

Semantic Characterization of MapReduce Workloads

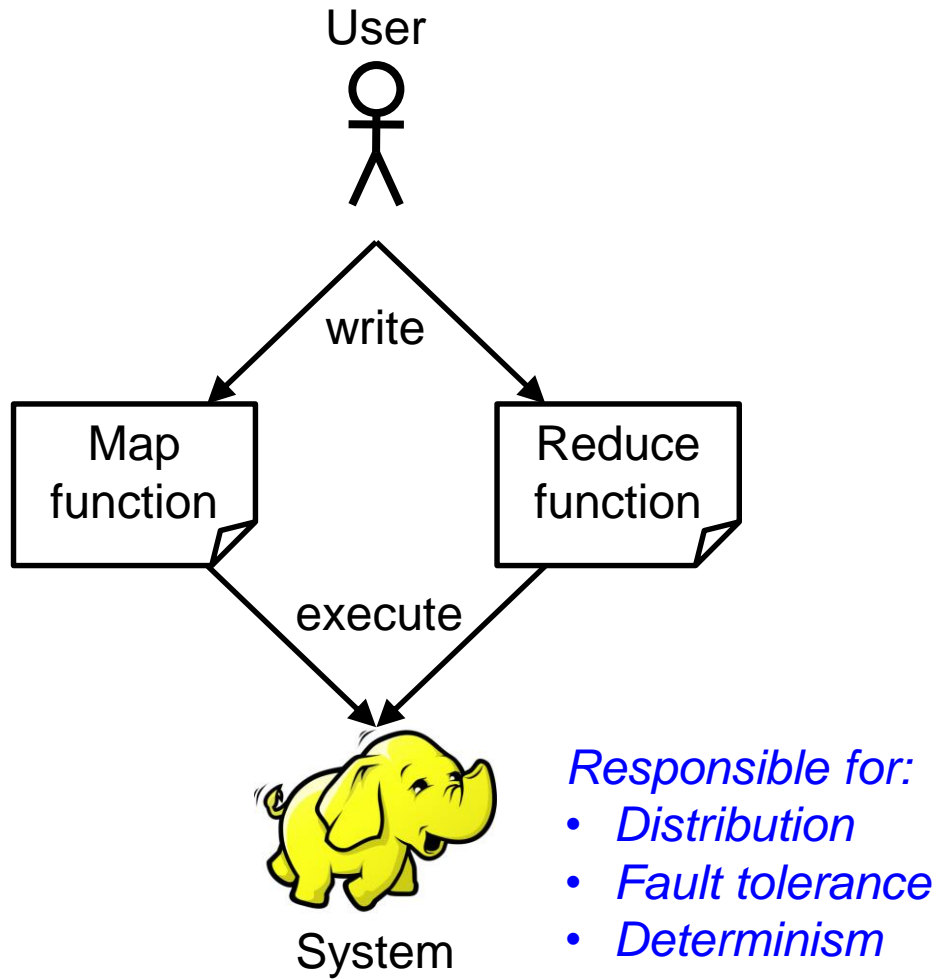
Zhihong Xu
Salesforce

Gregg Rothermel
U. Nebraska, Lincoln

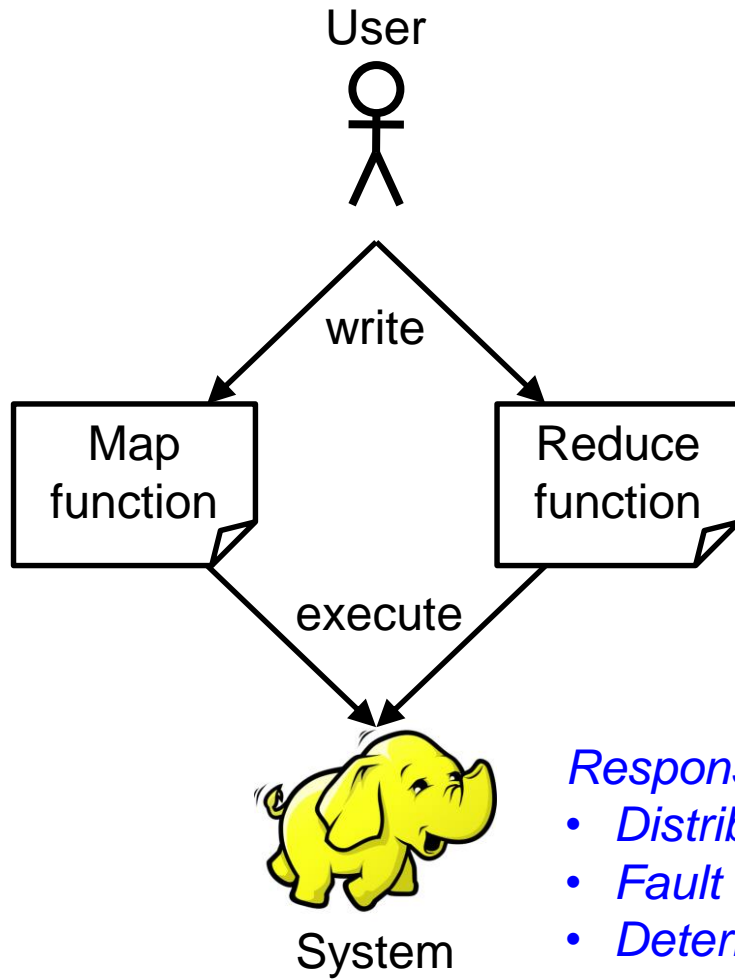
Martin Hirzel
IBM Research

IISWC 2013

MapReduce Promise: Simplicity



Is it Really so Simple?



Responsible for:

- *Distribution*
- *Fault tolerance*
- *Determinism*

