# *Software testing in MapReduce applications*

Jesús Morán

**Software Engineering Research Group**
**http://giis.uniovi.es**
**University of Oviedo**

# Research lines

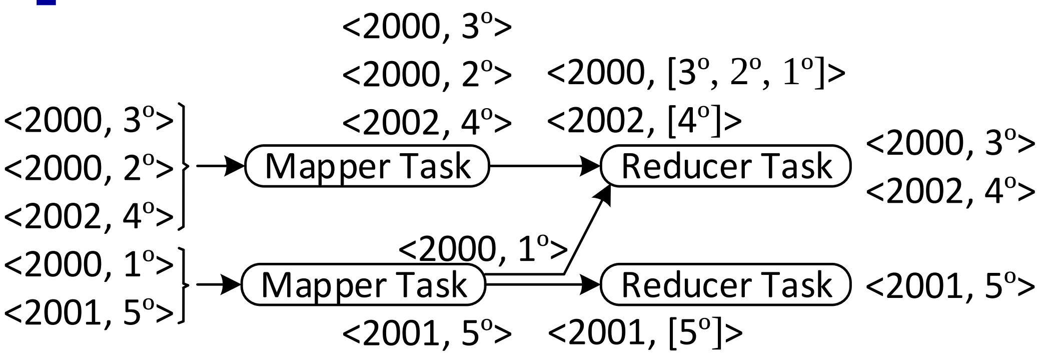
- Test case generation
- Test case execution
- Functional improvement

# MapReduce

- Divide and Conquer:

