The Stratosphere platform for big data analytics

Michelle Cone, Andrew Miller, Ethan Anthony
The Problem

data hoarding + eCommerce = scientific

What problems / use cases arise when analyzing or using Big Data?
Existing Solutions
What is Stratosphere?

Efficient general-purpose data analytics platform

Emphasizes ease of programming

User Defined Functions are first class citizens
Stratosphere

use cases

data warehousing

data cleansing

extraction and integration

graph analysis

statistical analysis
Directed Acyclic Graphs
DAG  Vs  MapReduce
Similar Systems

- **Hadoop**
  - Map, Shuffle & Reduce steps = high latency
  - Fault tolerance achieved through redundancy

- **Spark**
  - Uses DAGs to achieve low latency
  - DAGs make fault tolerance easy
Stratosphere Features

- **Declarative** high level languages
- "in situ" data analysis
- **Richer** set of primitives
- UDFs are 1\textsuperscript{st} Class citizens
- **Query optimizer**
- Support for iterative programs
- **inter-operator** & **intra-operator** parallelism
Stratosphere Stack

more declarative

more expressive
Meteor Script

```javascript
using ie;
using cleansing;

$articles = read from 'news.json';
$articles = annotate sentences $articles
use algorithm 'morphAdorner';
$articles = annotate entities $articles
use algorithm 'regex' and type 'person';

$peopleInNews = pivot $articles around $person = $article.annotations[*].
            entity
            into {
                name: $person,
                articles: $articles
            };

$persons = read from 'person.json';
$persons = remove duplicates
where average(levenshtein(name),
dateSim(birthDay)) > 0.95
retain longest(name);

$personsInNews = join $refPerson in $persons,
$newsPerson in $peopleInNews
where $refPerson.name == $newsPerson.name
            into {
                $refPerson.*,
                articles: $newsPerson.articles[*].url
            };

write $personsInNews to 'result.json';
```

- **lines 1 – 4:** Information Extraction
- **lines 4 – 14:** Annotation of relevant info
- **lines 9 – 14:** Articles grouped by person names
- **lines 16 – 20:** Duplicate operator removes duplicates
- **lines 21 - 27:** dataset joined on person name
Second Order Functions
(PACTS)

Map
Reduce
Cross
Match
CoGroup

not included in MapReduce
MapReduce and PACT - Comparing Data Parallel Programming Models, Alexandrov et al 2011
How do the challenges of optimizing a PACT program compare to challenges of a relational optimizer?
Stratosphere Optimizer

Original PACT program

Modified PACT program

Physical Plan

Nephele Job Graph
Nephele Process Model

Job Manager

Heartbeats / Task Status / Profiling Data

Push Tasks

Job Graph

Execution Graph
Higher Order Fixpoint Operator

(a) Bulk Iteration

(b) Incremental Iteration
TeraSort Job

Measured efficiency of the execution engines
Hadoop and Stratosphere
Wordcount Job

Compared performance

Hadoop and Stratosphere
Modified TPC-H Benchmark

![Graph showing execution time vs. DOP for Stratosphere PACT and Hive.]

Measured impact of cost-based optimization for **Stratosphere** and **Hive**.
Triangle Enumerations

Pipelined execution strategy versus Hadoop MapReduce (2 MapReduce jobs)

Stratosphere PACT and Hadoop MapReduce
Connected Components Algorithm

Observe execution times

**Giraph** and **Stratosphere**
Next Steps?