Problem

- *Required properties are under-specified*

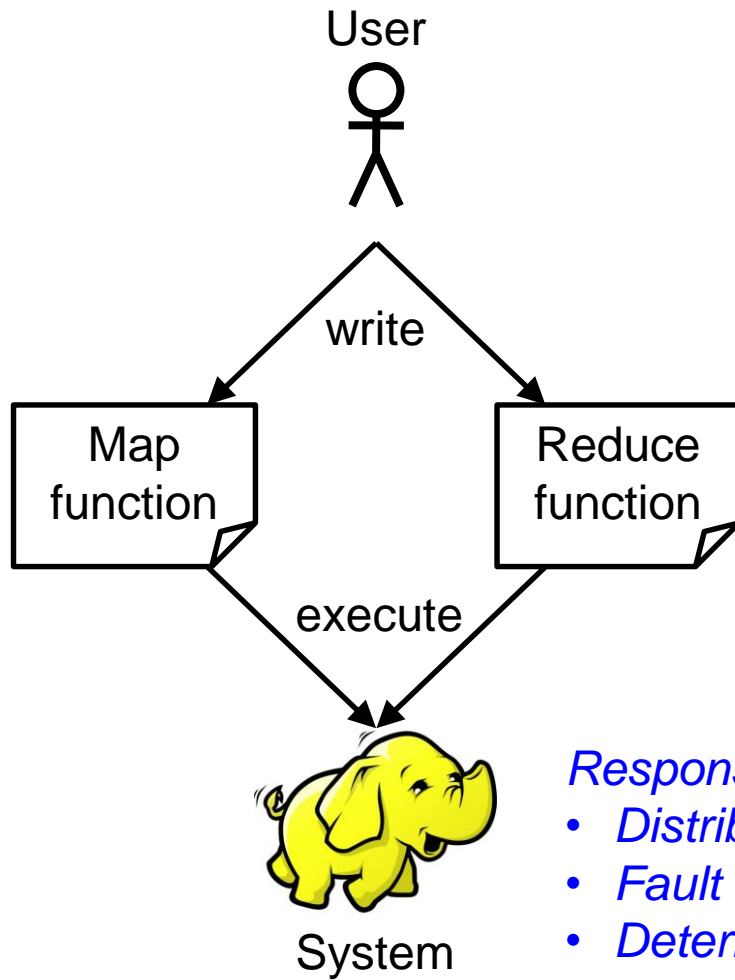
Problem

- *Unknown if properties hold in real workloads*

Problem

- *Implications of property violations not clear*

Contributions



- Responsible for:*
- *Distribution*
 - *Fault tolerance*
 - *Determinism*

Problem

- *Required properties are under-specified*

This paper:

- **Requirements specification**

Problem

- *Unknown if properties hold in real workloads*

This paper:

- **Workload characterization**

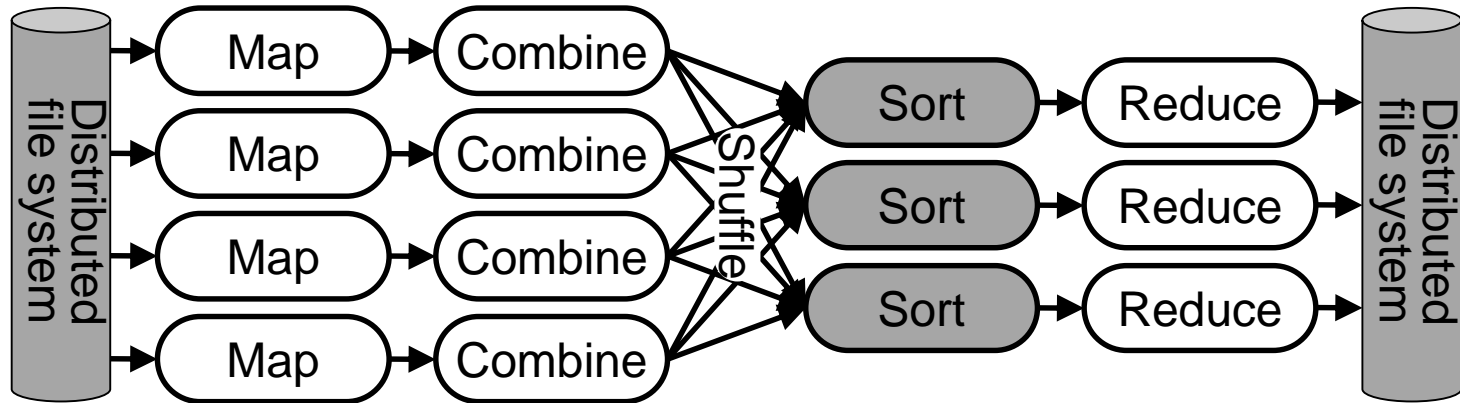
Problem

- *Implications of property violations not clear*

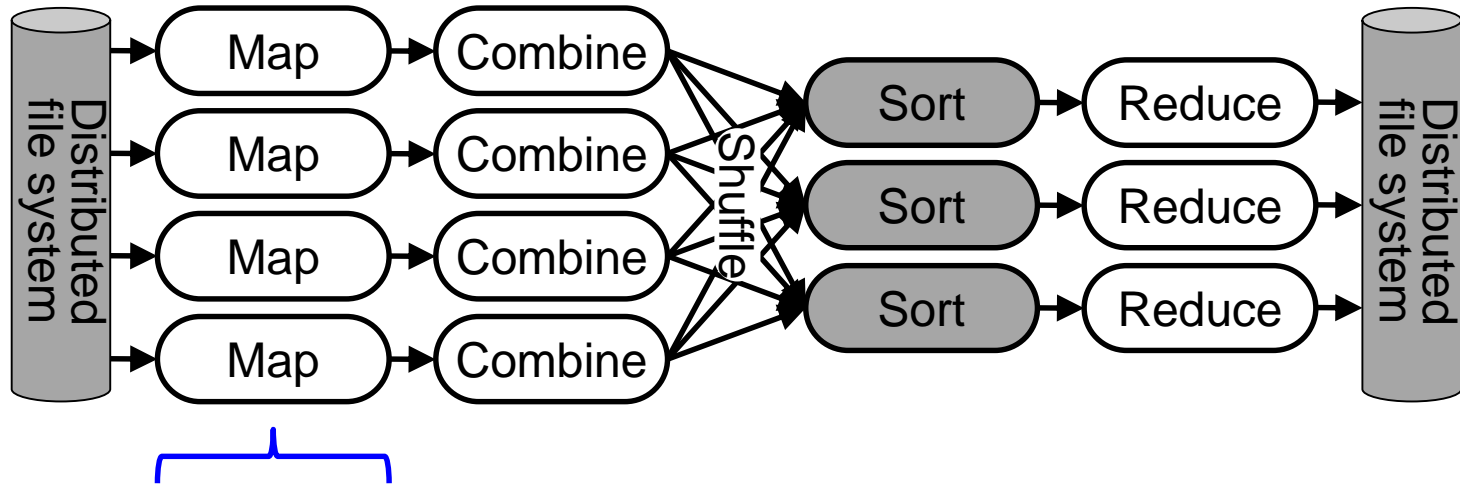
This paper:

- **Discussion of implications**

MapReduce

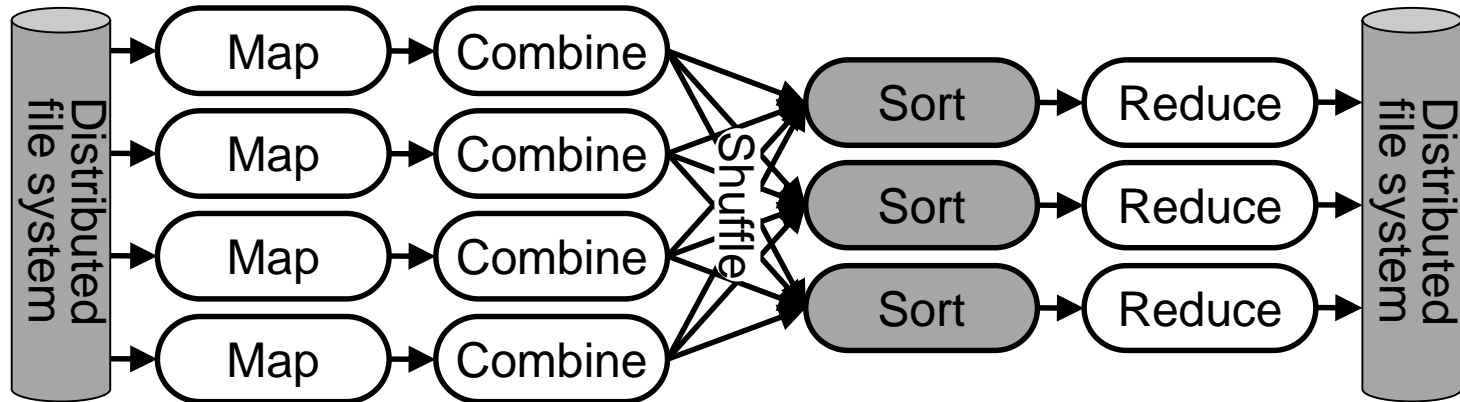


Usage of Map



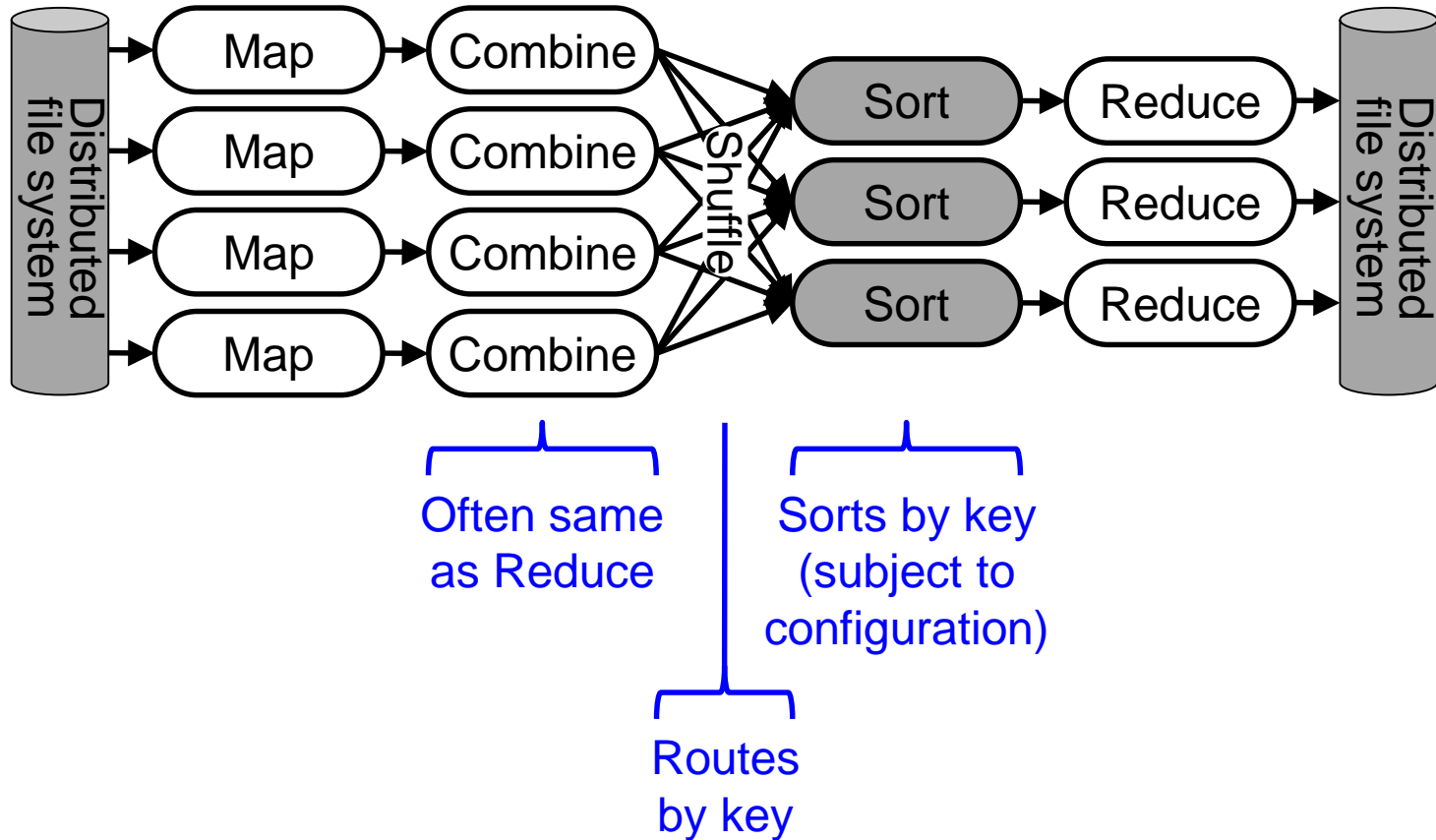
- Called one value at a time
- Can run on any worker
- May be restarted for fault tolerance
- Can run in any order

Usage of Reduce

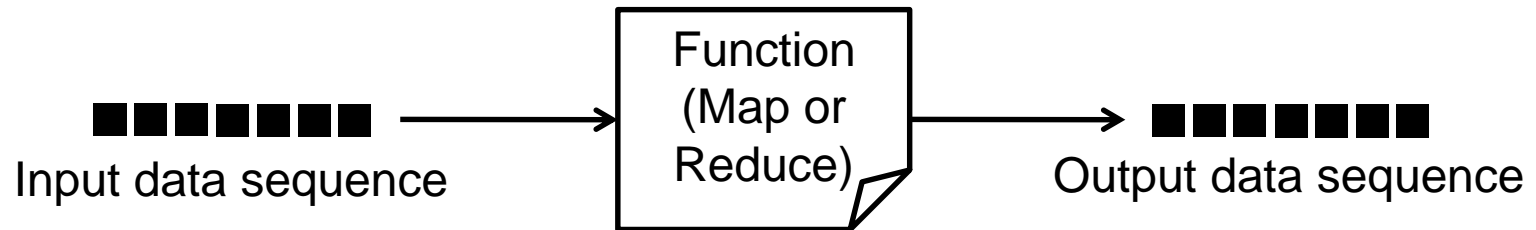


- Called one key at a time
- Can run on any worker
- May be restarted for fault tolerance

Between Map and Reduce



Property Definitions



- Determinism
- Statelessness
- Commutativity
- Partition-isolation
- Associativity
- Selectivity

Determinism

Same sequence of inputs implies same sequence of outputs.

Run 1:

Input	Output
2	2
2	4
3	7

Run 2:

Input	Output
2	2
2	4
3	5

non-deterministic

Statelessness

Historical inputs do not affect current output.

Input	Output
1	1
2	3
1	4

stateful

Commutativity

Order of inputs does not affect output.

Run 1:

Input	Output
1	
2	
3	3

Run 2:

Input	Output
1	
3	
2	2

non-commutative

Partition-Isolation

Inputs from key c_2 don't affect outputs for key c_1 .

