Pragh:

Locality-preserving Graph Traversal with Split Live Migration

Xiating Xie, Xingda Wei, Rong Chen, Haibo Chen
Shanghai Jiao Tong University

IPADS
INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEMS

Graphs are Everywhere

Alipay™

Pay

DBpedia

Graph *traversal queries* are key operations to support emerging applications.

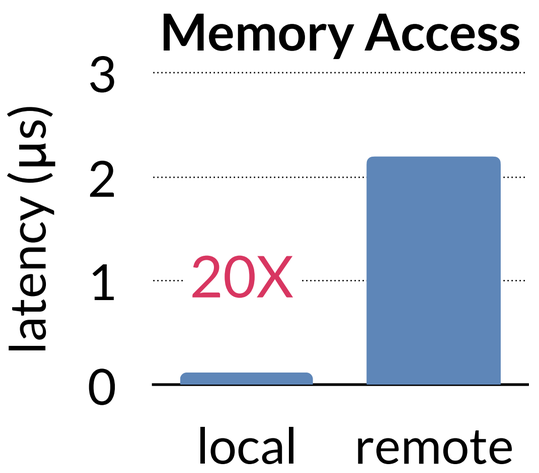
FRAUD DETECTION

Urban Monitoring

User Profiling

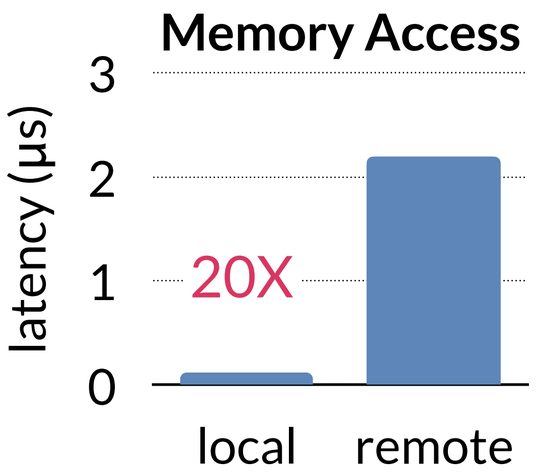
Locality is Important on Traversal

Remote access is much **slower** than local access (cross-node)

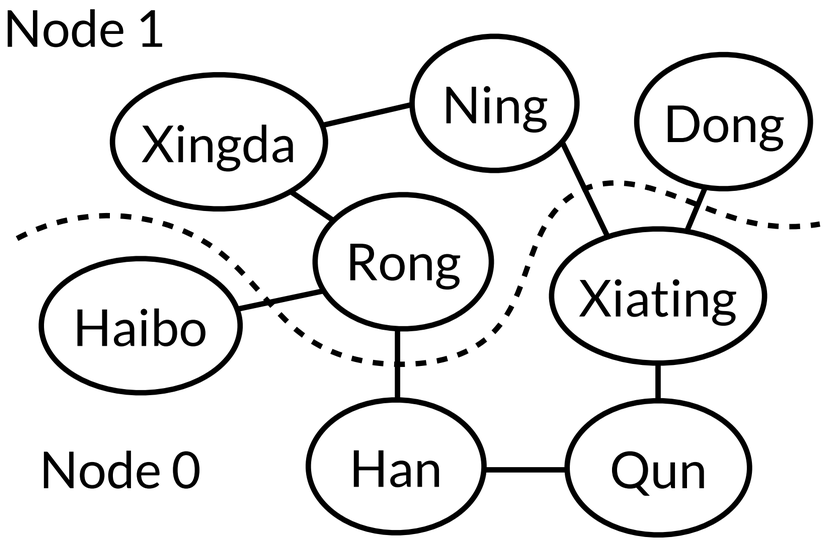


Locality is Important on Traversal

Remote access is much **slower** than local access (cross-node)

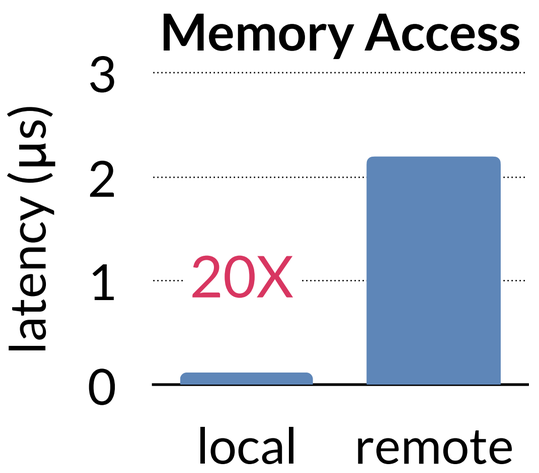


Who is friend of friend of *Haibo* ?

