

Performance study of Spindle, a web analytics query engine implemented in Spark CloudCom 2014

Brandon Amos* and David Tompkins
Adobe Research

*Adobe intern, Ph.D. Student at Carnegie Mellon University.

December 19, 2014

Motivation

Spindle
Architecture

Overview.
Features.
Queries.

Empirical Results

Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking concurrent queries.

Scaling Spark and HDFS workers.

Future Work

Conclusions

Motivation

Spindle
Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking
concurrent
queries.

Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking concurrent queries.

Scaling Spark and HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking
concurrent
queries.

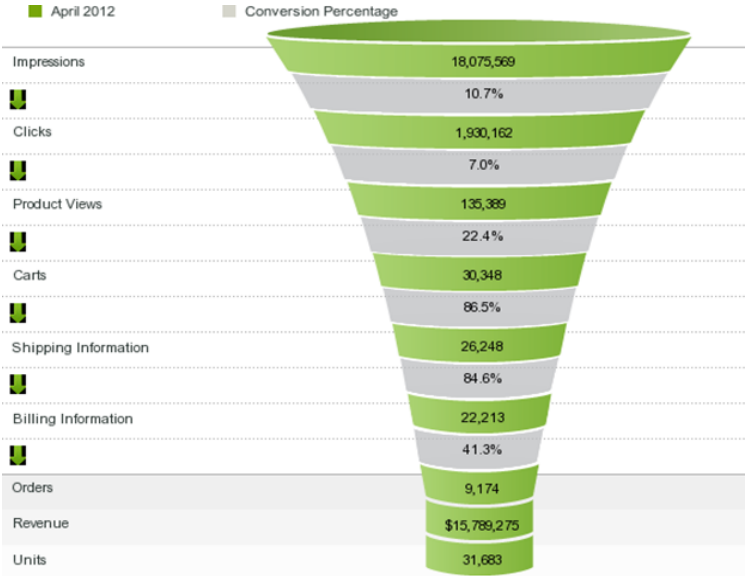
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

- Adobe Marketing Cloud offers web analytics.



Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

- ▶ Adobe Marketing Cloud offers web analytics for interactive data exploration.
- ▶ Terabytes of data, thousands of servers.
- ▶ Trending general-purpose distributed data processing engines.
 - ▶ Apache Spark
 - ▶ Queries implemented with map and reduce functions.
 - ▶ In-memory caching.
 - ▶ Cloudera Impala
 - ▶ Analytic Database for Apache Hadoop.
 - ▶ Google Dremel
 - ▶ Analytics of web-scale datasets.
- ▶ We present **Spindle**, which is an early investigation of the feasibility of Apache Spark for web analytics
- ▶ Goal: Low-latency query execution time.

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking concurrent queries.

Scaling Spark and HDFS workers.

Future Work

Conclusions

Motivation

**Spindle
Architecture**

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking
concurrent
queries.

Scaling Spark and
HDFS workers.

Future Work

Conclusions

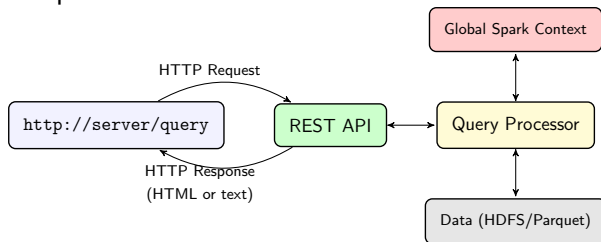
Spindle Architecture

Overview.

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

What is Spindle?



Motivation

Spindle
Architecture

Overview.
Features.
Queries.

Empirical Results

Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Spindle Architecture

Features.

- ▶ Data format challenges:
 - ▶ Operates on archival data with 250 columns.
 - ▶ Data is sparse and queries use <10 columns at a time.
- ▶ Use columnar data format on distributed filesystem.
- ▶ Spindle makes tuning parameters easy.
 - ▶ Intermediate data partitioning
 - ▶ Caching

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Spindle Architecture

Queries.

- Experimental setup: Representative set of analytics queries.

| Shorthand | Name |
|-----------|--|
| Q0 | Pageviews |
| Q1 | Revenue |
| Q2 | RevenueFromTopReferringDomains |
| Q3 | RevenueFromTopReferringDomainsFirstVisitGoogle |
| Q4 | TopPages |
| Q5 | TopPagesByBrowser |
| Q6 | TopPagesByPreviousTopPages |
| Q7 | TopReferringDomains |

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results

Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Spindle Architecture

Queries.

