MINNI

Light-weight MapReduce

Athula Balachandran,
Wolf Richter
Erik Zawadzki

GOALS

- Simple MapReduce implementation
- Lower memory footprint than Hadoop
  - Run on a FAWN
- Doesn't sacrifice too much
  - Performance
  - Generality and expressiveness
- Free
- Distributed environments

DESIGN OVERVIEW

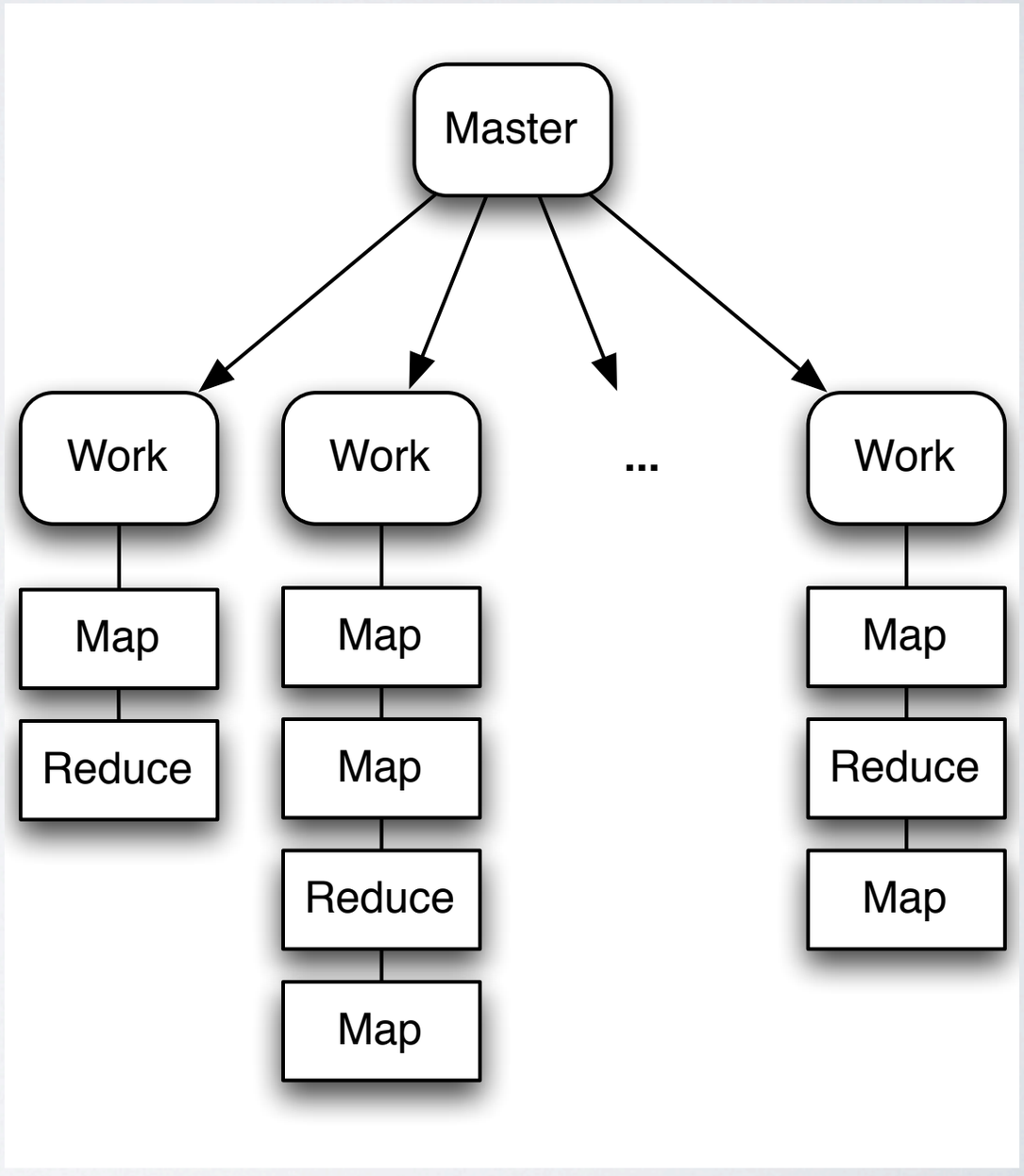
- C++ over Java
- Partial Aggregation Objects
  - Hash, not sort
- Early Reducer Start
- WorkDaemon

COMPONENTS

- Master
  - Controls job scheduling, rerunning, balancing
- WorkDaemon
  - Communicates with Master, transfers data
- Wrapper
  - Manages user code