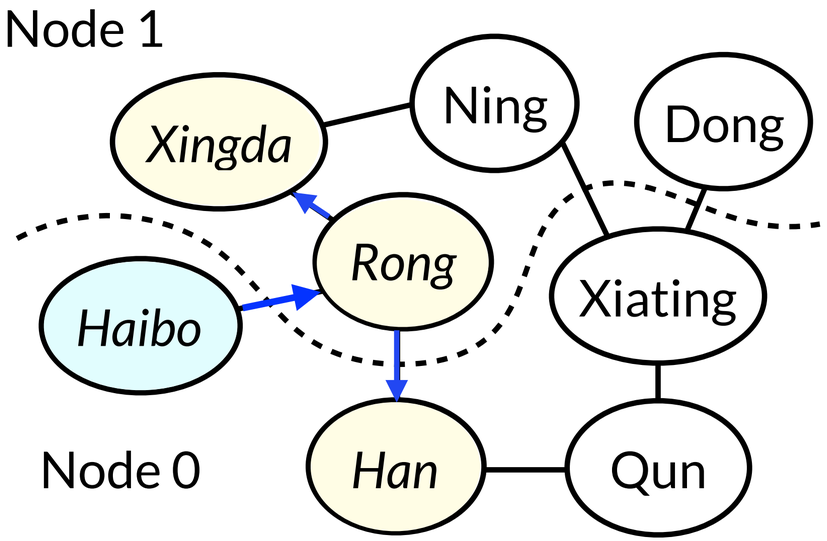


Locality is Important on Traversal

Remote access is much **slower** than local access (cross-node)

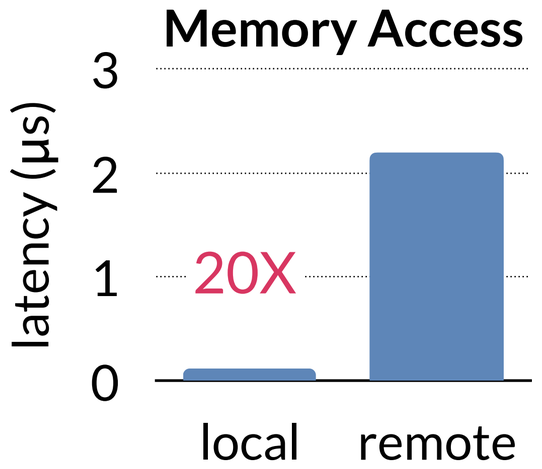


Who is friend of friend of *Haibo* ?