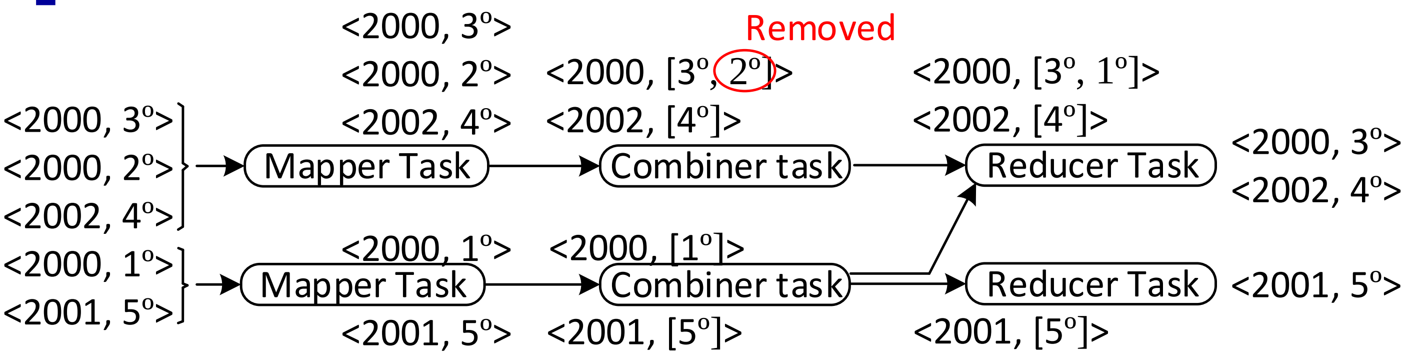
- Mapper: Divide
- Reducer: Conquer

# MapReduce (maximum temperature per year)



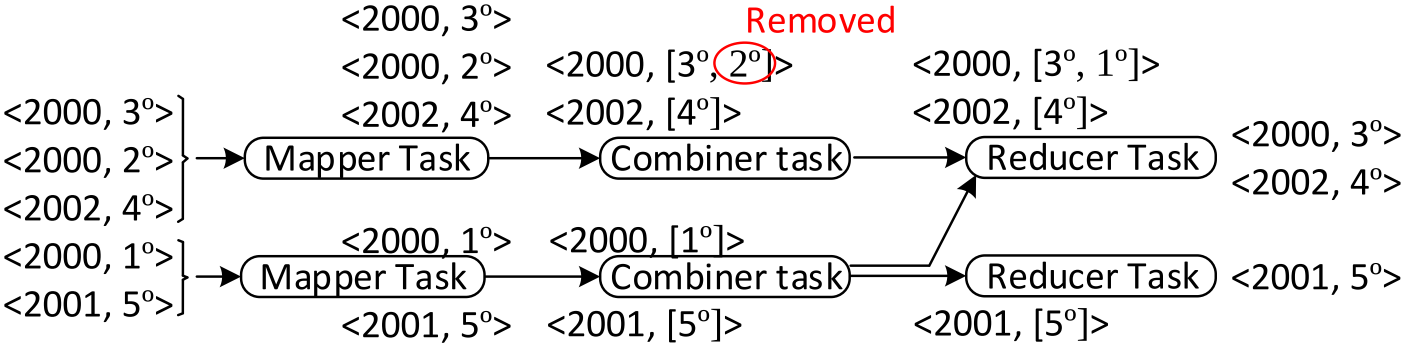
- Divide and Conquer:
  - □ 3 subproblems: 2000, 2001 and 2002
- Mapper: Divide
- Reducer: Conquer

# MapReduce (maximum temperature per year)



- Divide and Conquer:
  - 3 subproblems: 2000, 2001 and 2002
- Mapper: Divide
- Reducer: Conquer
- Combiner: Local Reducer

