FLOWPROPHET: Generic and Accurate Traffic Prediction for Data-parallel Cluster Computing

Hao Wang\textsuperscript{1,2}, Li Chen\textsuperscript{2}, Kai Chen\textsuperscript{2}, Ziyang Li\textsuperscript{2,3}, Yiming Zhang\textsuperscript{3}, Haibing Guan\textsuperscript{1}, Zhengwei Qi\textsuperscript{1}, Dongsheng Li\textsuperscript{3}, Yanhui Geng\textsuperscript{4}

\textsuperscript{1} Shanghai Jiao Tong University
\textsuperscript{2} Hong Kong University of Science and Technology
\textsuperscript{3} National University of Defense Technology
\textsuperscript{4} Huawei Technologies Co. Ltd.
Dryad

Apache Hama

Spark

Hadoop

Microsoft
Flow-based optimization mechanisms:
- PDQ [Sigcomm’12], pFabric [Sigcomm’13], PASE [Sigcomm’14], Varys [Sigcomm’14], Baraat [Sigcomm’14]

Architectural bandwidth provisioning:
- c-Through [Sigcomm’10], Helios [Sigcomm’11], Mordia [Sigcomm’13], OSA [NSDI’12]

Traffic engineering:
- Hedera [NSDI’10], MicroTE [CoNEXT’11], D³ [Sigcomm’11]
Knowing the Flow Information Ahead of Time

Flow-based optimization mechanisms:
- PDQ [Sigcomm’12], prabnc [Sigcomm’13], PASE [Sigcomm’14], Varys [Sigcomm’14], Baraat [Sigcomm’14]

Architectural bandwidth provisioning:
- c-Through [Sigcomm’10], Helios [Sigcomm’11], Mordia [Sigcomm’13], OSA [NSDI’12]

Traffic engineering:
- Hedera [NSDI’10], MicroTE [CoNEXT’11], D$^3$ [Sigcomm’11]
FlowProphet

• Generic for DCFs
• Accurate and fined-grained
• Ahead-of-time
• Scalable and low-overhead
Toy Example: Word Count
Logical View

map()

reduce()
Physical View
Physical View

...... map()
Physical View

(map)
Physical View

map()
Physical View

(A,1)
(C,1)
(E,1)

(B,1)
(D,1)
(E,1)

...... map()
Physical View

map()
Physical View

...... map()
Physical View

... map()

Shuffle
Physical View

map()

Shuffle

reduce()
User

Distributed Computing Frameworks
User \rightarrow \text{Logical View} \rightarrow \text{Physical View}

Predict flow info.
Logical View

Predict

Physical View
Logical View

Predict

Flow info.
DAG

Flow info.

Predict
Directed Acyclic Graph (DAG)
Dryad

input data

computing vertices

output files
input data

map tasks

reduce tasks

output data
supserstep(i)

computing nodes

barrier synchronization

(BSP Model)
Task Assignment

Worker#1  Worker#2  Worker#n
LIFE CYCLE
Observation

— DAG contains necessary time, data, and flow dependencies for accurate flow prediction.
## API EXAMPLES

- **Required APIs for DCF master**

<table>
<thead>
<tr>
<th>Event Definition</th>
<th>Trigger Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>newStageEvent(stageID, childStageID)</code></td>
<td>a new stage is created</td>
</tr>
<tr>
<td><code>stageStartEvent(List[task], stageID)</code></td>
<td>a stage is beginning</td>
</tr>
<tr>
<td><code>stageFinishedEvent(stageID)</code></td>
<td>a stage is finished</td>
</tr>
</tbody>
</table>

- **The DAG Builder event handlers**

<table>
<thead>
<tr>
<th>Event Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>newStageHandler(newStageEvent)</code></td>
</tr>
<tr>
<td>⇒ (currentStage, childStage)</td>
</tr>
<tr>
<td><code>stageStartHandler(stageStartEvent)</code></td>
</tr>
<tr>
<td>⇒ Event(List[task], List[stageID])</td>
</tr>
<tr>
<td><code>stageFinishedHandler(stageFinishedEvent)</code></td>
</tr>
<tr>
<td>⇒ Event(stageID)</td>
</tr>
</tbody>
</table>
FlowProphet

- Generic
- Accurate and fine-grained
- Ahead-of-time
- Scalable and low-overhead
TESTBED

- Dell PowerEdge R320 x 37
- Intel Xeons E5-1410 2.8GHz CPU
- 24GB 1600MHz DDR3
- Broadcom Gigabit Ethernet NIC
- Pronto-3295 Gigabit Ethernet Switch
BENCHMARKS

- WikiPageRank
- SparkPageRank
- Spark K-means
- Hadoop TeraSort
- π (Pi)
- WordCount

METRICS

- Time advance
- Prediction accuracy
- Overhead
- Scalability
- Benefits
TIME ADVANCE

- WikipediaPageRank-13G (Spark)
CDF OF LEAD TIME

- **Spark WikiPR-13G**: Avg: 414.1ms
- **Spark WikiPR-26G**: Avg: 478ms
- **Hadoop TeraSort-10G**: Avg: 12.3123s
- **Hadoop WordCount-20G**: Avg: 7.7348s
**Prediction Accuracy**

![Graphs showing actual traffic vs predicted traffic for different scenarios.](image)

- **Spark WikiPR-26G**
  - Actual Traffic
  - Predicted Traffic

- **Hadoop TeraSort-10G**
  - Actual Traffic
  - Predicted Traffic

- **Hadoop WordCount-10G**
  - Actual Traffic
  - Predicted Traffic

The graphs above illustrate the accuracy of traffic prediction for different computational tasks. The y-axis represents the volume of traffic in MB or GB, while the x-axis lists the task IDs.
OVERHEAD

![Completion Time (s)](chart)

- **Wikipedia PageRank**:
  - 13G
  - 26G
- **SparkPi**:
  - 500M
  - 1000M
- **WordCount**:
  - 20G
  - 40G
- **KMeans**
  - 20G

- **Pure Spark**
- **Spark with FlowProphet**
OVERHEAD

Completion Time (s)

- Pure Hadoop
- Hadoop with FlowProphet
- Hadoop with HadoopWatch

HadoopPi
-100M
-500M

WordCount
-20G
-40G

TeraSort-10G
TeraSort-20G
SCALABILITY

- Overhead Ratio (OR): \[ OR = \frac{t_{enabled} - t_{disabled}}{t_{disabled}} \]

![Graph showing Overhead Ratio vs Number of Worker Nodes](image_url)
SCALABILITY

Hadoop TeraSort-10G

- Pure Hadoop
- Hadoop with FlowProphet
- OR on testbed
- OR by projection

Number of Worker Nodes

Job Completion Time (s)

Overhead Ratio (%)
**Benefits**

- Hadoop TeraSort-25G
- 12.52% JCT reduction by a simple network scheduler

![Graph showing average coflow and job completion times for Original and Optimized scenarios.]
Related Work

- Analyze past statistics
  - Traffic Engineering with Estimated Traffic Matrices
- Monitor buffers or counters in switches
  - c-Through, Hedera, Helios
- Tracing and profiling toolkits
  - X-Trace
- File system monitoring
  - HadoopWatch
SUMMARY

• DCF execution pattern
• DAG for predicting flows
• Design and implementation
• Evaluation on testbed
Thank you | Q&A