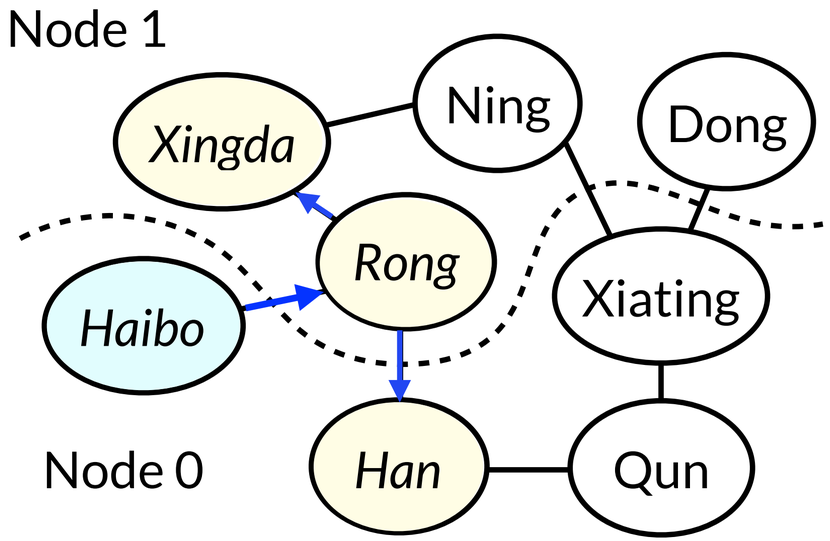


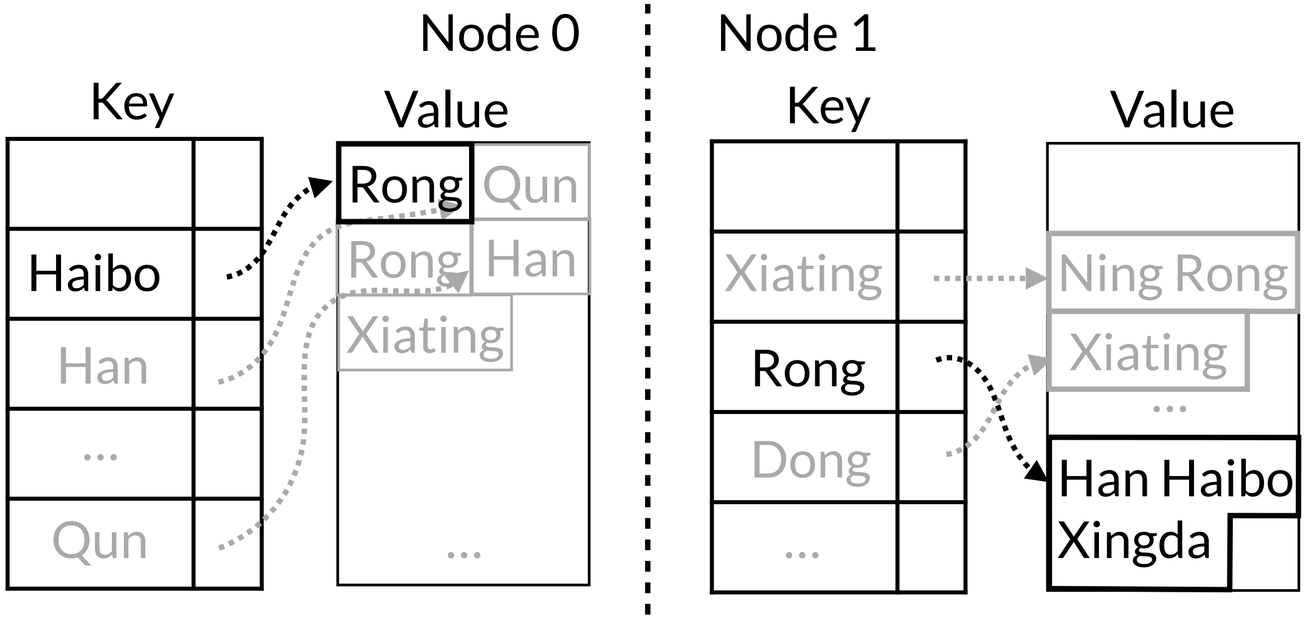
Locality is Important on Traversal

Remote access is much **slower** than local access (cross-node)



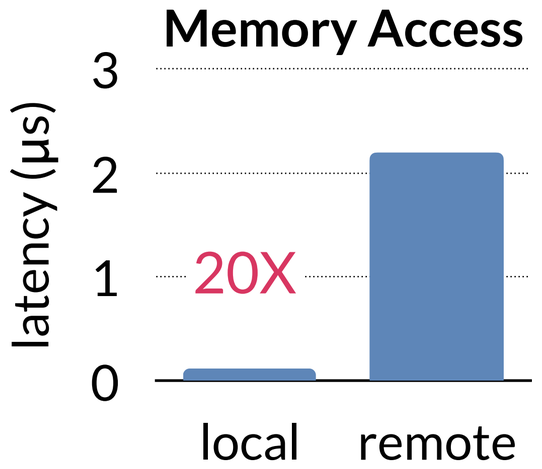
Who is friend of friend of *Haibo* ?