Run 1:

Input	Output
$c_1:1$	1
$c_1:3$	4

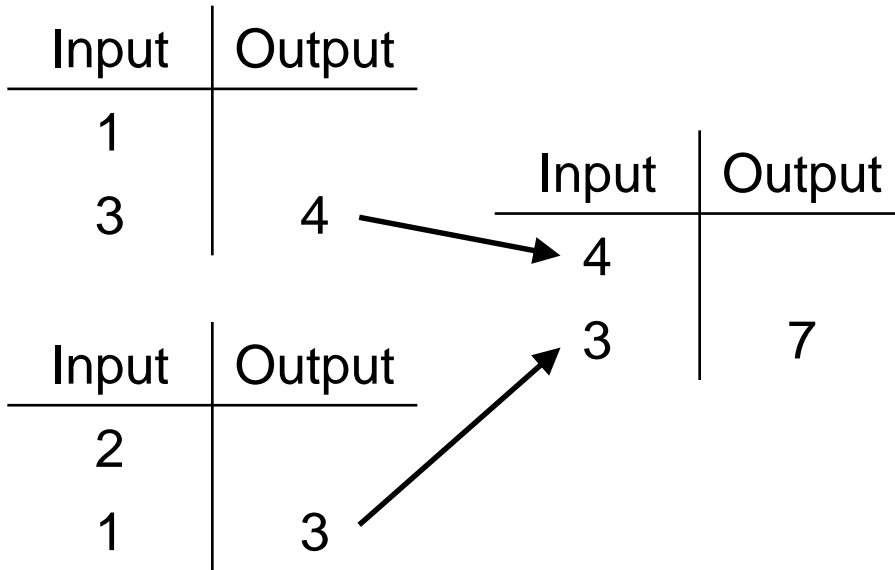
Run 2:

Input	Output
$c_1:1$	1
$c_2:2$	2
$c_1:3$	6

partition-interfering

Associativity

Grouping of inputs does not affect output.



Input	Output
1	
3	
2	
1	7

Selectivity

Selective:
Always ≤ 1 and
sometimes 0
outputs per input.

Input	Output
1	
3	3
0	

One-to-one:
Always =1 output
per input.

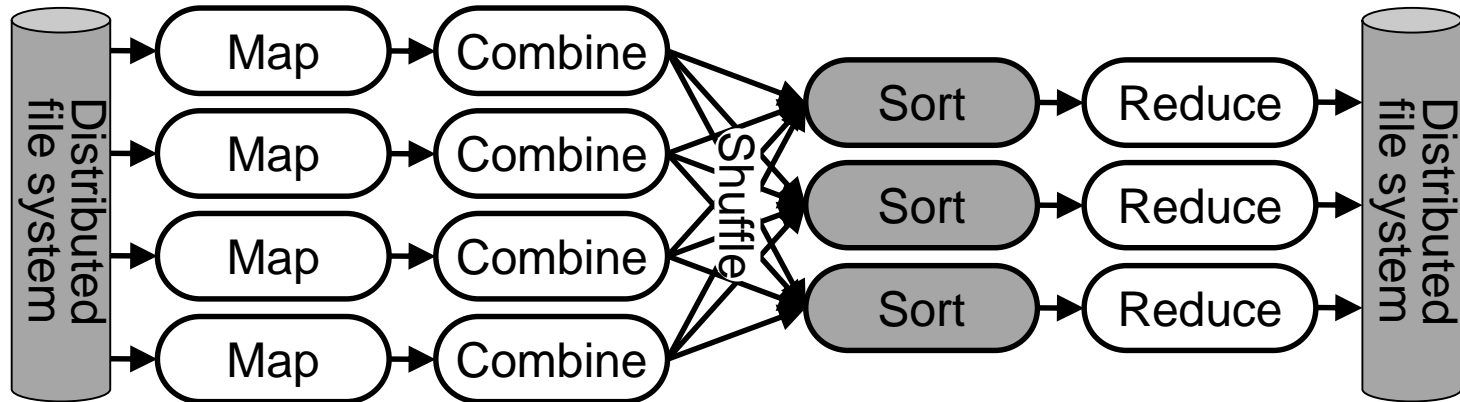
Input	Output
1	2
3	6
0	0

Prolific:
Sometimes >1
outputs per input.

