



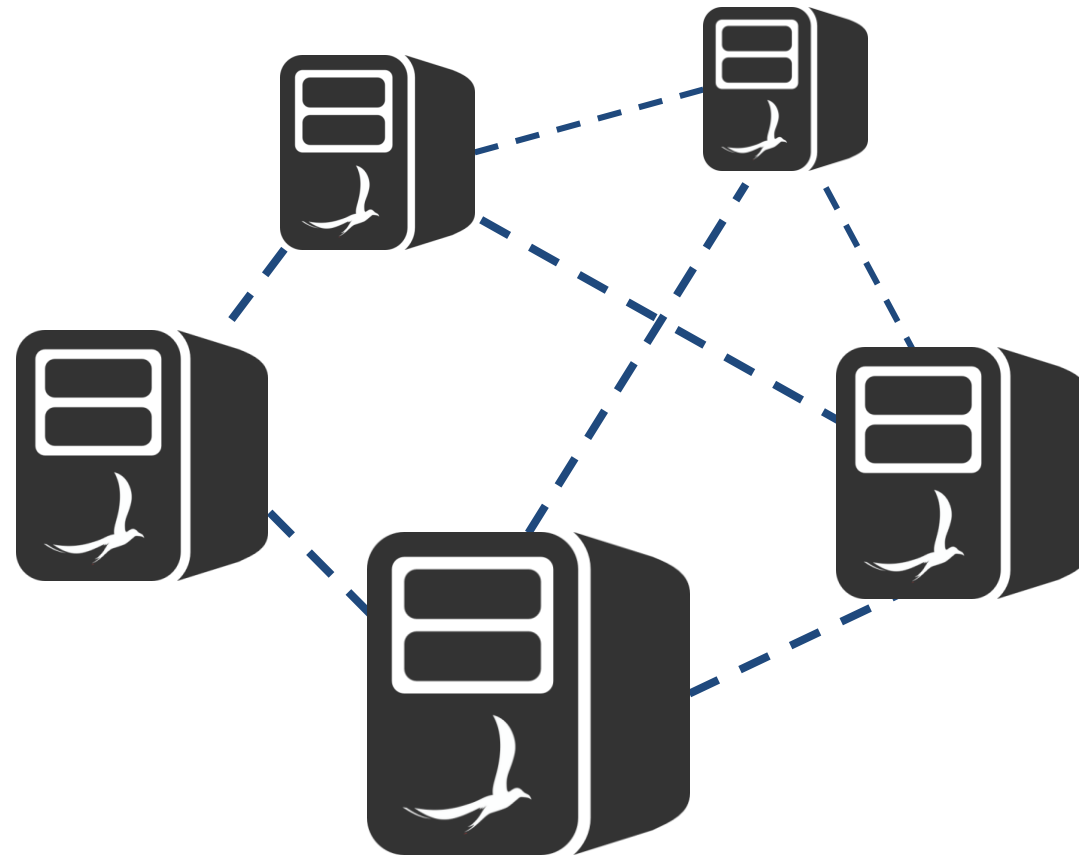
ONOS Distributed Core

James Won-Ki Hong, Jian Li, Seyeon Jeong

**Dept. of Computer Science & Engineering
POSTECH**

<http://dpm.postech.ac.kr/~jwkhong>
jwkhong@postech.ac.kr

Distributed Core



Distributed Architecture (1/7)

