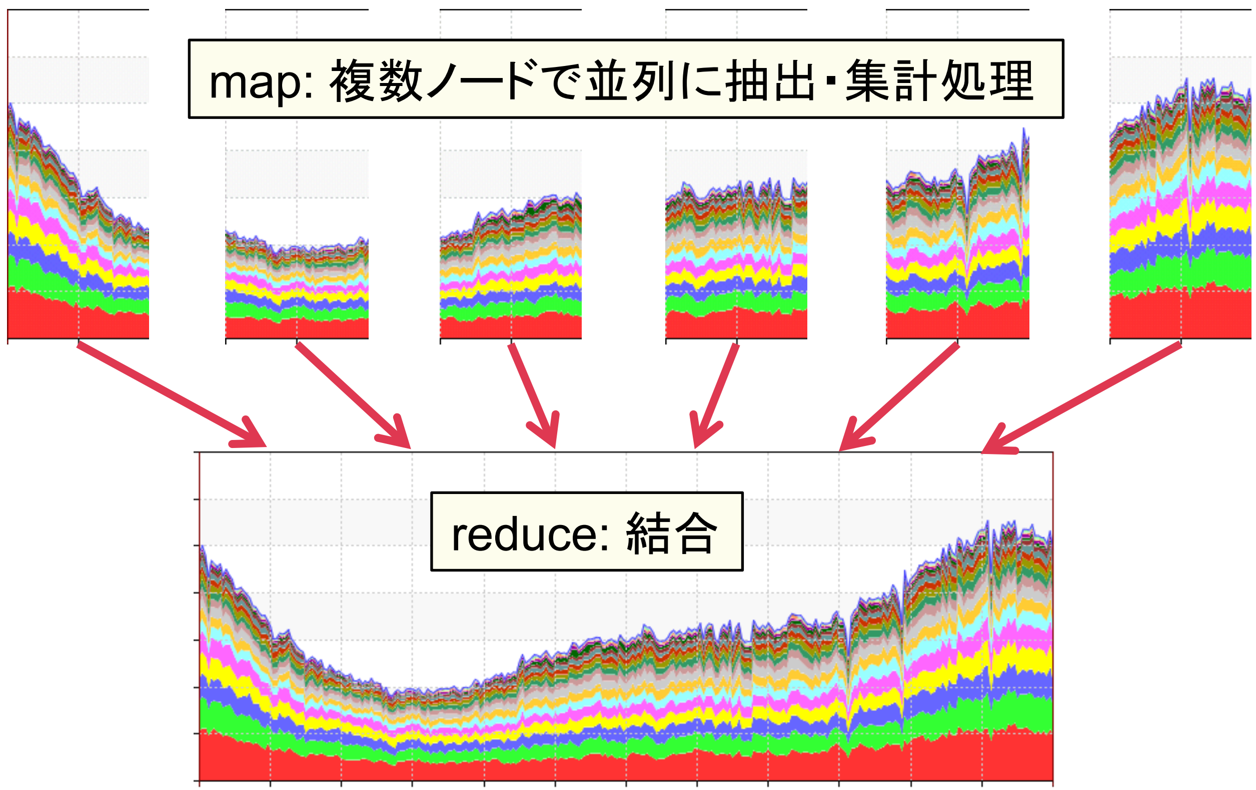
r02/2010/00001

time	router	bytes	pkts	proto	src AS	dst AS	src port	dst port	src ip	dst ip	...
1288253400	172.16.200.2	1433	1	6	2497	9999	80	4321	192.168.0.1	192.168.9.8	
1288253400	172.16.200.2	987	1	6	1234	1000	25	2224	10.168.20.1	10.168.20.1	
...											