Input	Output
1	1
3	1,2,3
0	

Requirements (and Expectations)

Assuming top-level determinism requirement



Deterministic

Stateless

Commutative

Partition-isolated

(Associative)

(One-to-one)

Deterministic

(Intra-key stateful)

Intra-key commutative

Partition-isolated

Associative if Combine

(Selective)

Workload Characterization

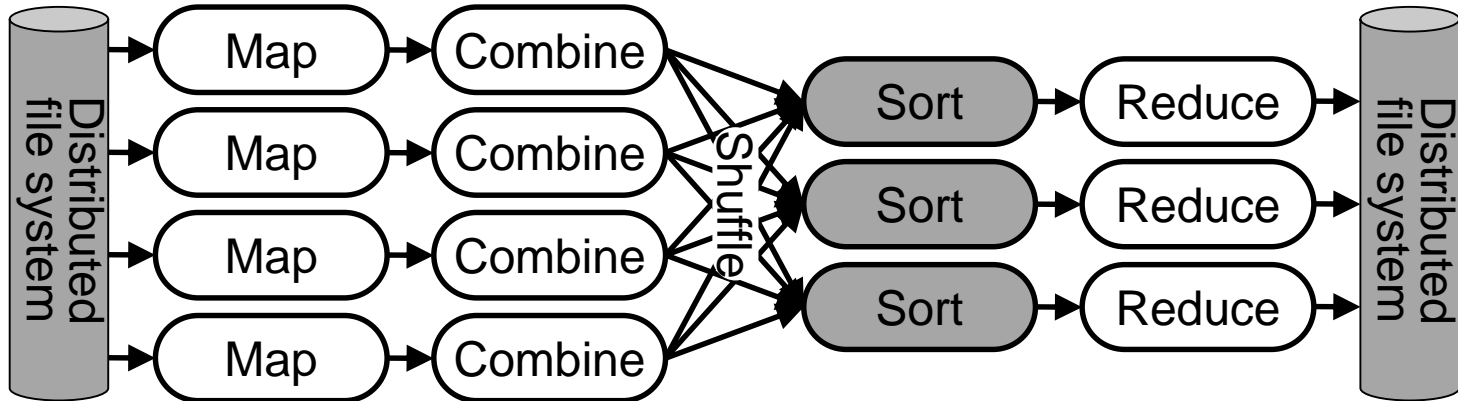
```
for each func in workloads
  for each prop in {determinism, statelessness, ...}
    evidence ← null
    start timer
    while evidence == null and not (timelimit reached)
      test ← generate random test for prop on func
      evidence ← run test using Hadoop libraries
    result[func, prop] ← (evidence == null ? true : violated)
```

Workloads

Application	Source	Lines of code	Functions map red.	
MultiFileWordCount	Hadoop	178	1	1
QuasiMonteCarlo	Hadoop	208	1	1
RandomTextWriter	Hadoop	651	1	0
RandomWriter	Hadoop	185	1	1
SecondarySort	Hadoop	161	1	1
Sort	Hadoop	143	1	1
WordCount	Hadoop	60	1	1
Anagrams	GitHub	79	1	1
ApacheLogAnalyzer	GitHub	224	1	1
CustomKey	GitHub	255	1	1
CVSPairThreshold	GitHub	80	1	1
Dictionary	GitHub	92	1	1
FacebookBuzzCount	GitHub	107	1	1
Geolocation	GitHub	91	1	1
ReduceSideJoin	GitHub	119	2	1
ScoreFriends	GitHub	273	4	2
UserAccessCount	GitHub	69	1	1
FarmerMarket	YouTube	99	1	1
Canopy	Mahout	170,913	3	1
Dirichlet	Mahout	170,913	2	1
FuzzyKMeans	Mahout	170,913	4	2
KMeans	Mahout	170,913	3	1
MeanShift	Mahout	170,913	4	1

Property Violations

Assuming top-level determinism requirement



- ~~Deterministic~~
- ~~Stateless~~
- ~~Commutative~~
- ~~Partition-isolated~~
- ~~(Associative)~~
- ~~(One-to-one)~~

- Deterministic
- ~~(Intra-key stateful)~~
- ~~Intra-key commutative~~
- Partition-isolated
- ~~Associative if Combine~~
- ~~(Selective)~~