# MapReduce (maximum temperature per year)



<2000, 3°>
<2000, 2°>
<2002, 4°>

Mapper Task

<2000, 3°>
<2000, 2°>
<2002, 4°>

Combiner task

<2000, [3°, 2°]>
<2002, [4°]>

Removed

<2000, [3°, 1°]>
<2002, [4°]>

Reducer Task

<2000, 3°>
<2002, 4°>

<2000, 1°>
<2001, 5°>

Mapper Task

<2000, 1°>
<2001, 5°>

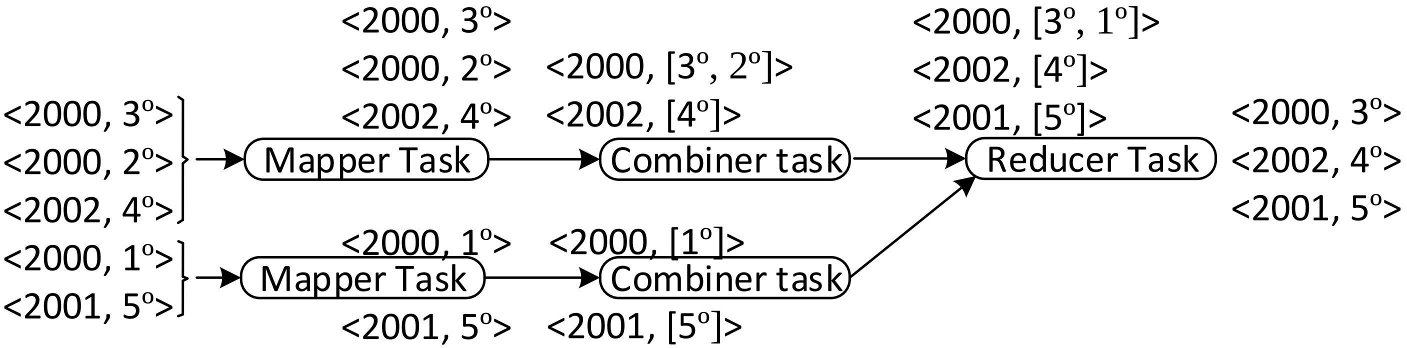
Combiner task

<2000, [1°]>
<2001, [5°]>

Reducer Task

<2001, [5°]>

<2001, 5°>



<2000, 3°>
<2000, 2°>
<2002, 4°>

Mapper Task

<2000, 3°>
<2000, 2°>
<2002, 4°>

Combiner task

<2000, [3°, 2°]>
<2002, [4°]>

<2000, [3°, 1°]>
<2002, [4°]>
<2001, [5°]>

Reducer Task

<2000, 3°>
<2002, 4°>
<2001, 5°>

<2000, 1°>
<2001, 5°>

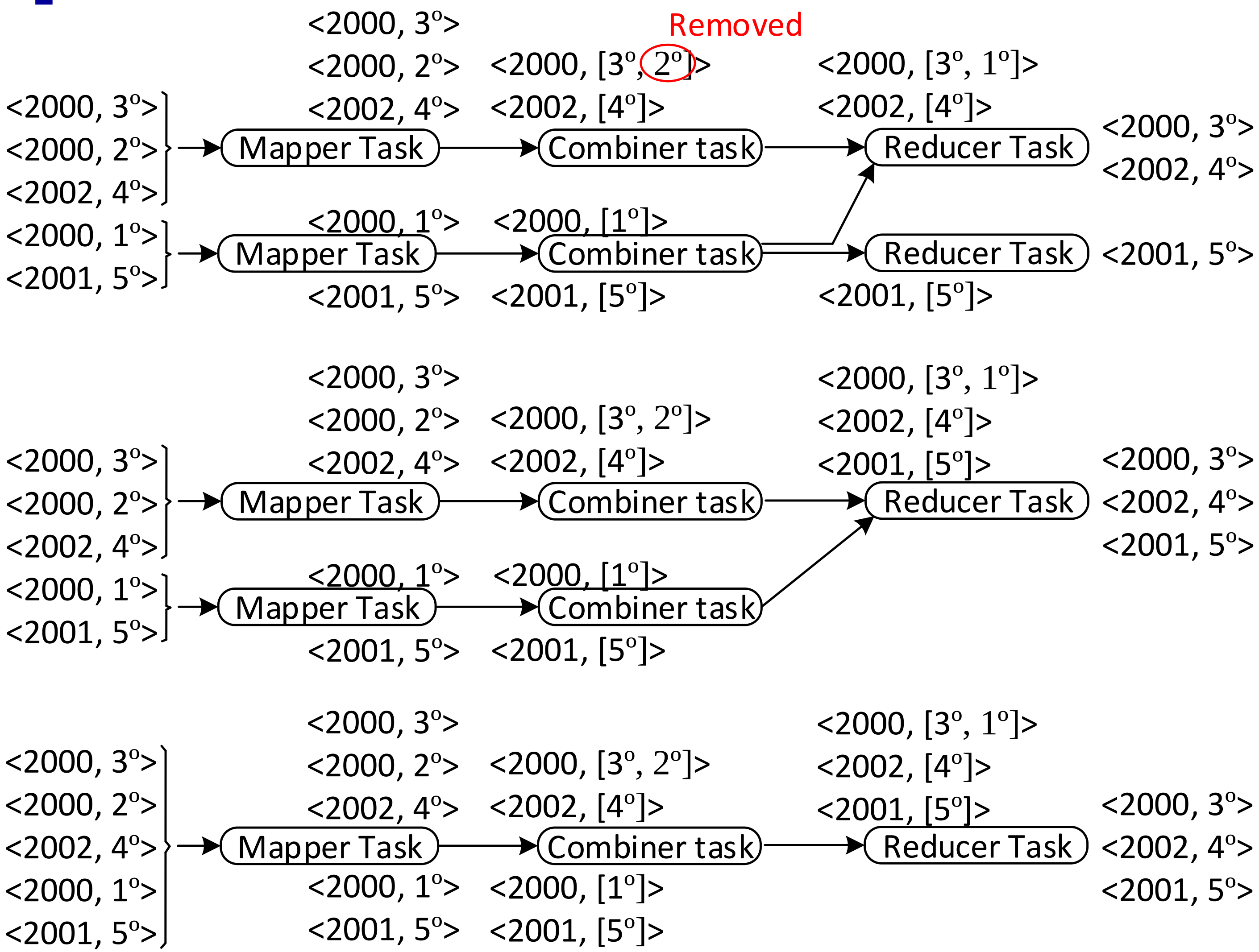
Mapper Task

<2000, 1°>
<2001, 5°>

Combiner task

<2000, [1°]>
<2001, [5°]>

# MapReduce (maximum temperature per year)



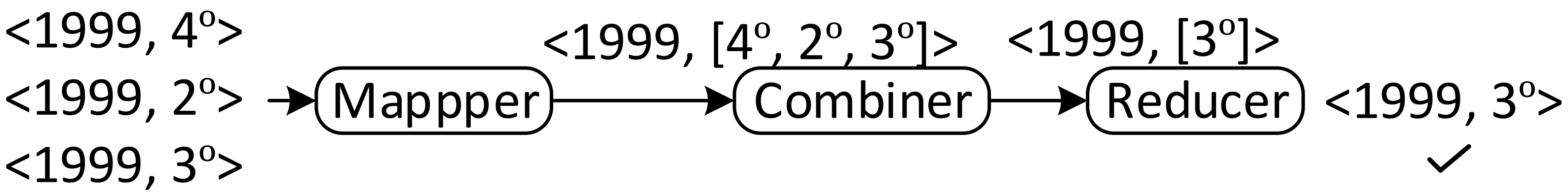
Removed

<2000, 3°>
