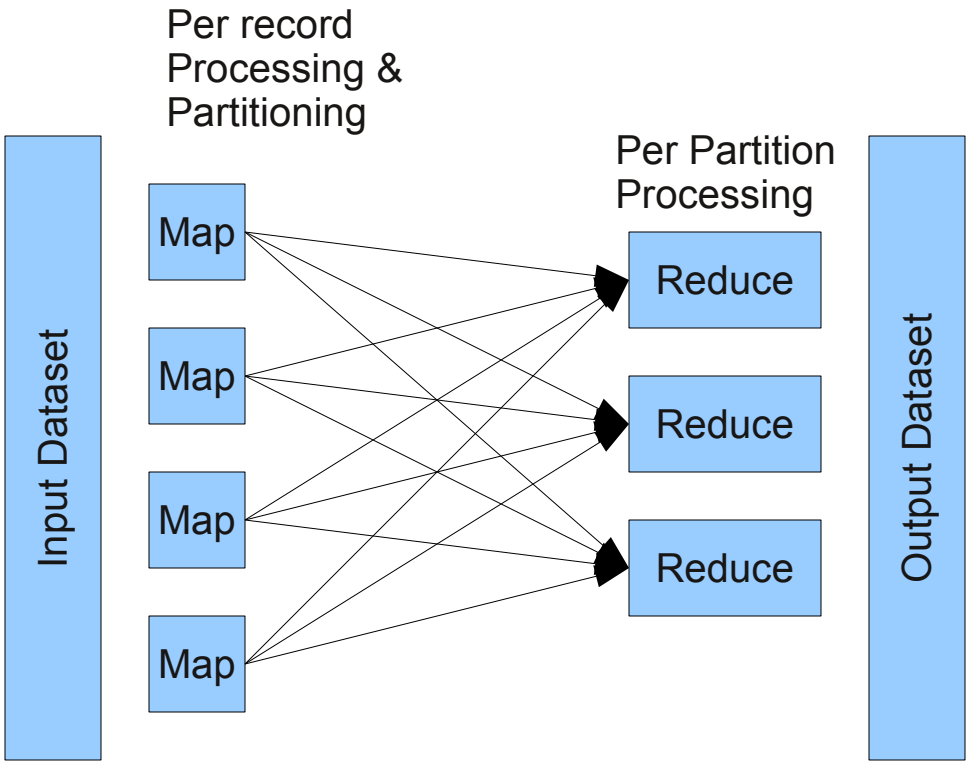
Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows

Christopher Olston and Benjamin Reed
Yahoo! Research

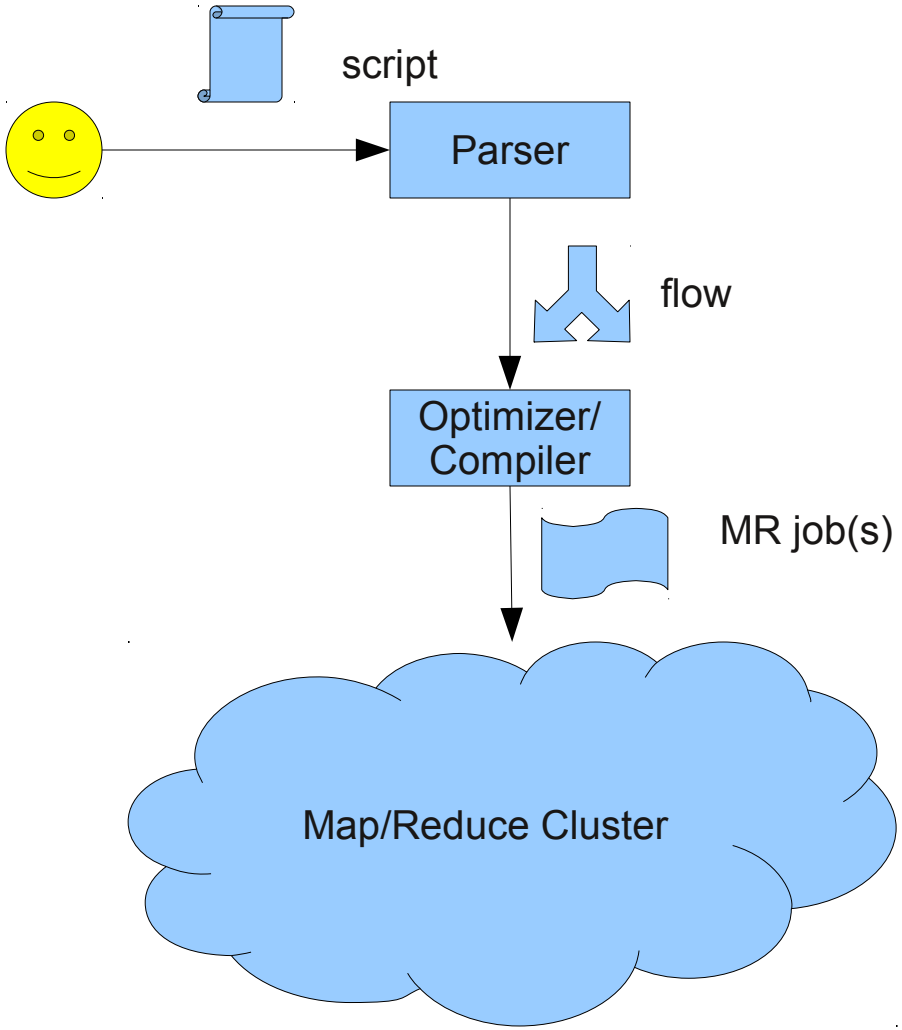
Web Scale problems

- Lots of servers, users, and data
- Fun to have power at your fingertip
- Sucks when things go wrong