❖ Distributed

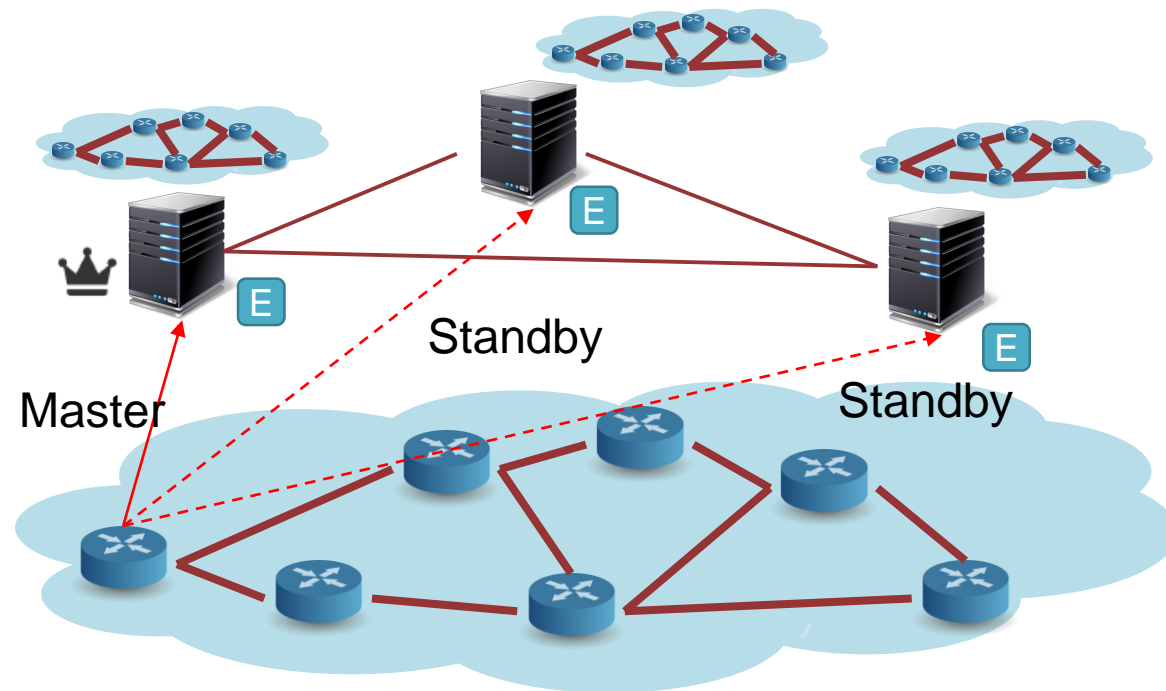
- Setup as a cluster of instances

❖ Symmetric

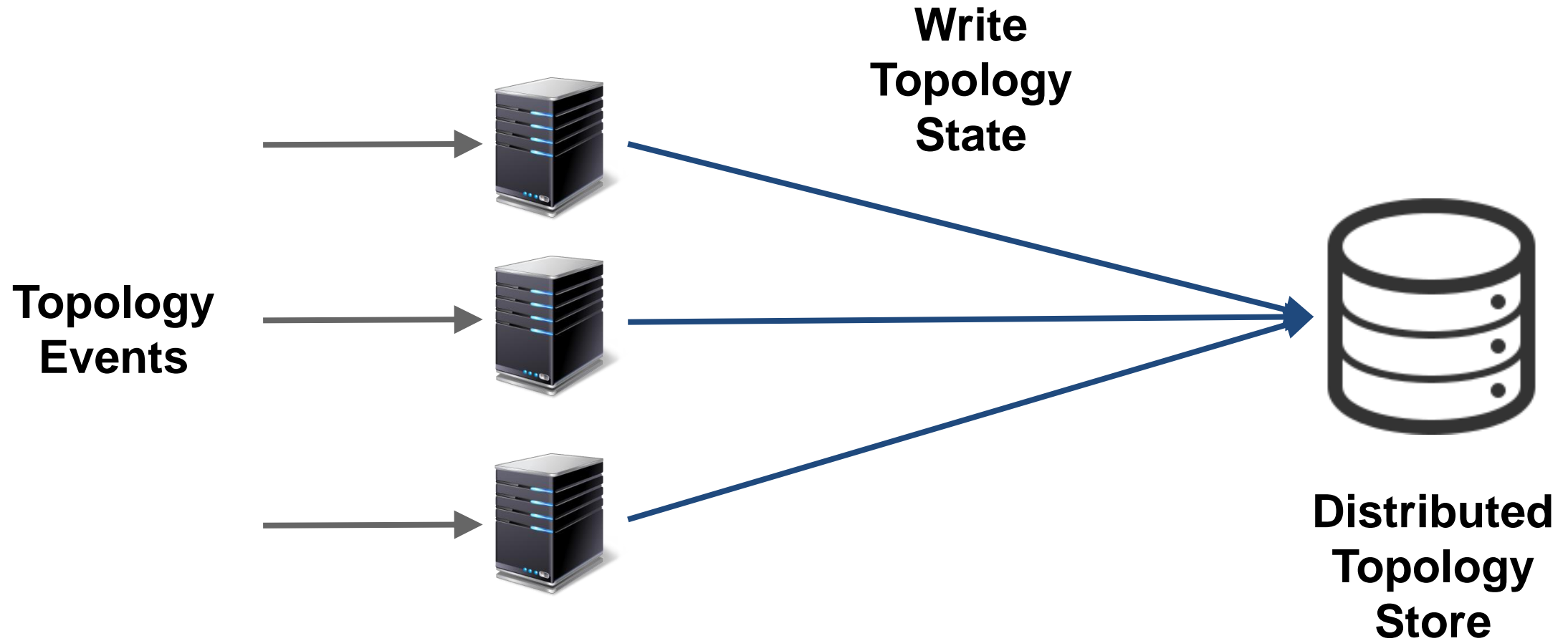
- Each instance runs identical software and configuration

