Pre-production and Debugging Tools
for Timely Dataflow

CS 848: Models and Applications of Distributed Data Systems
Mon, Dec 5th 2016

Amine Mhedhbi & Saifuddin Hitawala
Distributed Data Processing Systems in 2006
Distributed Data Processing Systems in 2016

- S4
- Spark
- Storm
- Mahout
- MongoDB
- GraphX
- Accumulo
- Cassandra
- FlockDB
- CouchDB
- HBase
- MapReduce
- pig
- Flume
- Apache Hama
- Stanford GPS
- Pregel
- Hive
- Dryad
- DryadLINQ
Many topics of interest within these systems.

- Application Optimizations
- Managing Memory
- Resource Scheduling
- Debugging
- Testing
- Programming
- Dealing with Failures
- Data Privacy & Security
- Data Visualization
- Algorithms
- Models of Computation
- Network Optimizations
- Distributed Stream Computing Platform
- Spark
- TEZ
- Storm
- Mahout
- MongoDB
- Giraph
- Apache Hama
- GraphLab
- MapReduce
- Hadoop
- Cassandra
- Accumulo
- FlockDB
- CouchDB
- PIG
- Hive
- SQL
- Dryad
- DryadLINQ
- Stanford GPS
- Pregel
WE PICKED ....
Project Statement

- “Timely Dataflow” is a rewrite of Naiad System in Rust under the MIT License. *Prototype*

- Goal: DEVELOPER-TOOLS
Flash Back of the Past
Background
"OperatesEvent": // Type of the logged obj
{
    "id": int, // unique id.
    "addr": [int, int, int], // address in terms of scope & id.
    "name": String, // operators name in timely dataflow
}
Background

"OperatesEvent":
{
    ...
    "name": "OP1"
}

"OperatesEvent":
{
    ...
    "name": "OP2"
}
"ChannelsEvent":
{
    "id": int, // unique id
    "scope_addr": [int, int], // scope & worker id
    "source": [int, int], // [op_id, scope_id]
    "target": [int, int], // [op_id, scope_id]
}
"MessageEvent":
{
    "is_send": bool,  // push or pull
    "channel": int,   // unique id
    "source": int,    // worker id
    "target": int,    // worker id
    "length": int     // number of typed records
}
Related Work
Related Work: Tensorflow Dashboard & Apache Stats
Features
Features

- Visualize The Computation Topology
FEATUES

- Visualize The Computation Topology
- Report skew between workers
 FEATURES

- Visualize The Computation Topology
- Report skew between workers
- Replay computation step-by-step visually
- Visualize The Computation Topology
- Report skew between workers
- Replay computation step-by-step visually
- Real-Time Machine Monitoring
DEMO TIME(ly)!
Experiments & Evaluation
PINGPONG: Topology
PINGPONG: EXPERIMENTAL RUNS, NUM OF ITERATIONS = 10000

Used Himrod Cluster with machines having 256GB memory
Pingpong: Experimental Runs, num of iterations = [10, 100, 1000, 10000]
BFS: Topology
BFS: Experimental Runs

Time VS Workers

System Time (seconds)

Number of Workers

- w/o logging
- logging as binary
- logging as JSON

- 1 W
- 2 W
- 3 W
- 2 M x 4 W
Web App Back-end Profiling

In Progress:

- Profile server-client response time for the 4 features.
Conclusion
Conclusions

- JSON -> Binary for logging.
Conclusions

- JSON -> Binary for logging.
- Large scale testing is a must.
Conclusions

- Project is a prototype. A lot of needed improvements:
Conclusions

- Project is a prototype. A lot of needed improvements:
Conclusions

- Project is a prototype. A lot of needed improvements:
Conclusions

- Project is a prototype. A lot of needed improvements:
Future Work
Future Work

- Real-Time Computation Monitoring
Future Work

- Real-Time Computation Monitoring
- UI code generation (drag & drop) for small computation
Future Work

- Real-Time Computation Monitoring
- UI code generation (drag & drop) for small computation
- Step-by-step debugging of multiple workers computations?
Resources

- Timely Dataflow (Rust Implementation)
- Frank blog posts:
  - Timely dataflow
  - Differential dataflow
- Naiad Paper
- For slides [2-5]: Class slides by Prof. Semih Salihoglu
FIN.

Thank you! Q&A?!