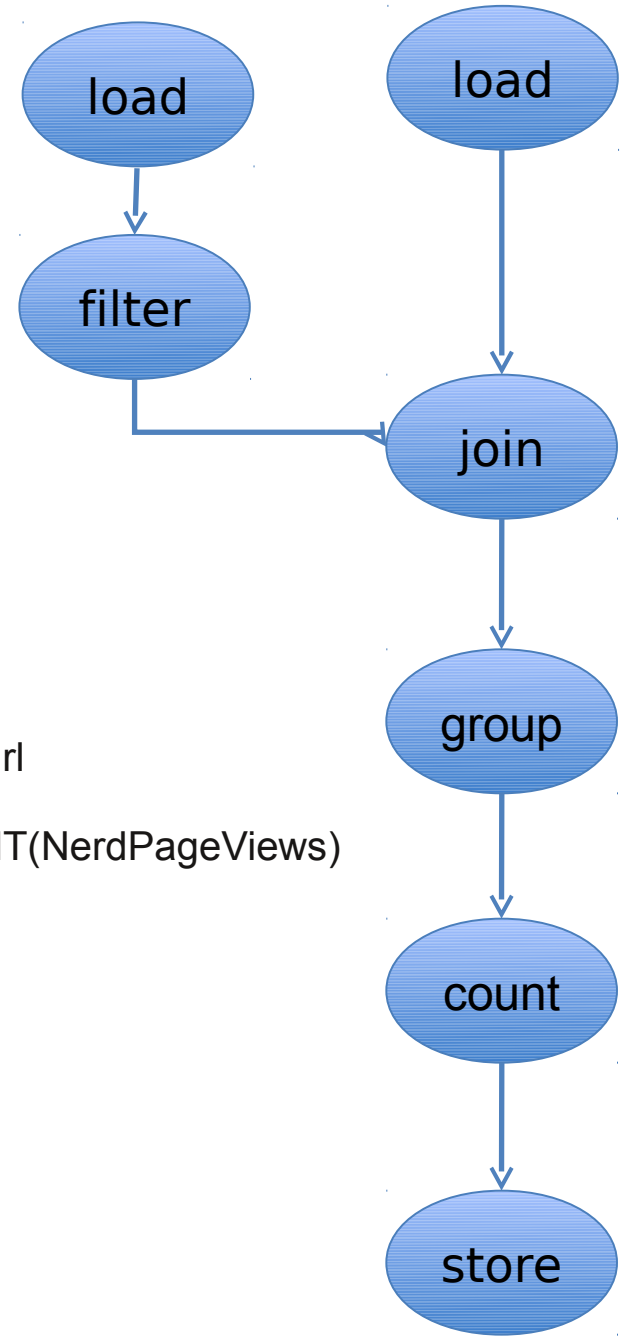
Map/Reduce


Per record
Processing &
Partitioning
Per Partition
Processing
Input Dataset
Map
Map
Map
Map
Reduce
Reduce
Reduce
Output Dataset

Pig on Map/Reduce



Example Pig Workflow




```
Pages = load 'webpages'
UserViews = load 'userclicks'
NerdPages  =filter Pages by NerdFilter(content)
NerdPageViews = join NerdPages, UserViews by url
NerdUsers  = group NerdPageViews by user
Counts = foreach NerdUsers generate user, COUNT(NerdPageViews)
store Counts into 'nerdviewcounts'
```

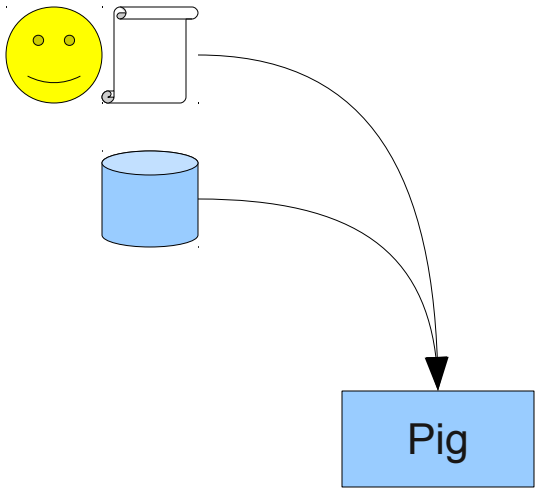
Motivated by User Interviews

Interviewed 10 Yahoo dataflow programmers (mostly Pig users; some users of other dataflow environments)
Asked them how they (wish they could) debug

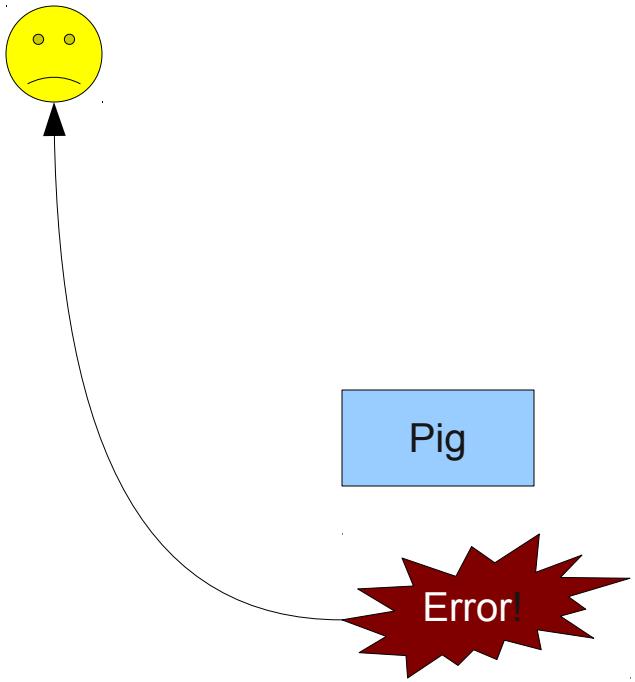
Summary of User Interviews

# of requests	feature
7	**crash culprit determination**
5	**row-level integrity alerts**
4	table-level integrity alerts
4	data samples
3	data summaries
3	memory use monitoring
3	backward tracing (provenance)
2	**forward tracing**
2	golden data/logic testing
2	step-through debugging
2	latency alerts
1	latency profiling
1	overhead profiling
1	trial runs