❖ Fault-tolerant

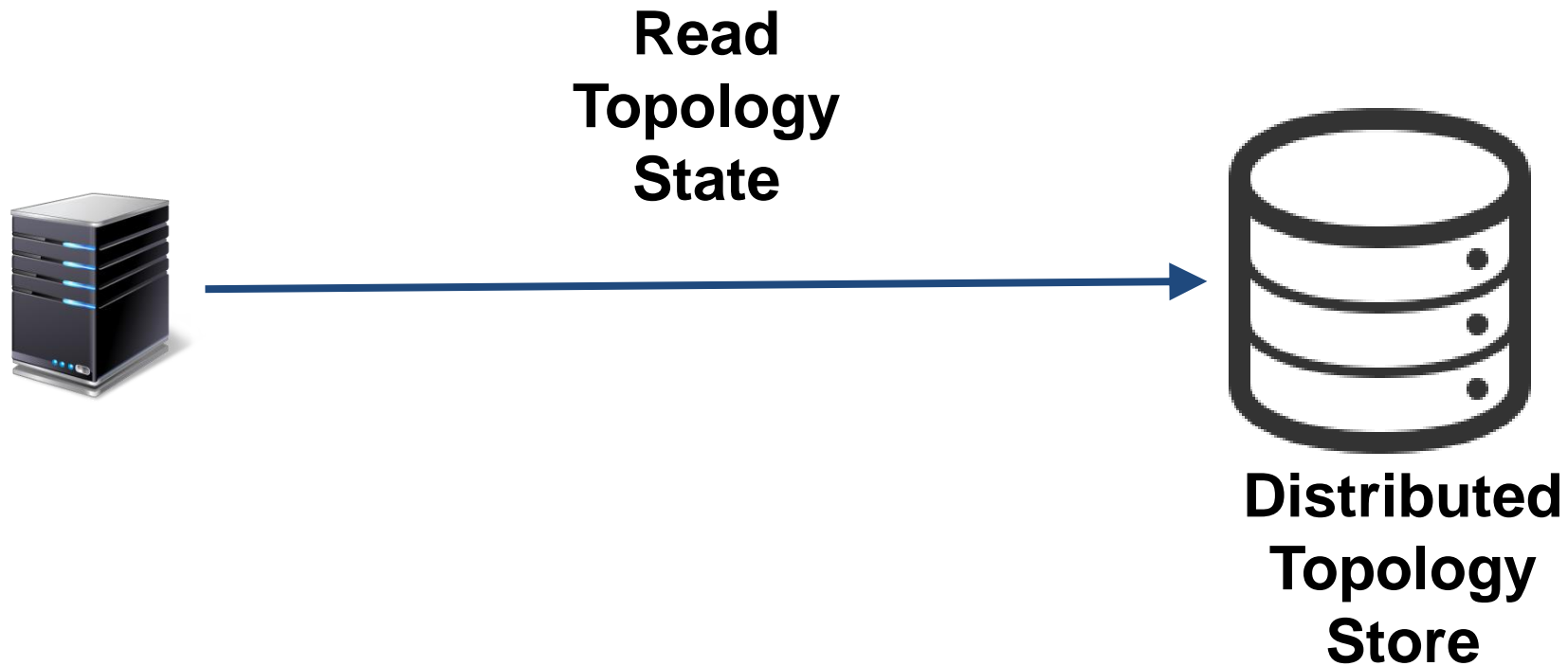
- Cluster remains operational in the face of node failures