MASTER

- Scheduling
  - Need to make sure that we don't overwhelm nodes
    - Data close to mapper
- How much control should the master have?
  - Assume that we have the master on a bigger node
- Starts Reducers as soon as some Mappers finish
  - More complicated state
  - Inform the WorkDaemons as data becomes available

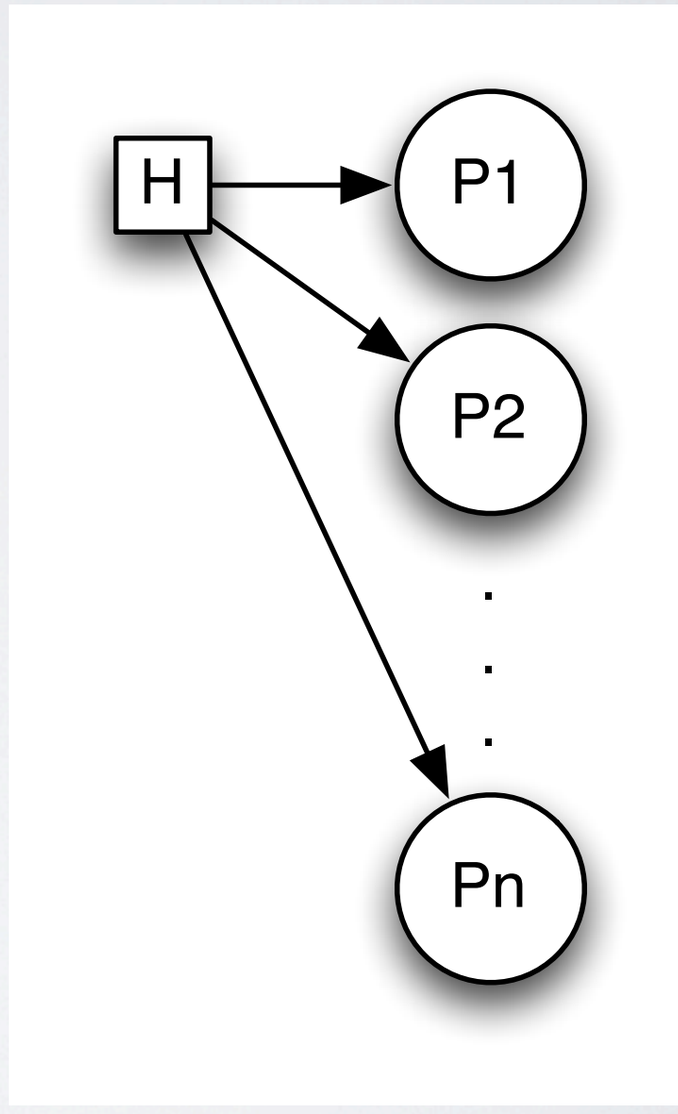


WORKDAEMON

- Master provides high-level instructions, WorkDaemon automates tasks
- Sits on top of a work node
- Mappers can exit as soon as they are done, WorkDaemon handles file transfer
  - Including incremental transfers
- Communicates with Master for Wrappers
  - Batches status reports, only reports abnormal statuses

WRAPPER

- Partial Aggregation Objects (PAO)
  - Think: suspended Reduce jobs
- Hash PAOs rather than sort key-value pairs
- Eliminates spurious notions of ordering
  - need =, not ≤



PARTIAL AGGREGATION OBJECTS

- User defines PAO operations rather than a Reduce function
  - *add* : P x (k,v) → P
  - *merge* : P x P → P
  - serialize/deserialize
- Internal state might be much smaller than the list of values
  - e.g. implementing *max* or *sum*

USER

- Map function
- Partition function
- PAO functions
  - add, merge, serialize/deserialize
    - future: provide support for common STL containers

IMPLEMENTATION

- Off-the-shelf solutions
  - **Concurrency**: Intel's Thread Building Block library
    - Tasks, not threads
    - Tested concurrent containers (no sets, though)
  - **RPC**: Apache's Thrift
    - No direct data transfer
    - Noticed some communication problems during stress tests

IMPLEMENTATION

- First-cut
  - Missing some of the refinements
    - Skipped records, counters, side-effects
- Evaluation in progress
  - Eight nodes running both Hadoop and Minni
    - Performance, memory footprint
    - FAWN test if possible

INVOICE

- Total Physical Source Lines of Code (SLOC)
  - 3,010+
- Development Effort Estimate, Person-Years (Person-Months)
  - 0.64 (7.63)
- Total Estimated Cost to Develop
  - $ 85,928
  - Please settle promptly.

THANK YOU

- Questions?
  - Answers?