<2000, 2°>
<2002, 4°>
<2000, 3°>
<2000, 2°>
<2002, 4°>
Mapper Task
<2000, [3°, 2°]>
<2002, [4°]>
Combiner task
<2000, [3°, 1°]>
<2002, [4°]>
Reducer Task
<2000, 3°>
<2002, 4°>
<2000, 1°>
<2001, 5°>
Mapper Task
<2000, 1°>
<2001, 5°>
Combiner task
<2000, [1°]>
<2001, [5°]>
Reducer Task
<2001, [5°]>
<2001, 5°>

<2000, 3°>
<2000, 2°>
<2002, 4°>
<2000, 3°>
<2000, 2°>
<2002, 4°>
Mapper Task
<2000, [3°, 2°]>
<2002, [4°]>
Combiner task
<2000, [3°, 1°]>
<2002, [4°]>
<2001, [5°]>
Reducer Task
<2000, 3°>
<2002, 4°>
<2001, 5°>
<2000, 1°>
<2001, 5°>
Mapper Task
<2000, 1°>
<2001, 5°>
Combiner task
<2000, [1°]>
<2001, [5°]>

<2000, 3°>
<2000, 2°>
<2002, 4°>
<2000, 1°>
<2001, 5°>
<2000, 3°>
<2000, 2°>
<2002, 4°>
Mapper Task
<2000, 1°>
<2001, 5°>
<2000, [3°, 2°]>
<2002, [4°]>
Combiner task
<2000, [1°]>
<2001, [5°]>
<2000, [3°, 1°]>
<2002, [4°]>
<2001, [5°]>
Reducer Task
<2000, 3°>
<2002, 4°>
<2001, 5°>