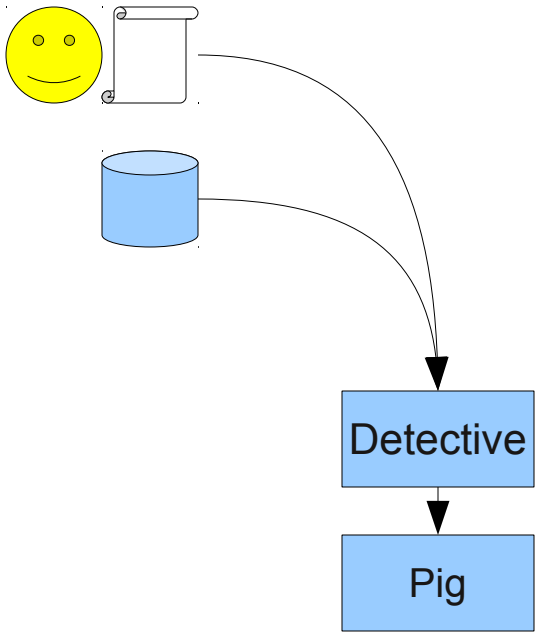
Running Pig



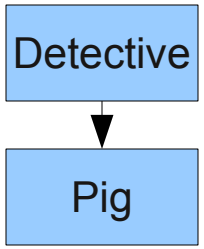
Running Pig



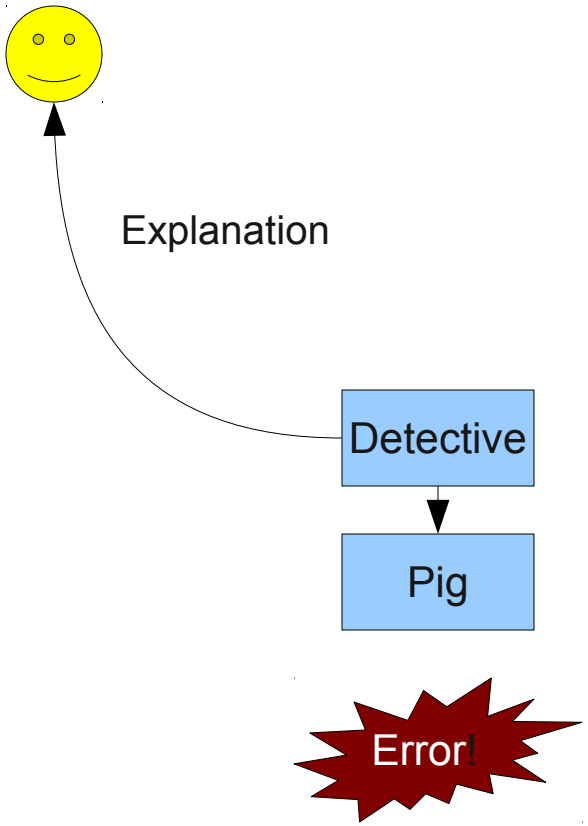
Running Pig



Running Pig



Running Pig



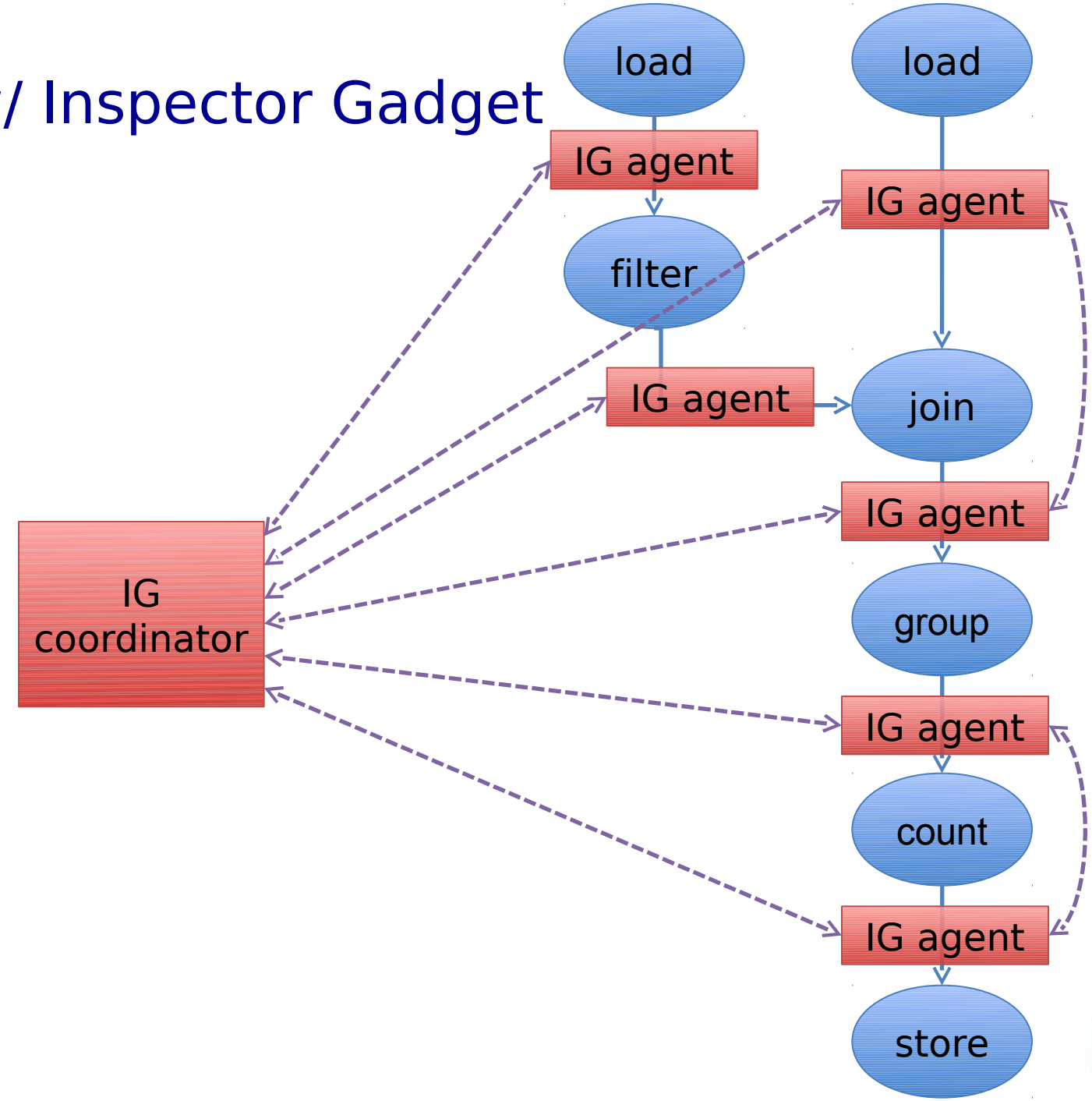
Our Approach