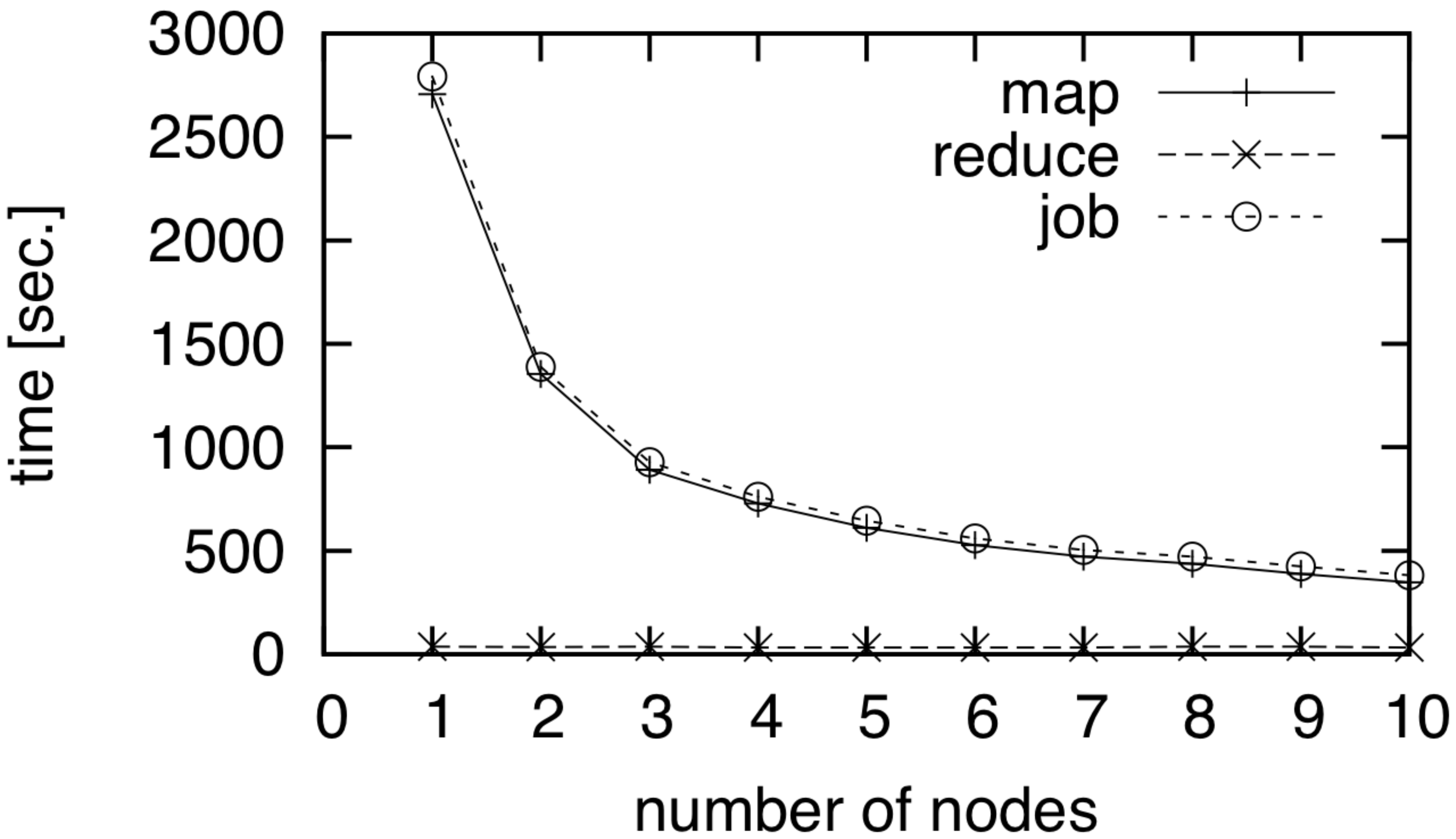


MapReduceのグラフ描画への応用

- グラフの断片を複数ノードで並列抽出・集計



スケーラブル



電子情報通信学会論文 Vol.J93-D,No.7,pp.1072-1081,Jul. 2010.
「大規模データ処理のための分散システムの実装とその応用」より

小さなデータを素通しするMapReduceの速度