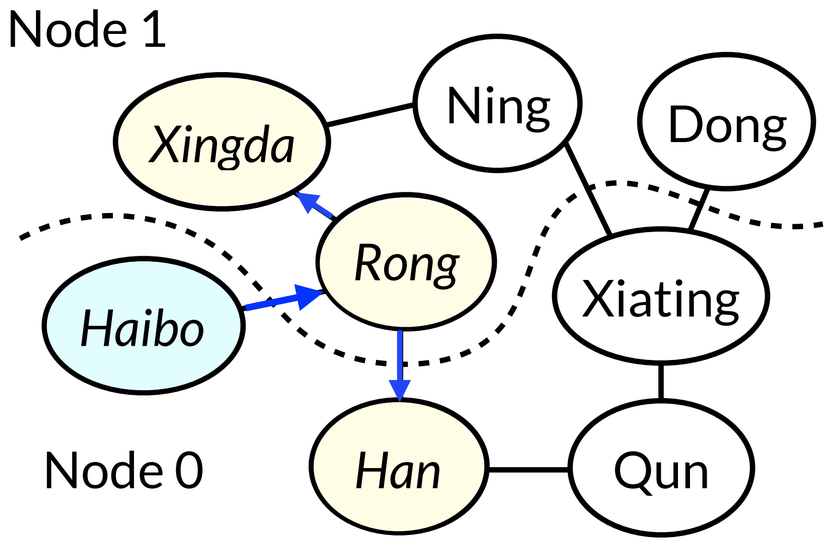


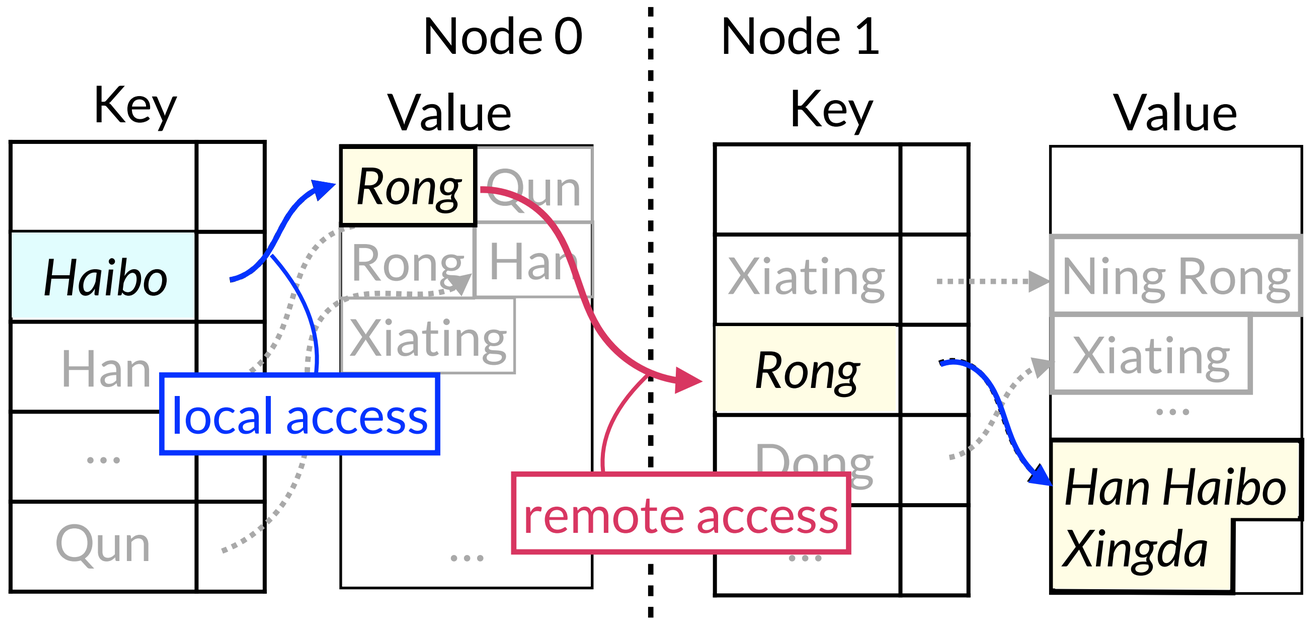
Locality is Important on Traversal

Remote access is much **slower** than local access (cross-node)



Who is friend of friend of *Haibo* ?