Goal: a programming framework for adding debugging features to Pig

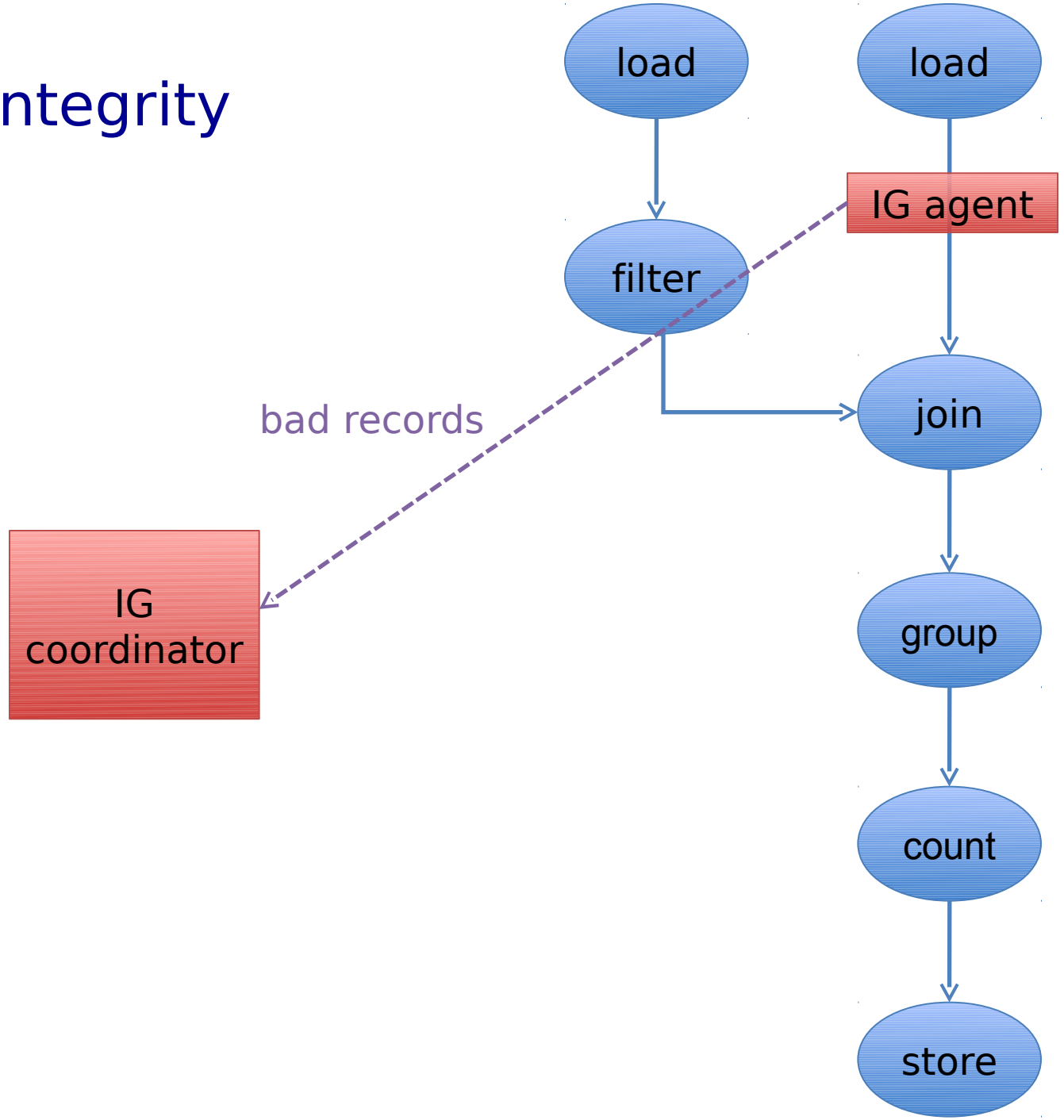
Precept: avoid modifying Pig or tampering with data flowing through Pig

Approach: perform Pig script rewriting – insert special (User Defined Functions) UDFs that look like no-ops to Pig