# MapReduce (avg temperature per year)



<1999, 4º>
<1999, 2º>
<1999, 3º>

Mappper

<1999, [4º, 2º, 3º]>

Combiner

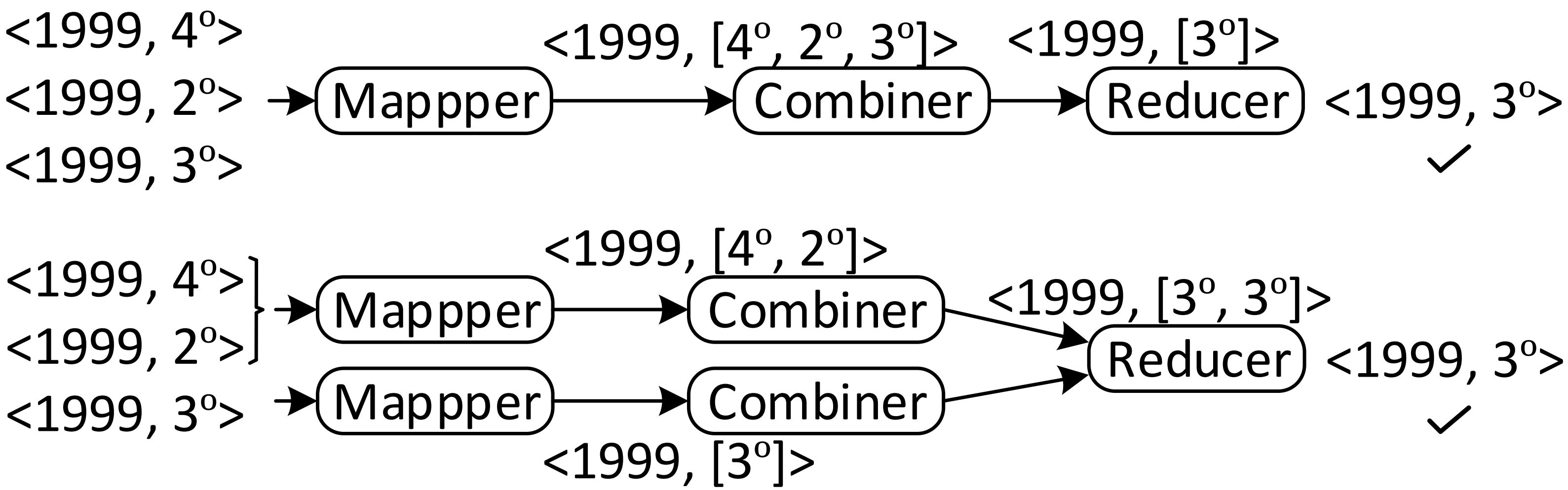
<1999, [3º]>

Reducer

<1999, 3º>

✓

# MapReduce (avg temperature per year)



<1999, $4^o$>
<1999, $2^o$>
<1999, $3^o$>

Mappper

<1999, [$4^o$, $2^o$, $3^o$]>

Combiner