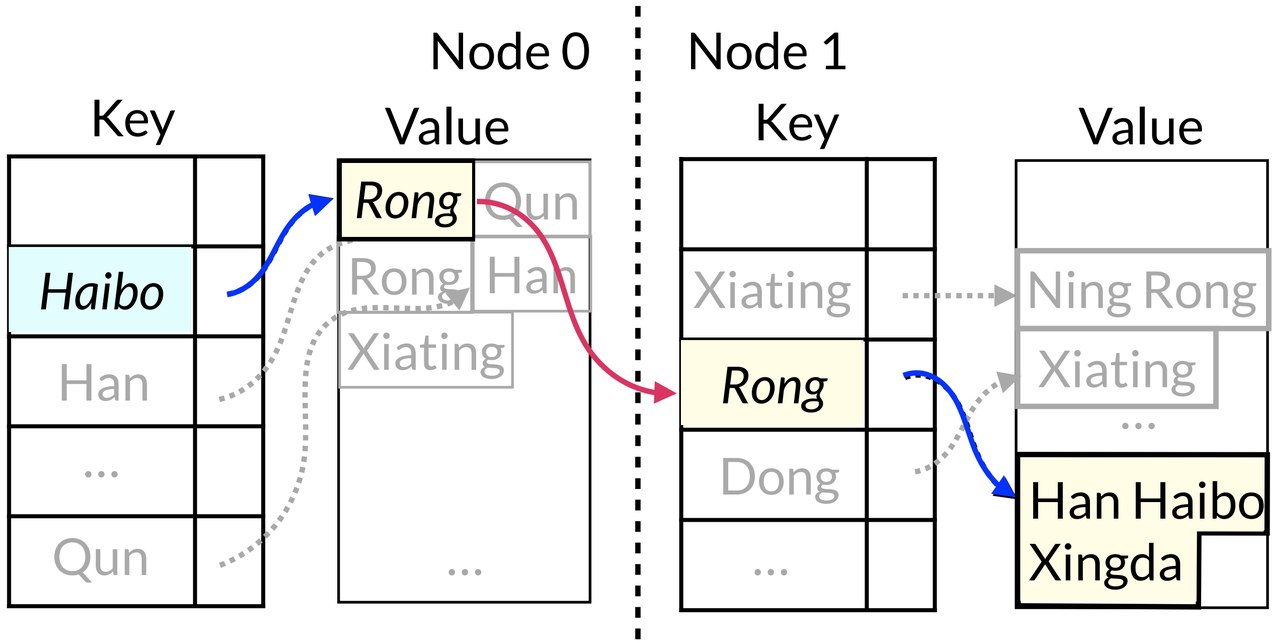




How to Preserve Locality? Live Migration

Goal: benefit <—> overhead

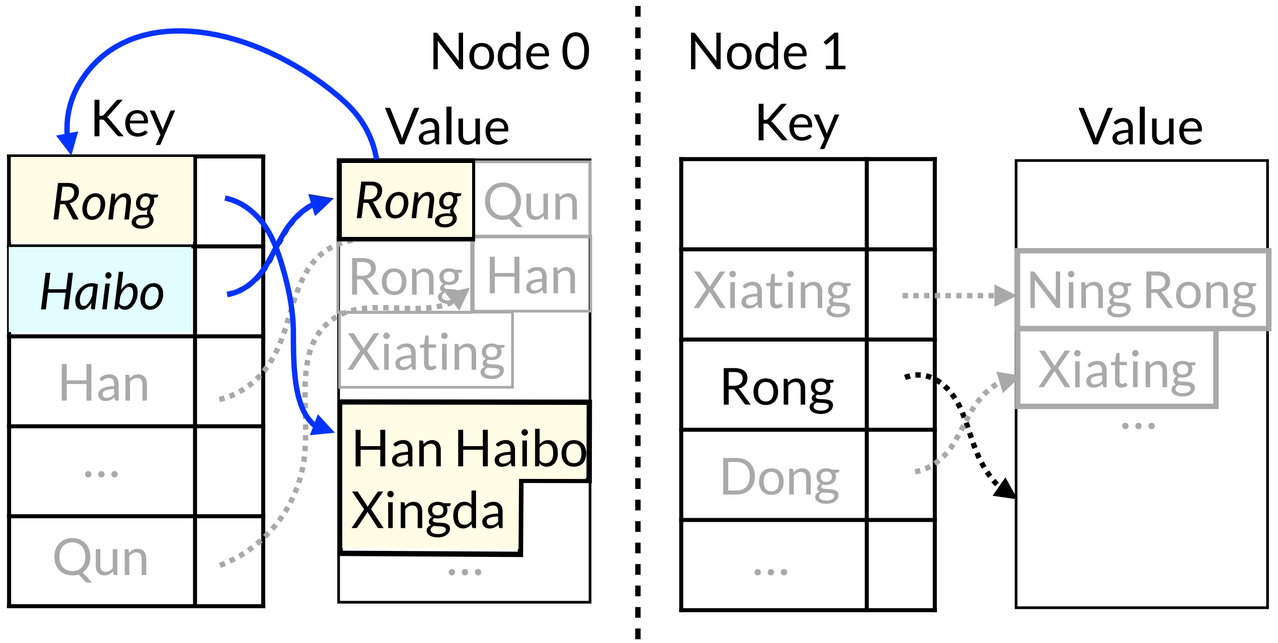
Traditional approach: *shard-based migration*