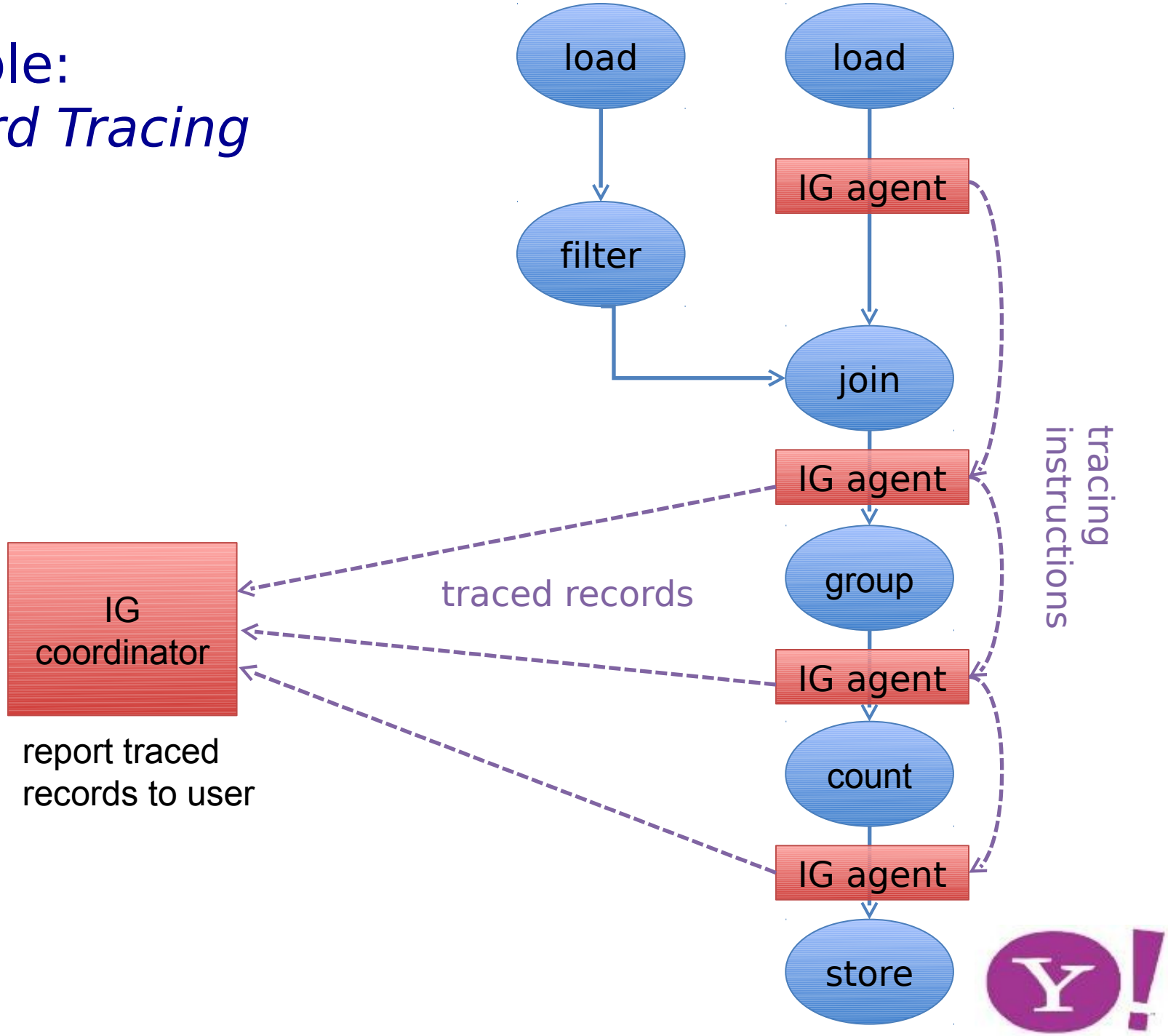
Pig w/ Inspector Gadget



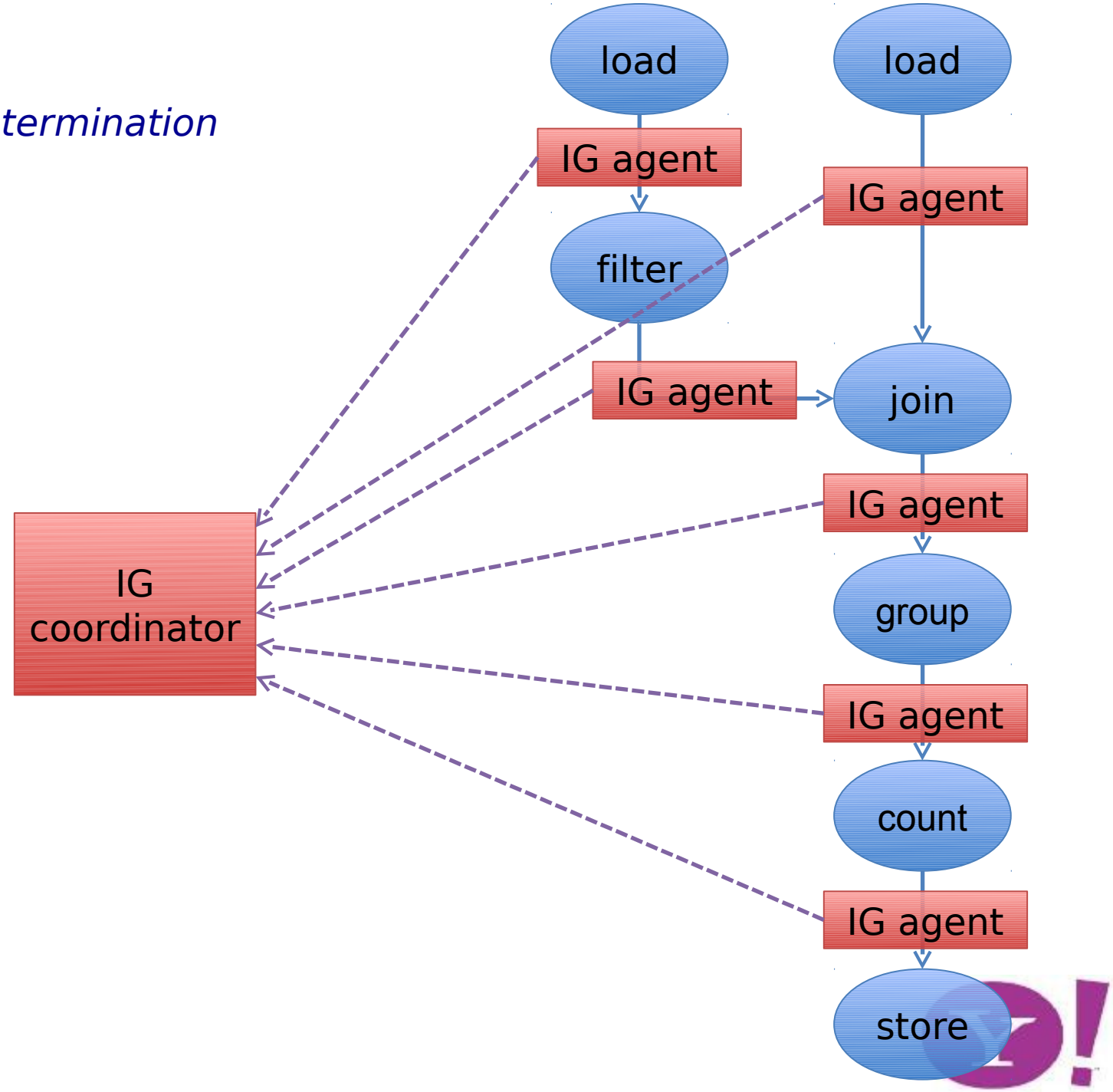
Row Integrity



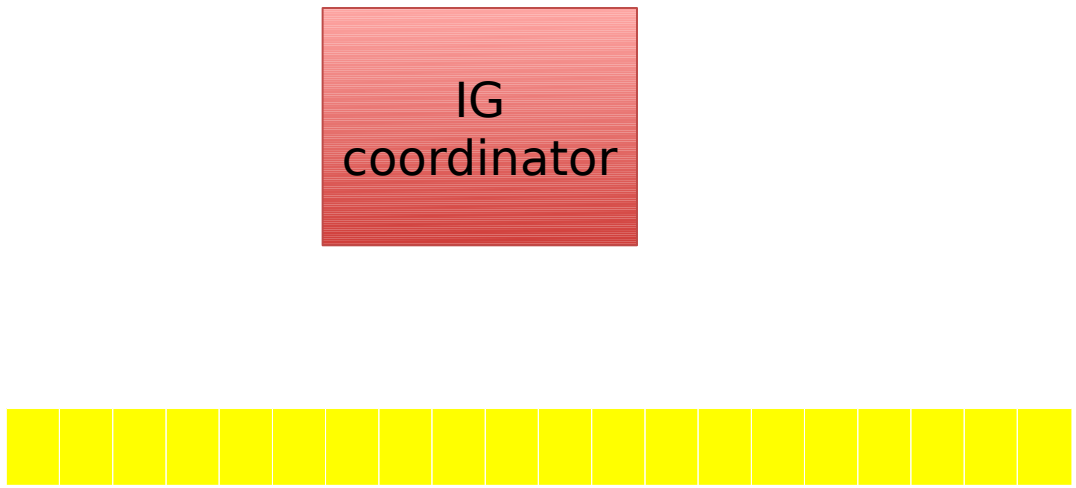
Example:
Forward Tracing



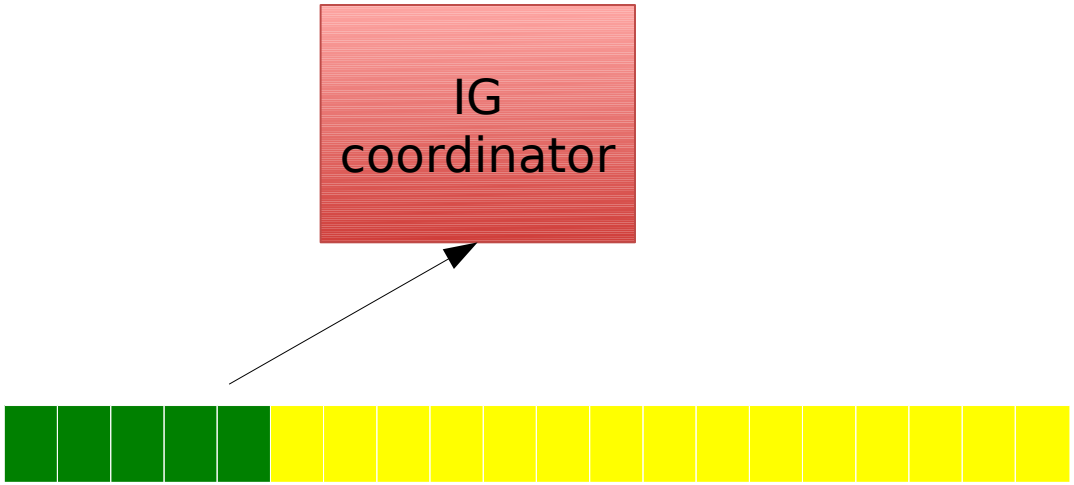
Example:
Crash Culprit Determination



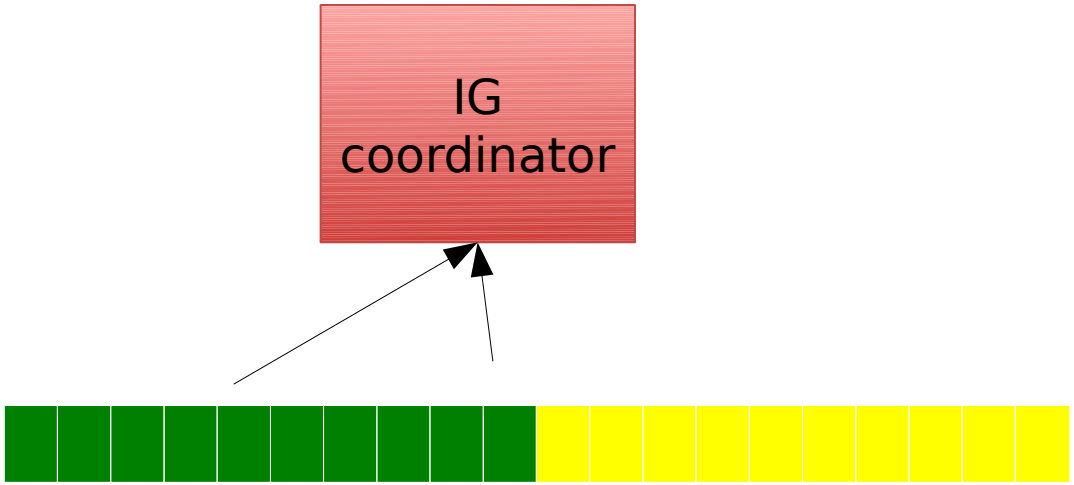