<1999, [$3^o$]>

Reducer

<1999, $3^o$>

✓

<1999, $4^o$>
<1999, $2^o$>

Mappper

<1999, [$4^o$, $2^o$]>

Combiner

<1999, $3^o$>

Mappper

Combiner

<1999, [$3^o$]>

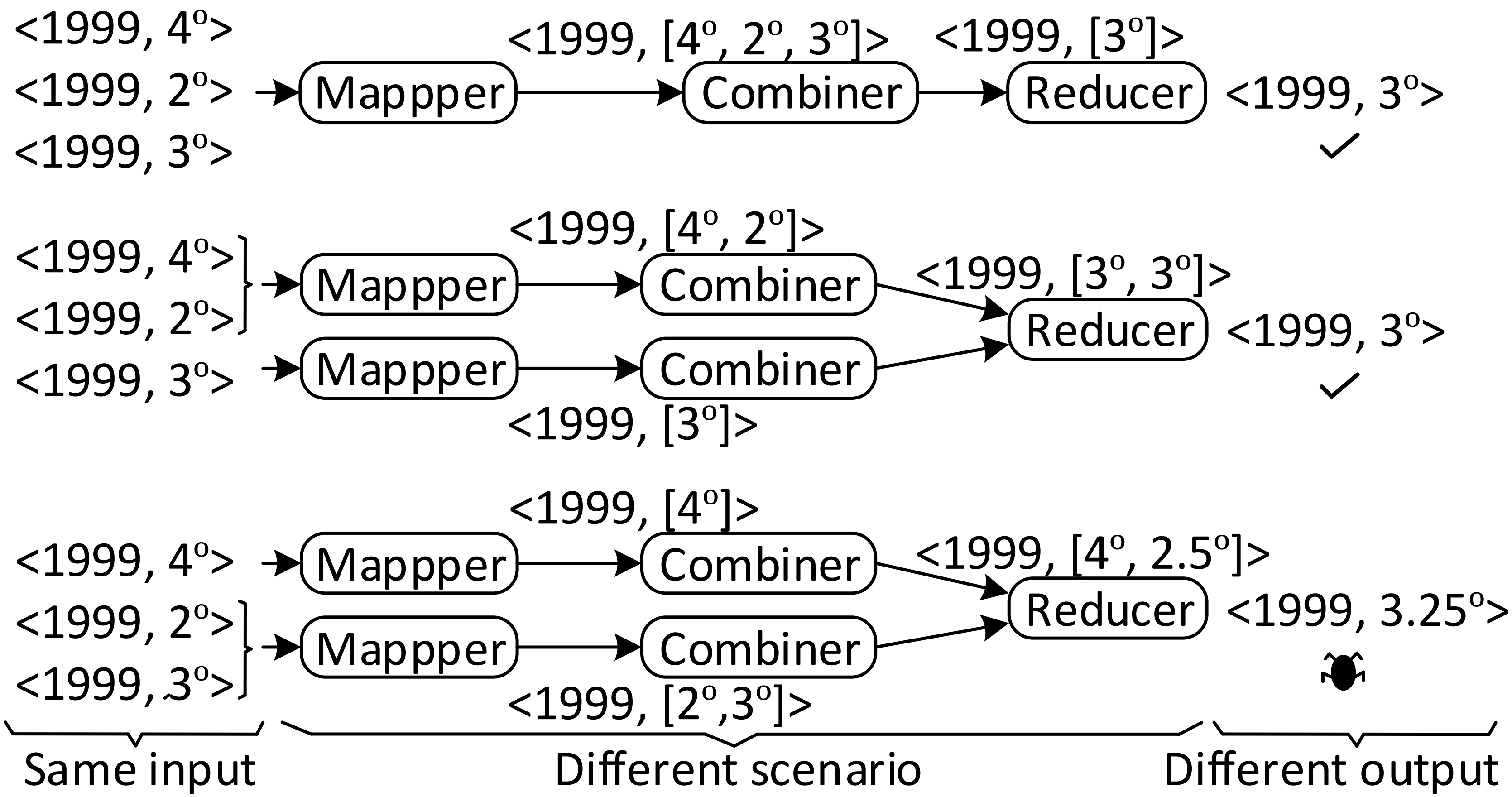
<1999, [$3^o$, $3^o$]>

Reducer

<1999, $3^o$>

✓

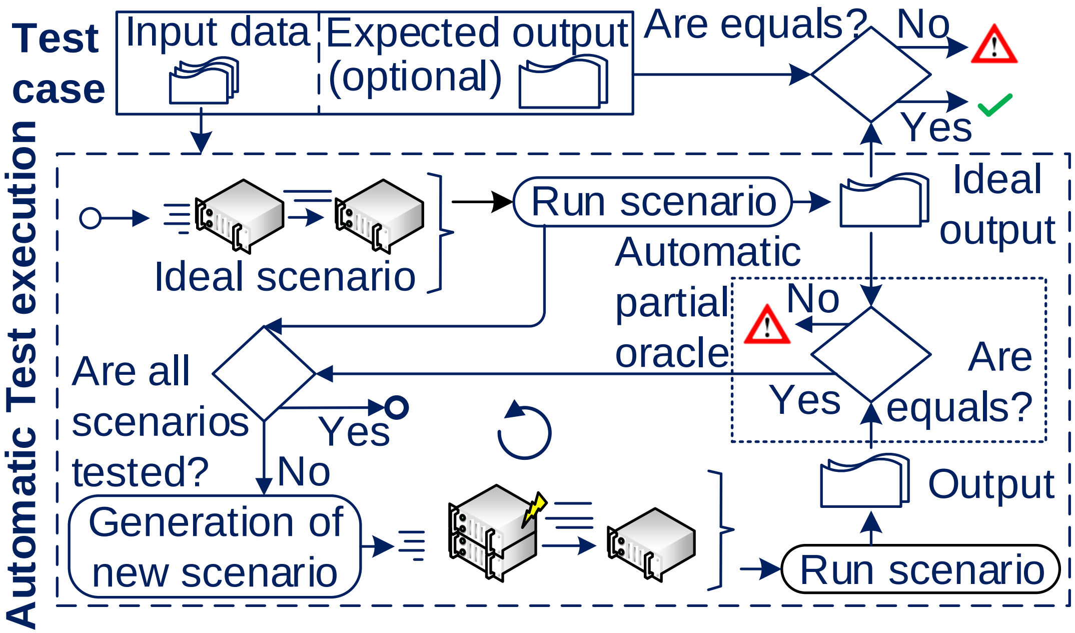
# MapReduce (avg temperature per year)