How to Preserve Locality? Live Migration

Goal: benefit <—> overhead

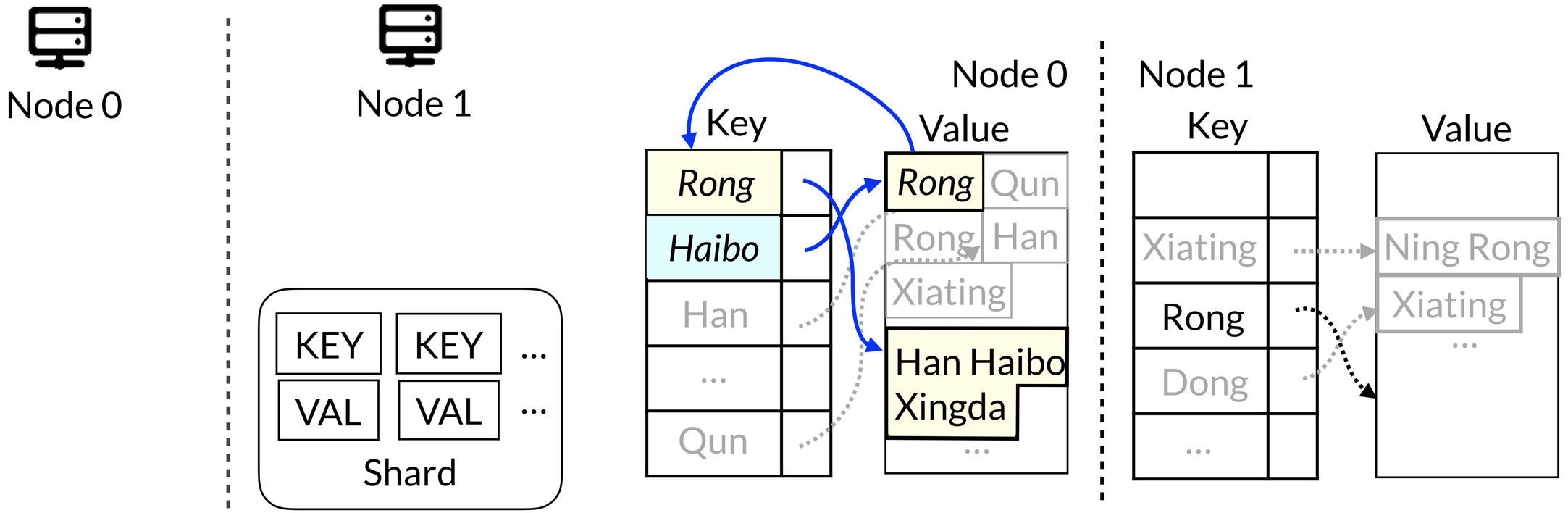
Traditional approach: *shard-based migration*