Crash Culprit Sending every 5th



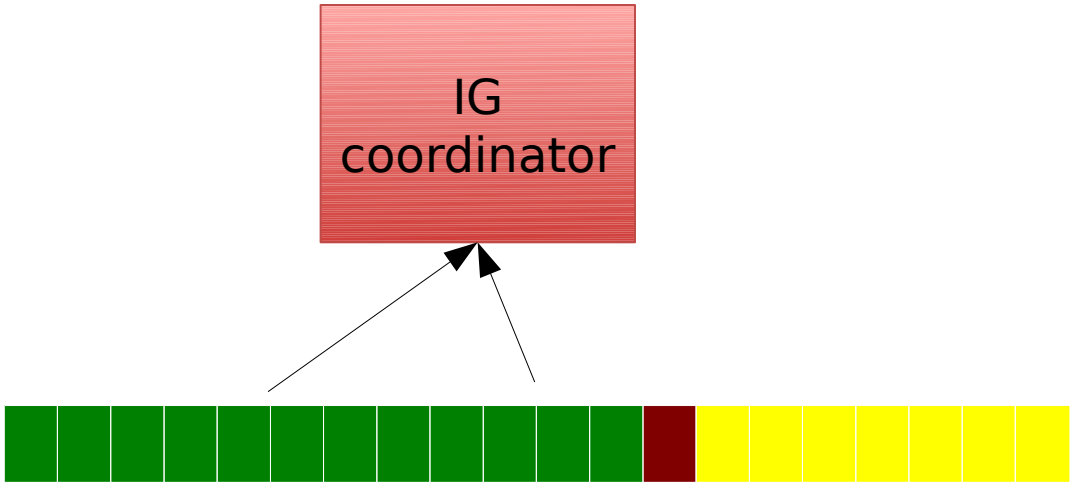
Crash Culprit Sending every 5th



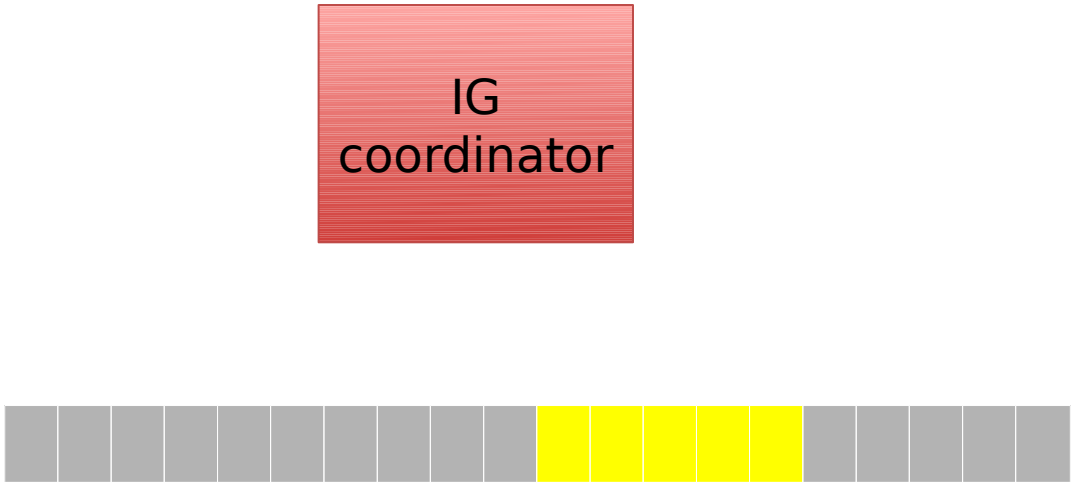
Crash Culprit sending every 5th



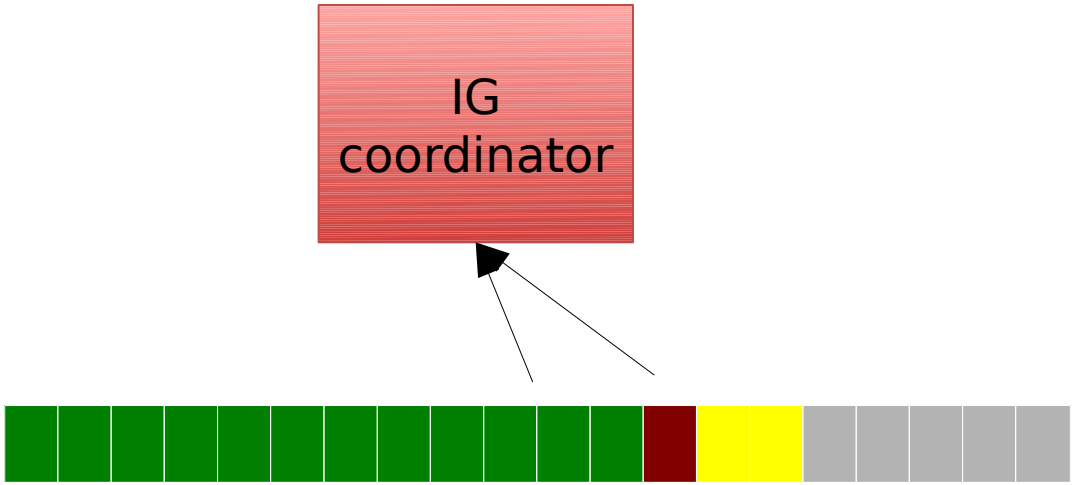
Crash Culprit Sending 5th



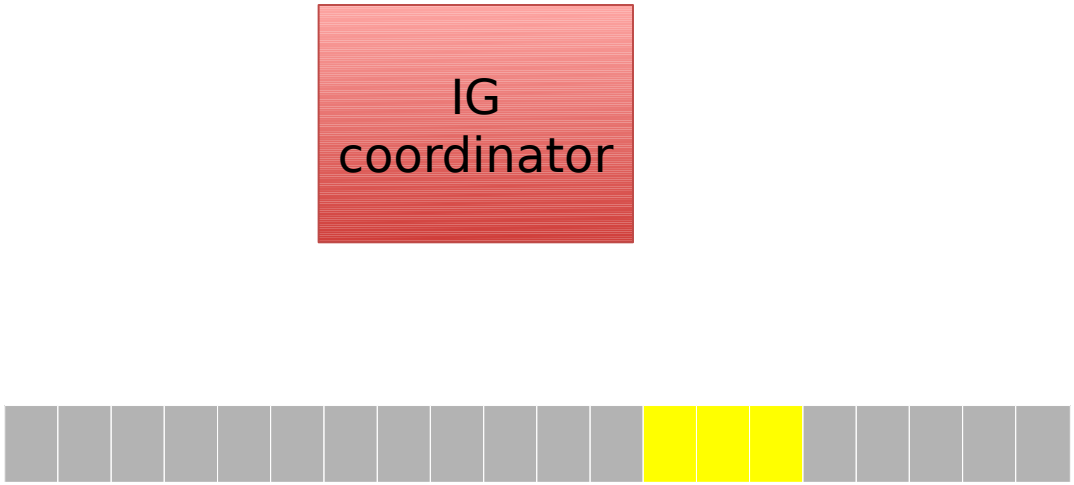