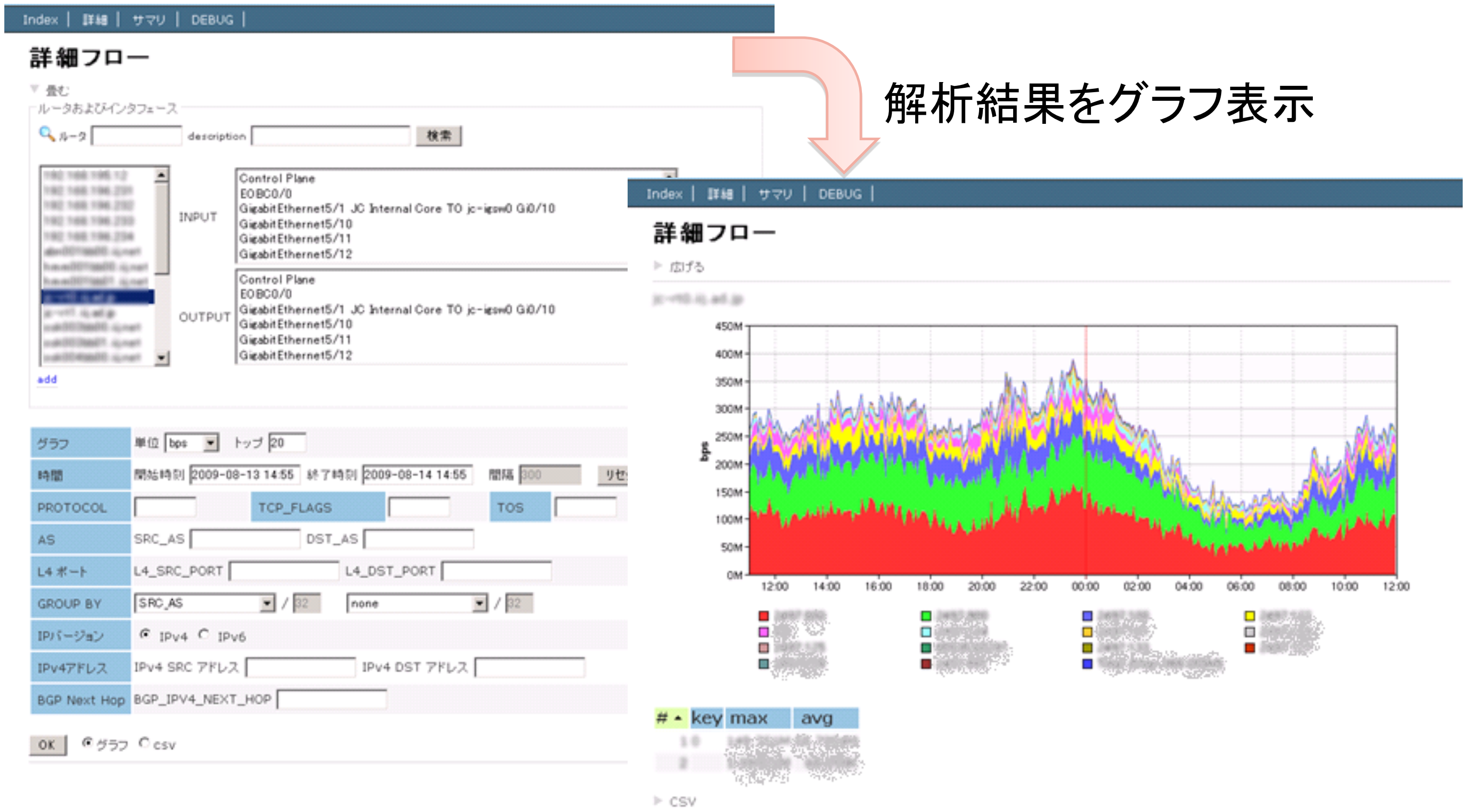
- **Hadoop (0.21) と ddd の比較**
  – 同一データ、同一MapReduceプログラム
  – Hadoop streaming 使用(標準入出力でデータをやりとり)

node数	Hadoop	ddd
1	20秒	0.12秒

- **The fastest MR job takes 24 seconds**
  – (Todd Lipcon, Cloudera)