# Test execution problem

- Some failures depend on the execution
- Solution: Test case executed in all configurations

# Test execution engine



- Automatic test execution engine
- All configurations based on 7 parameters:
  - Number of Mappers, Combiners, Reducers, order of execution, …

# Case studies

| Program | Our test engine | MRUnit | Hadoop test environment | Hadoop production |
|---|---|---|---|---|
| Avg. temperature per year | Fault | - | - | - |
| Recommendation system | Fault | - | - | - |
| Data quality framework | Fault | - | - | - |
| Movies evaluation | Fault | - | - | - |

- The faults are hard to reveal
- Hadoop production does not reveal the faults due few test input data

# Test execution problem

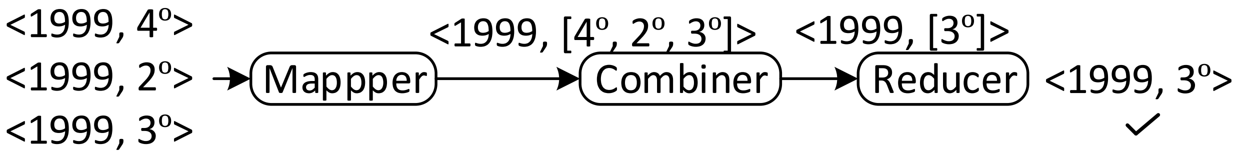
- Some failures depend on the execution
- Solution: Test case executed in all configurations
  - **Problem**: The number of configurations grow exponentially according to the test input data

# Test execution problem

- Some failures depend on the execution
- Solution: Test case executed in all configurations
  - Problem: The number of configurations grow exponentially according to the test input data
- Other solution: Search based testing to select the configurations prone to reveal a fault