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

- Queries use a small columnar subset.

| | | Q0 | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
|---------|--------------------|----|----|----|----|----|----|----|----|
| Columns | post_pagename | × | | | | × | × | × | |
| | user_agent | | | | | | × | | |
| | visit_referrer | | | × | × | | | | |
| | post_visid_high | | | × | × | | | × | × |
| | post_visid_low | | | × | × | | | × | × |
| | visit_num | | | × | × | | | × | × |
| | visit_referrer | | | | | | | | × |
| | hit_time_gmt | | | | | | | × | |
| | post_purchaseid | | × | × | × | | | | |
| | post_product_list | | × | × | × | | | | |
| | first_hit_referrer | | | | × | | | | |

Motivation

Spindle
Architecture

Overview.
Features.
Queries.

Empirical Results

Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking concurrent queries.

Scaling Spark and HDFS workers.

Future Work

Conclusions

Motivation

Spindle
Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking
concurrent
queries.

Scaling Spark and
HDFS workers.

Future Work

Conclusions

Empirical Results

Caching.

- ▶ Six cluster nodes (32 GB memory each), Spark and HDFS on each.
- ▶ 13.1GB of data, 1 week, 1 customer.
- ▶ **Question:** How does caching in-memory improve performance?

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.
Features.
Queries.

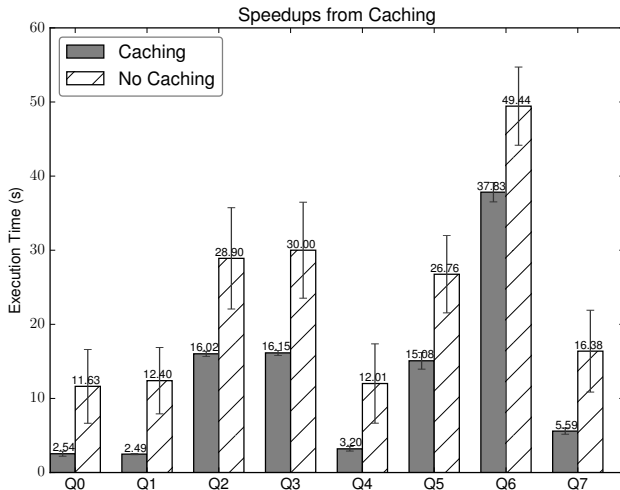
Empirical Results

Caching.

Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions



- Caching helps, but what else can be done to lower query execution times?

Empirical Results

Data partitioning.

- ▶ Partitions are groups of data executed in a batch.
- ▶ Partitions can be executed concurrently.
- ▶ Not clear how to partition the intermediate data.
 - ▶ Too small: Partition management overhead.
 - ▶ Too large: Data is processed in serial.

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.
Features.
Queries.

Empirical Results

Caching.

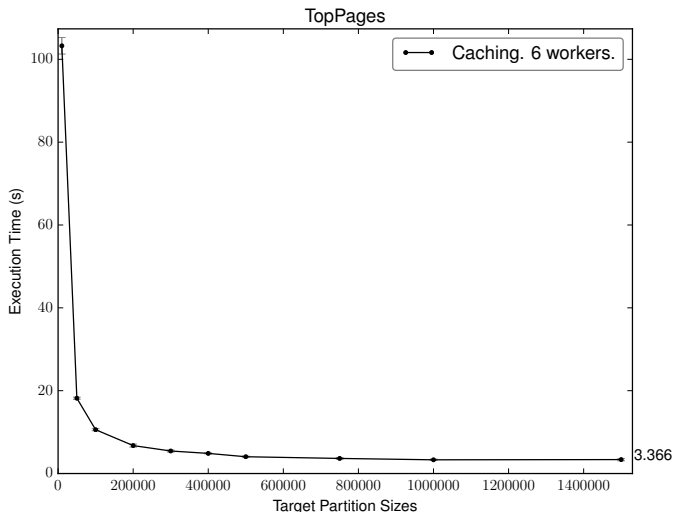
Data partitioning.

Benchmarking
concurrent
queries.

Scaling Spark and
HDFS workers.

Future Work

Conclusions



- Targeting 1.5M items in each partition is reasonable.