Example Property Violations

Application	Source	Lines of code	Functions map red.	
MultiFileWordCount	Hadoop	178	1	1
QuasiMonteCarlo	Hadoop	208	1	1
RandomTextWriter	Hadoop	651	1	0
RandomWriter	Hadoop	185	1	1
SecondarySort	Hadoop	161	1	1
Sort	Hadoop	143	1	1
WordCount	Hadoop	60	1	1
Anagrams	GitHub	79	1	1
ApacheLogAnalyzer	GitHub	224	1	1
CustomKey	GitHub	255	1	1
CVSPairThreshold	GitHub	80	1	1
Dictionary	GitHub	92	1	1
FacebookBuzzCount	GitHub	107	1	1
Geolocation	GitHub	91	1	1
ReduceSideJoin	GitHub	119	2	1
ScoreFriends	GitHub	273	4	2
UserAccessCount	GitHub	69	1	1
FarmerMarket	YouTube	99	1	1
Canopy	Mahout	170,913	3	1
Dirichlet	Mahout	170,913	2	1
FuzzyKMeans	Mahout	170,913	4	2
KMeans	Mahout	170,913	3	1
MeanShift	Mahout	170,913	4	1

Random map

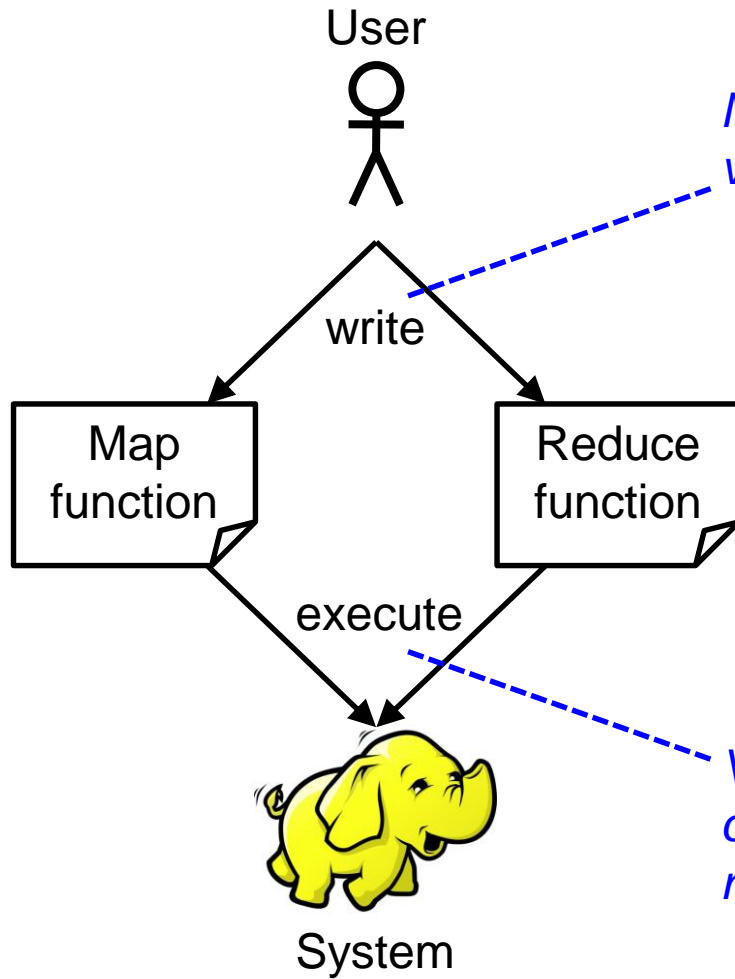
Prolific reduce
(adds split line)

Intra-key stateless
reduce

Non-associative
combine (commas)

Stateful map

Implications



Most properties are violated at least once.

Remediation ideas:

- Offer bug-finding tools.
- Make system more robust when property violated.
- Make system more efficient when property satisfied.

Violated properties cause slow, incorrect, or non-deterministic runs.

Conclusions

- Contributions:
 - Requirements specification
 - Workload characterization
 - Discussion of implications
- We hope this paper will help:
 - Users write more robust MapReduce code
 - Designers develop more robust systems