# Future work

- Genetic algorithm



<1999, 4º>
<1999, 2º>
<1999, 3º>

Mappper

<1999, [4º, 2º, 3º]>

Combiner

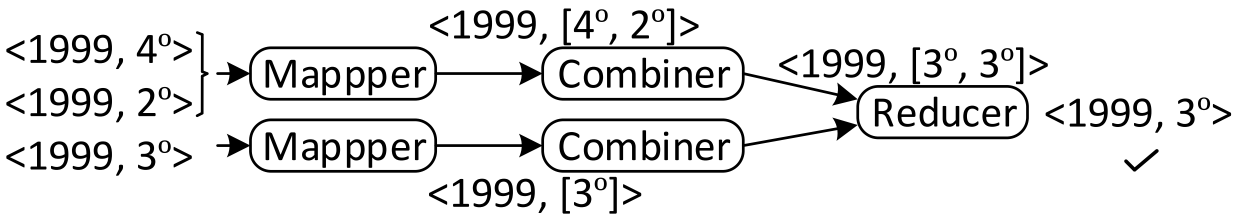
<1999, [3º]>

Reducer

<1999, 3º>
✓

# Future work

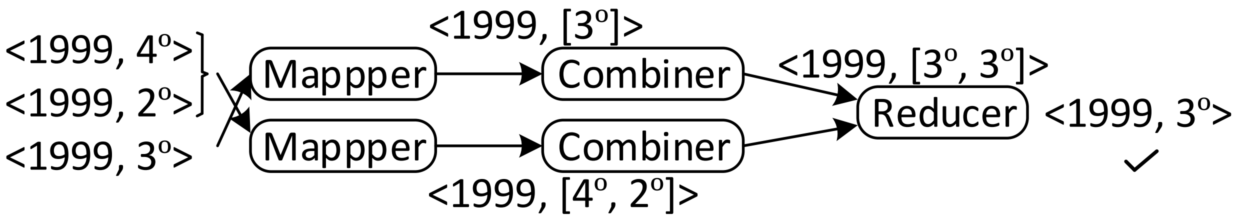
- Genetic algorithm



- Mutation: Add Mapper

# Future work

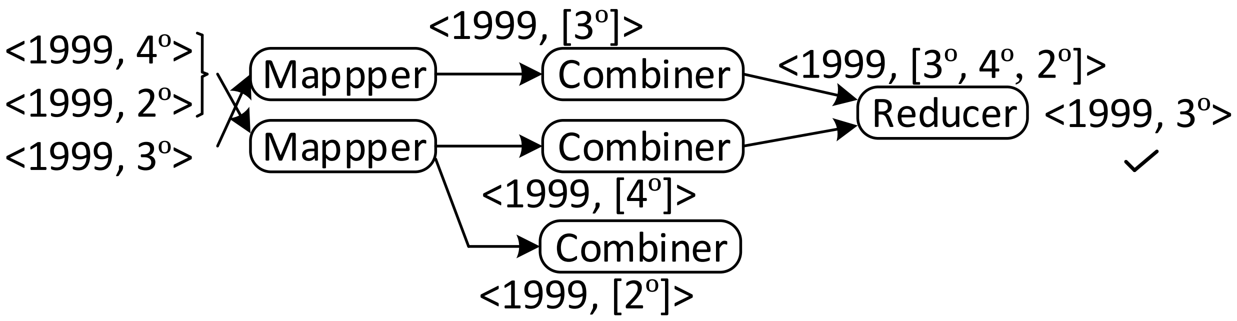
- Genetic algorithm



- Mutation: Different execution order

# Future work

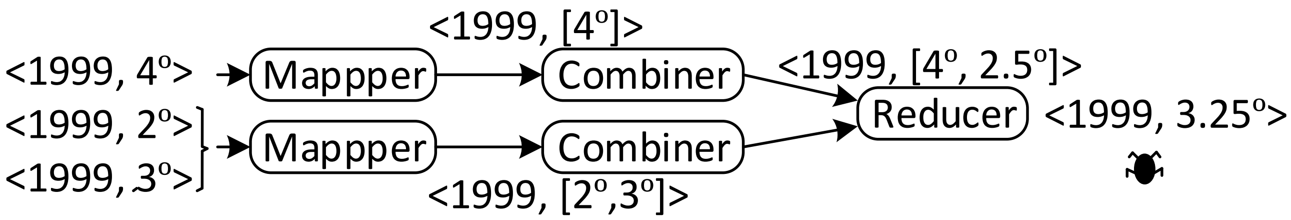
- Genetic algorithm



- Mutation: Add Combiner

# Future work

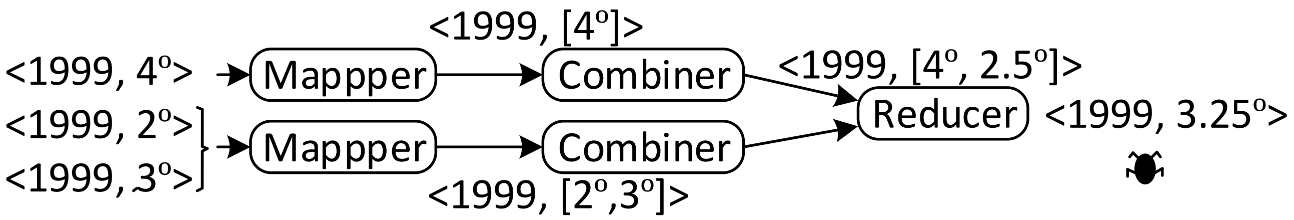
- Genetic algorithm



- Mutations and crossovers guided to detect faults

# Future work

- Genetic algorithm



<1999, 4º>

<1999, 2º>

<1999, 3º>

Mappper

Mappper

<1999, [4º]>

<1999, [2º,3º]>

Combiner

Combiner

<1999, [4º, 2.5º]>

Reducer

<1999, 3.25º>

- Mutations and crossovers guided to detect faults
- Fitness: Distance to failure
  - □ Comparison against the data of the ideal execution (1 Mapper + 1 Combiner + 1 Reducer)

# Questions?

Jesús Morán

**Software Engineering Research Group**
**http://giis.uniovi.es**
**Universidad of Oviedo**