How to Preserve Locality? Live Migration

Goal: benefit <—> overhead

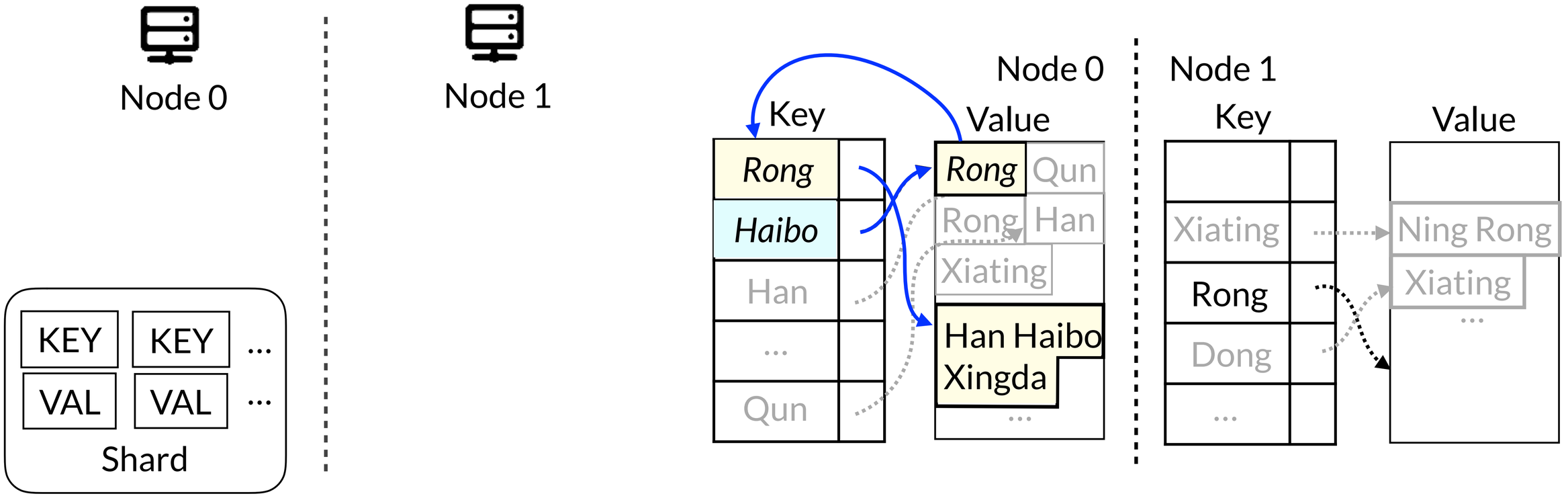
Traditional approach: *shard-based migration*



How to Preserve Locality? Live Migration

Goal: benefit <—> overhead

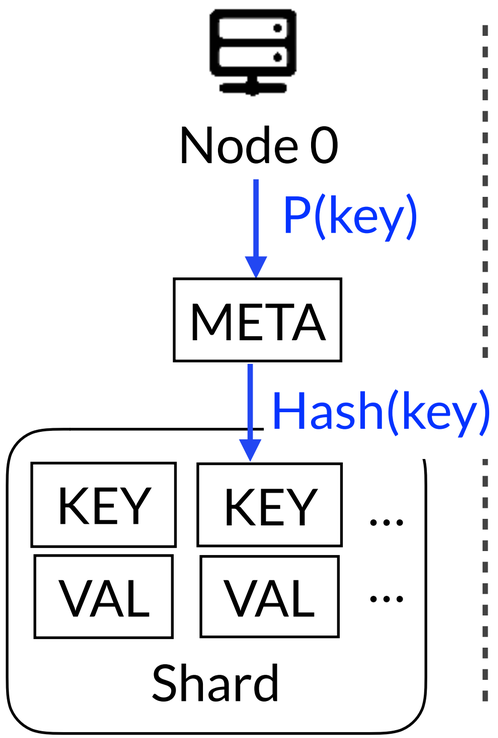
Traditional approach: *shard-based migration*

