Apache Nemo: A Framework for Building Distributed Dataflow Optimization Policies

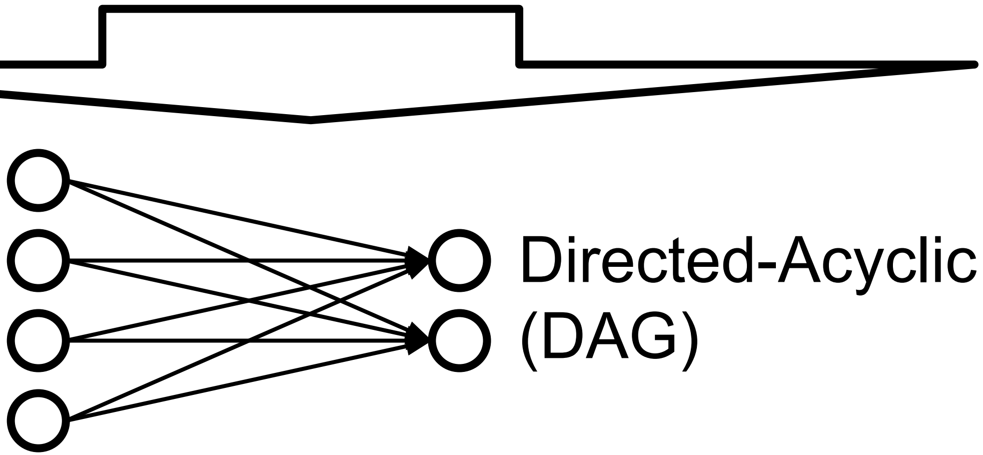
Youngseok Yang[1] Jeongyoon Eo[1] Geon-Woo Kim[2] Joo Yeon Kim[3]
Sanha Lee[4] Jangho Seo[1] Won Wook Song[1] Byung-Gon Chun[1]

[1]Seoul National University [2]Viva Republica [3]Samsung Electronics [4]Naver Corp.

Distributed data processing applications and runtime

Application

dataCollection.map(..).reduce(..)
(e.g., DryadLINQ, Spark RDD, Beam)



Runtime

Directed-Acyclic Graph
(DAG)

(e.g., Dryad, Tez, Spark, …)

Policy interfaces for scheduling/communication

Application-level
(e.g., Optimus)

```
register(stat,
subqueryA,subqueryB)
```

- Correctness: O
- Fine Control: X

Runtime-level
(e.g., Dryad)

```
onEvent(event) {
 modify(DAG, event)
}
```

- Correctness: X
- Fine Control: O

Our idea: Transform an intermediate representation

- **IR DAG**: Provides both correctness and fine control
- **Optimization pass**: A function to transform the IR DAG
- **Runtime extensions:** Apply the optimizations

Experimental Evaluation

- **Deployment scenarios**
  - Resources: Transient, Geo-distributed
  - Data: Large, Skewed
- **Results**
  - Nemo is on par with specialized runtimes
  - Nemo further improves performance for scenarios with combinations of resource/data characteristics

USENIX ATC 2019, 2:20PM, Track II, on July 10

https://nemo.apache.org/

Nemo