Crash Culprit Sending every 2nd



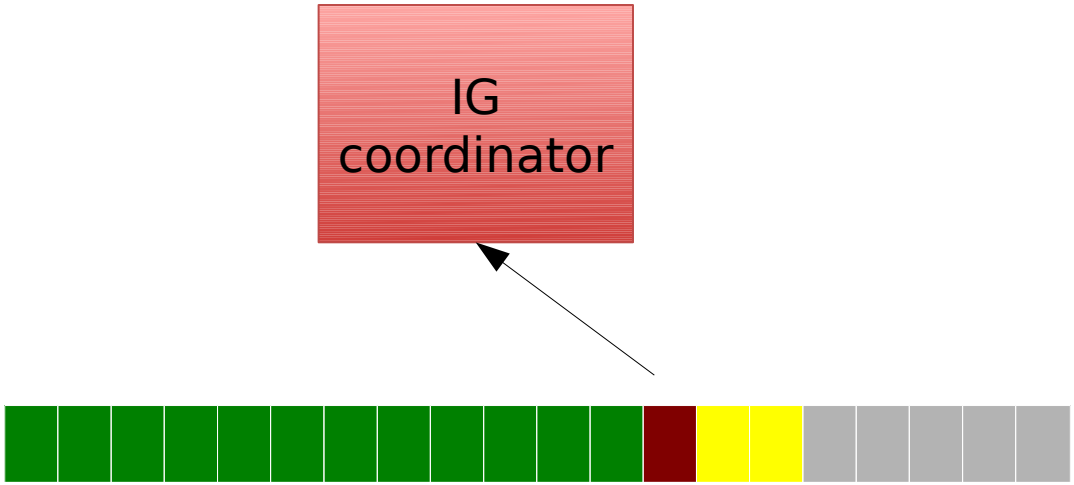
Crash Culprit Sending every 2nd



Crash Culprit Sending every tuple



Crash Culprit Sending every tuple



Agent & Coordinator APIs

Agent Class
init(args)
tags = observeRecord(record, tags)
receiveMessage(source, message)
finish()