Empirical Results

Benchmarking concurrent queries.

- ▶ How much will Spindle's performance degrade if multiple users are utilizing it at the same time?
- ▶ Concurrently call the same query on the same data.
- ▶ Average execution times.

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
**Benchmarking
concurrent
queries.**
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

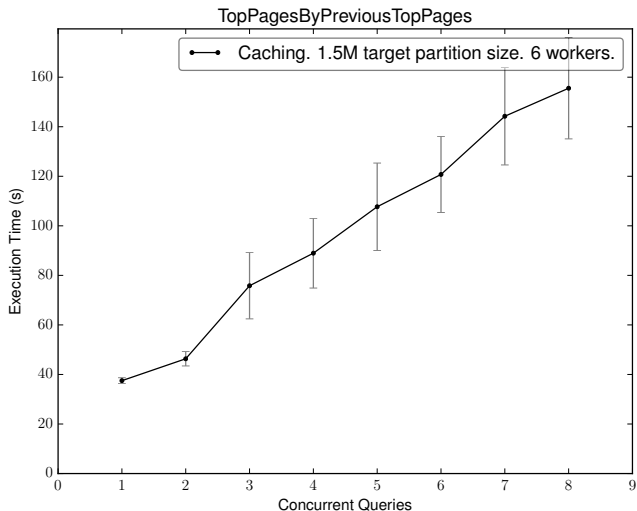
Overview.
Features.
Queries.

Empirical Results

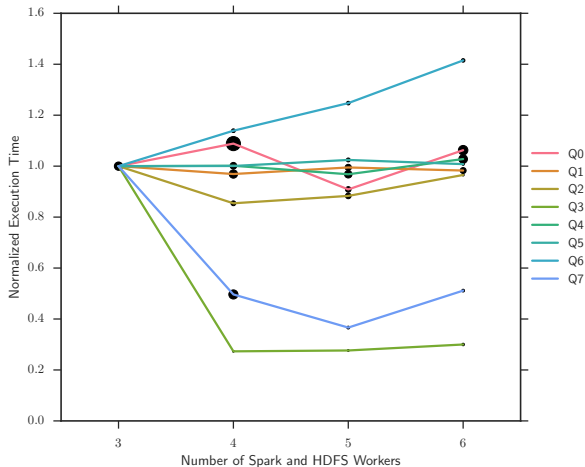
Caching.
Data partitioning.
**Benchmarking
concurrent
queries.**
Scaling Spark and
HDFS workers.

Future Work

Conclusions



- Performance better than serializing concurrent requests, but can be improved.



- Further profiling is needed to improve performance as increasing the number of workers.

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking concurrent queries.

Scaling Spark and HDFS workers.

Future Work

Conclusions

Motivation

Spindle
Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking
concurrent
queries.

Scaling Spark and
HDFS workers.

Future Work

Conclusions

- ▶ Lowering query execution time.
 - ▶ Goal: Sub-second.
- ▶ Automatically tuning parameter exploration space for a given workload.
 - ▶ Online/Dynamically
 - ▶ Offline
- ▶ Results caching for exact same queries.
- ▶ Data preprocessing to remove redundant computations.

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions

Motivation

Spindle Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking concurrent queries.

Scaling Spark and HDFS workers.

Future Work

Conclusions

Motivation

Spindle
Architecture

Overview.

Features.

Queries.

Empirical Results

Caching.

Data partitioning.

Benchmarking
concurrent
queries.

Scaling Spark and
HDFS workers.

Future Work

Conclusions

Conclusions

Spindle,
CloudCom 2014

Amos and
Tompkins,
Adobe Research

- ▶ We present **Spindle**.
 - ▶ **Open-source** prototype analytics processing engine.
 - ▶ Sample set of web analytics queries.
 - ▶ Interface for parameter tuning.

| | |
|-----------------|---|
| Spindle Project | http://github.com/adobe-research/spindle |
| Demo | http://adobe-research.github.io/spindle/ |
| Brandon Amos | http://github.com/bamos |
| David Tompkins | http://github.com/DavidTompkins |

Motivation

Spindle
Architecture
Overview.
Features.
Queries.

Empirical Results
Caching.
Data partitioning.
Benchmarking
concurrent
queries.
Scaling Spark and
HDFS workers.

Future Work

Conclusions