トラフィック解析システムの画面例

- 多様なパラメータで指定されたグラフを描画



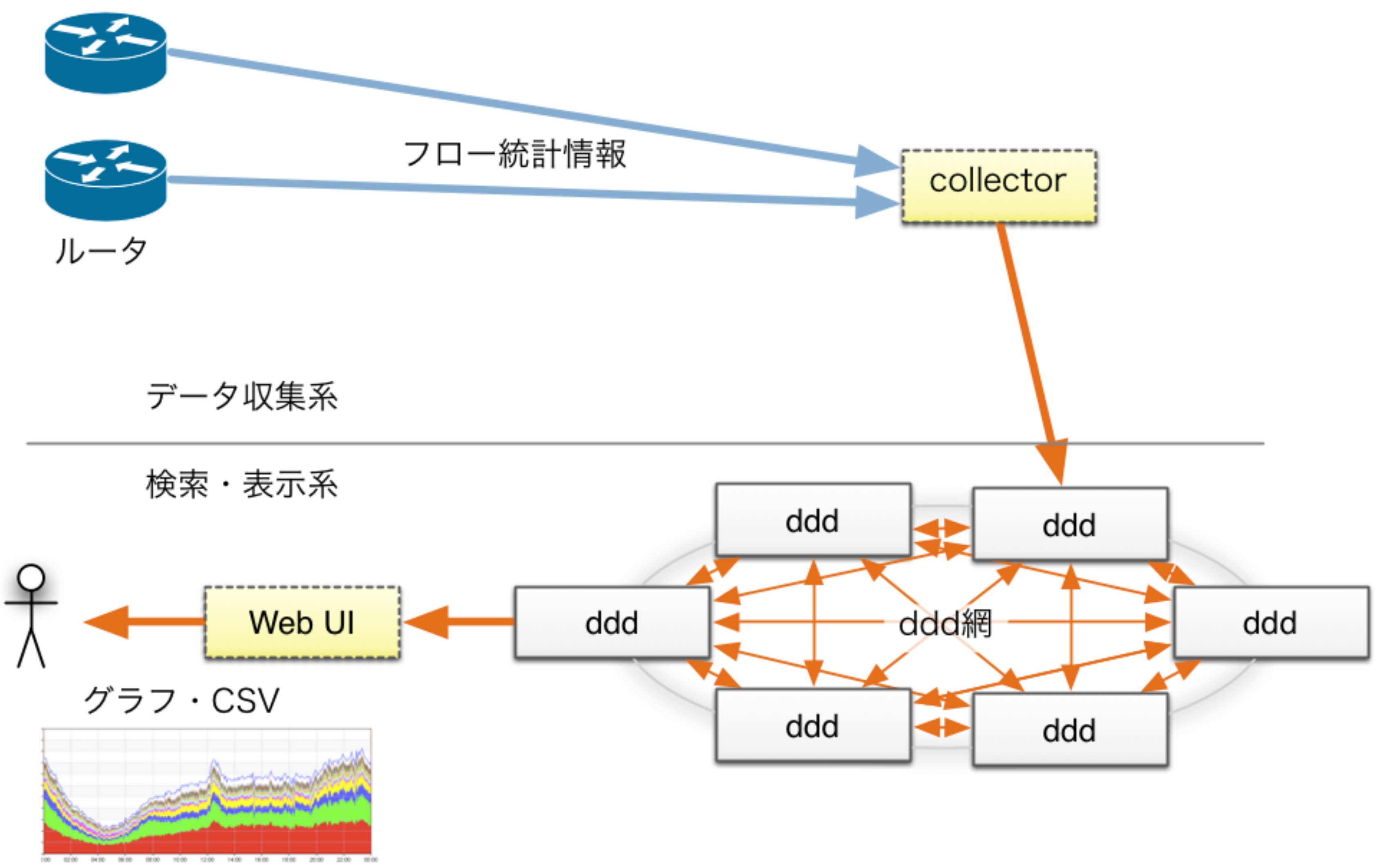
運用

- ハードウェア(Cloudera推奨 2010/03)
  - 4 x 1TB ハードディスク
  - 2 x quad core CPU (2～2.5GHz)
  - 16～24GB RAM
  - Gigabit Ethernet

出典:「Cloudera’s Support Team Shares Some Basic Hardware Recommendations」
http://www.cloudera.com/blog/2010/03/clouderas-support-team-shares-some-basic-hardware-recommendations/

  - RAIDなどのハードウェアによる冗長化機能は無し

トラフィック解析システム全体構成図



多数のコンピュータノードの管理

- **起動**
  - ネットワークブート、共通ブートイメージ
  - 起動時自動セットアップ
  - HDDは搭載しているがシステム(OS)領域はない
- **オペレーション**
  - 複数ノード同時オペレーション(Capistrano等)
- **モニタリング**
  - 死活監視、容量監視、性能監視、etc
- **IPアドレス**
  - IPv6, IPv4 両対応

障害対応

- **基本的に、故障しても放置**
  - なにかのついでのときに交換
  - 運用コスト削減

- **例: Microsoft**
  - 5000 machines / 1 administrator

将来

- 実験中: モジュール型データセンタ

まとめ

- **Big Data を扱う手段としての分散システム**
  - 適材適所
- **ddd**
  - ニーズに合わせて独自開発
  - 用途: トラフィック解析、ログ処理
- **運用**
  - 手間をかけない運用

IIJ Internet Initiative Japan

ご清聴ありがとうございました

お問い合わせ先　IIJインフォメーションセンター
TEL：03-5205-4466（9：30～17：30 土/日/祝日除く）
info@iij.ad.jp
http://www.iij.ad.jp/

Ongoing Innovation