Distributed Architecture (2/7)



Distributed Architecture (3/7)



❖ State Management in ONOS

- ONOS exposes a set of distributed primitives to cater to different use cases
- Primitives span the consistency continuum

❖ Distributed Primitives

- EventuallyConsistentMap<K, V>
 - Map abstraction with eventual consistency guarantee
- ConsistentMap<K, V>
 - Map abstraction with strong linearizable consistency
- DistributedQueue<E>
 - Distributed FIFO queue with long poll support
- AtomicCounter
 - Distributed version of Java AtomicLong
- Etc.



❖ Strong Consistency

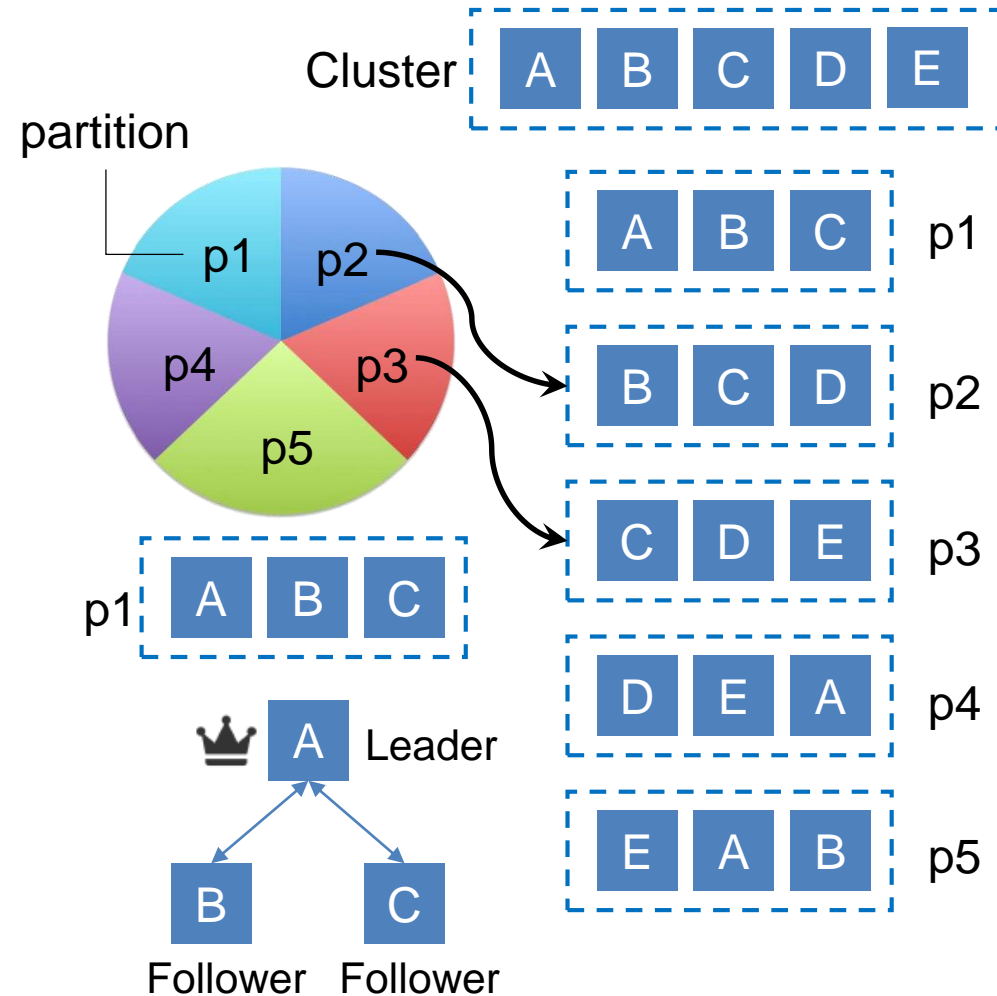
- Raft consensus protocol
- Data is partitioned into Replica Sets