Agent Messaging
sendToCoordinator(message)
sendToAgent(agentId, message)
sendDownstream(message)
sendUpstream(message)

Coordinator Class
init(args)
receiveMessage(source, message)
output = finish()

Coordinator Messaging
sendToAgent(agentId, message)

Applications Developed Using IG

# of requests	feature	lines of code (Java)
7	crash culprit determination	141
5	row-level integrity alerts	89
4	table-level integrity alerts	99
4	data samples	97
3	data summaries	130
3	memory use monitoring	N/A
3	backward tracing (provenance)	237
2	forward tracing	114
2	golden data/logic testing	200
2	step-through debugging	N/A
2	latency alerts	168
1	latency profiling	136
1	overhead profiling	124
1	trial runs	93

In Paper

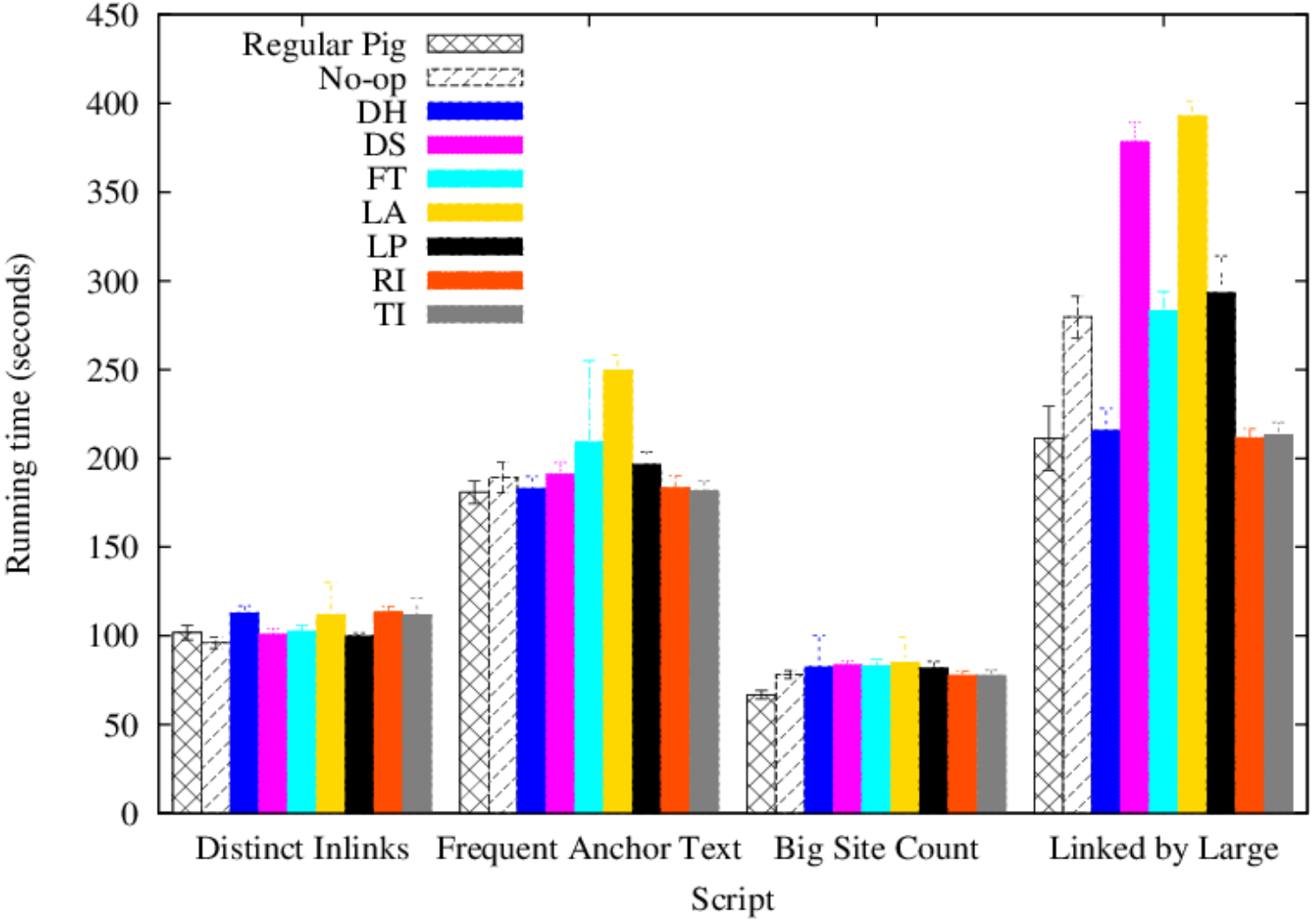
Semantics under parallel/distributed execution
Messaging & tagging implementation
Limitations
Performance experiments
Related work

Performance Experiments

15-machine Pig/Hadoop cluster (1G network)
Four dataflows over a small web crawl sample (10M URLs):

Dataflow Program	Early Projection Optimization ?	Early Aggregation Optimization ?	Number of Map-Reduce Jobs
Distinct Inlinks	N	N	1
Frequent Anchortext	Y	N	1
Big Site Count	Y	Y	1
Linked By Large	N	Y	2

Dataflow Running Times



Related Work

XTrace, etc.
taint tracking
aspect-oriented programming

Summary / Status

- Users have a long wish-list for “debuggability”
  - Make a general framework rather than tool for each
  - Addressed most features with few lines of code
- Rather than implement them as separate features in the Pig core, we built a layer on top
- IG (called Penny) is open source. Accepted into Apache Pig v0.9 release (http://pig.apache.org)

The End