❖ Eventual Consistency

- Gossip protocol

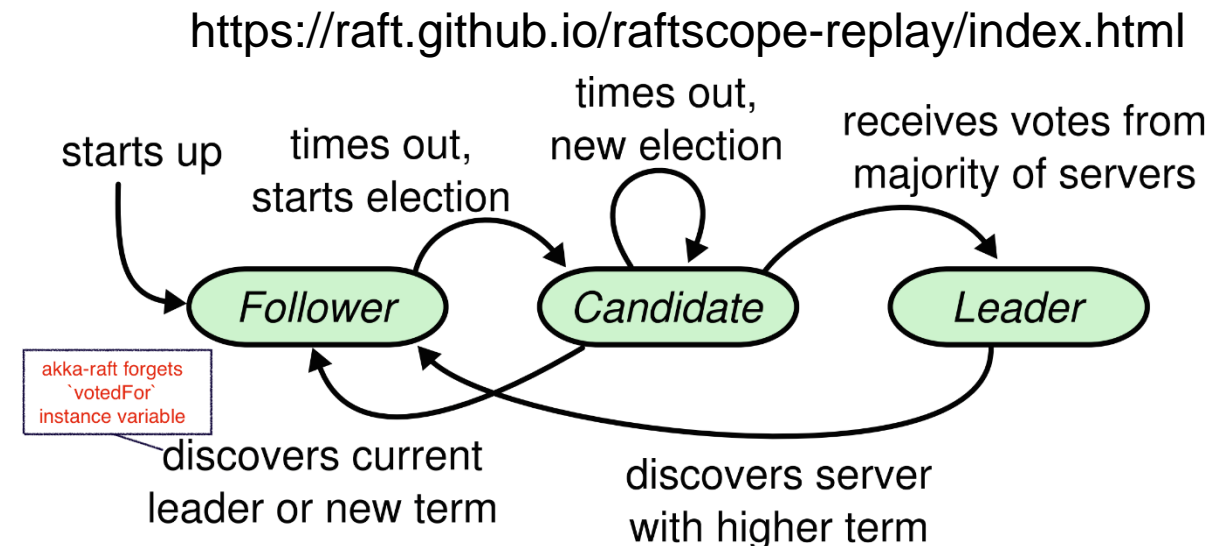
❖ Copicat

- A Java implementation of Raft consensus protocol
- Open source project



❖ Raft Consensus Protocol

- Leader sends **heartbeats** to maintain authority
- Randomize timeouts
- Upon **election timeout**, start new election
 - Increment current term
 - Change to *candidate* state
 - Vote for self
 - Send request vote to all other controllers
- wait until:
 - Receive votes from majority of controllers
→ become leader
 - Receive request vote from valid leader
→ become follower
 - No one wins election
→ increment term, start new election



❖ Gossip Protocol (or Epidemic Protocol)

- Analogy of office workers spreading rumors
- Power of gossip protocol lies in the robust spread of information