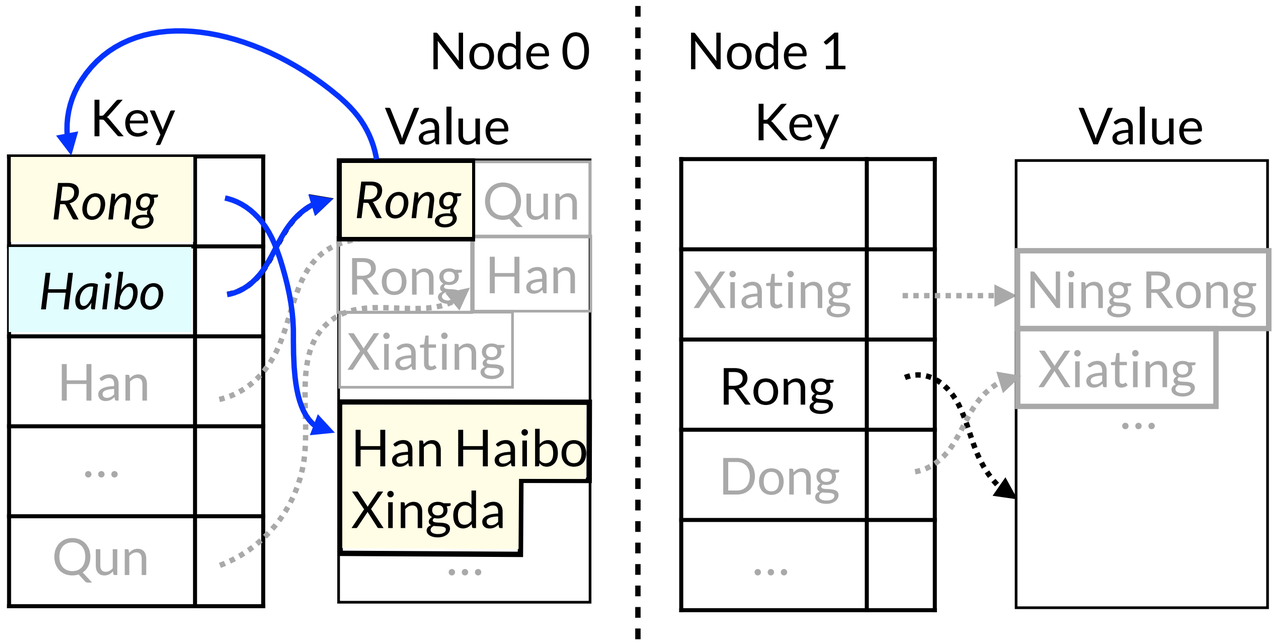


How to Preserve Locality? Live Migration

Goal: benefit <—> overhead

Traditional approach: *shard-based migration*

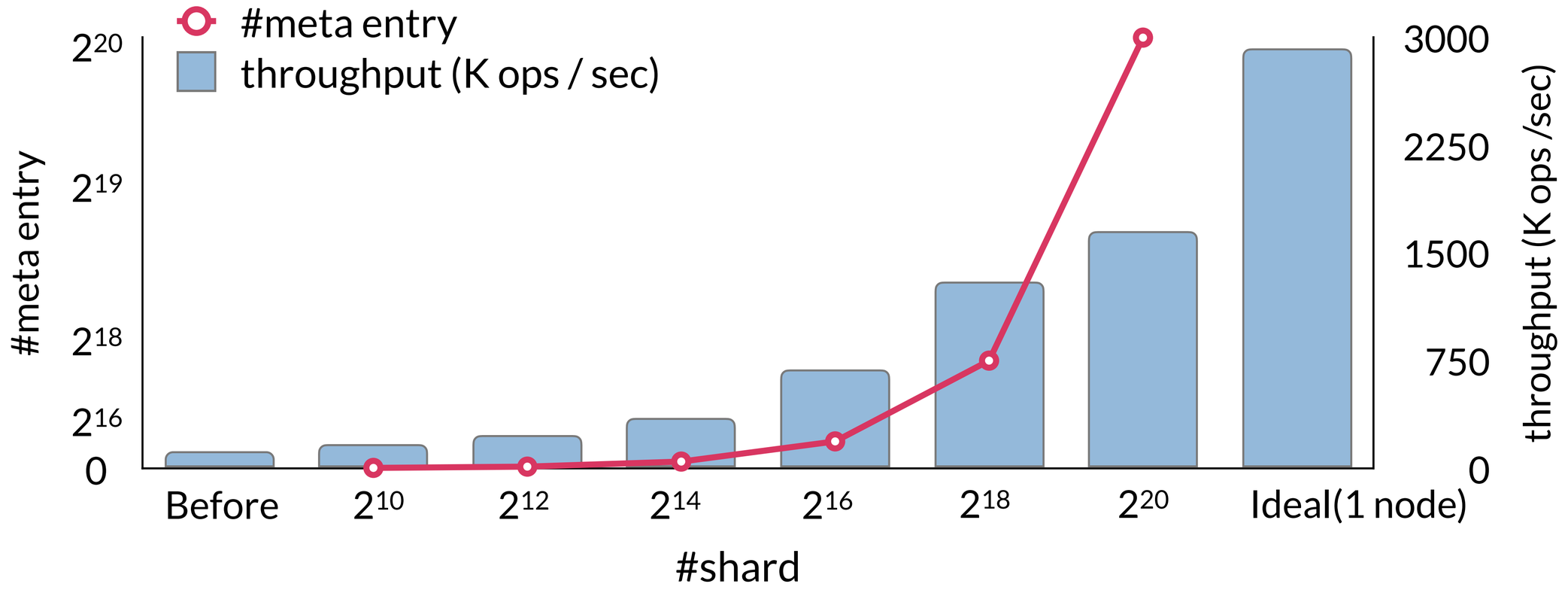




Shard-based Migration: Dilemma

graph	RMAT_26
#key value	54 M
#node	8

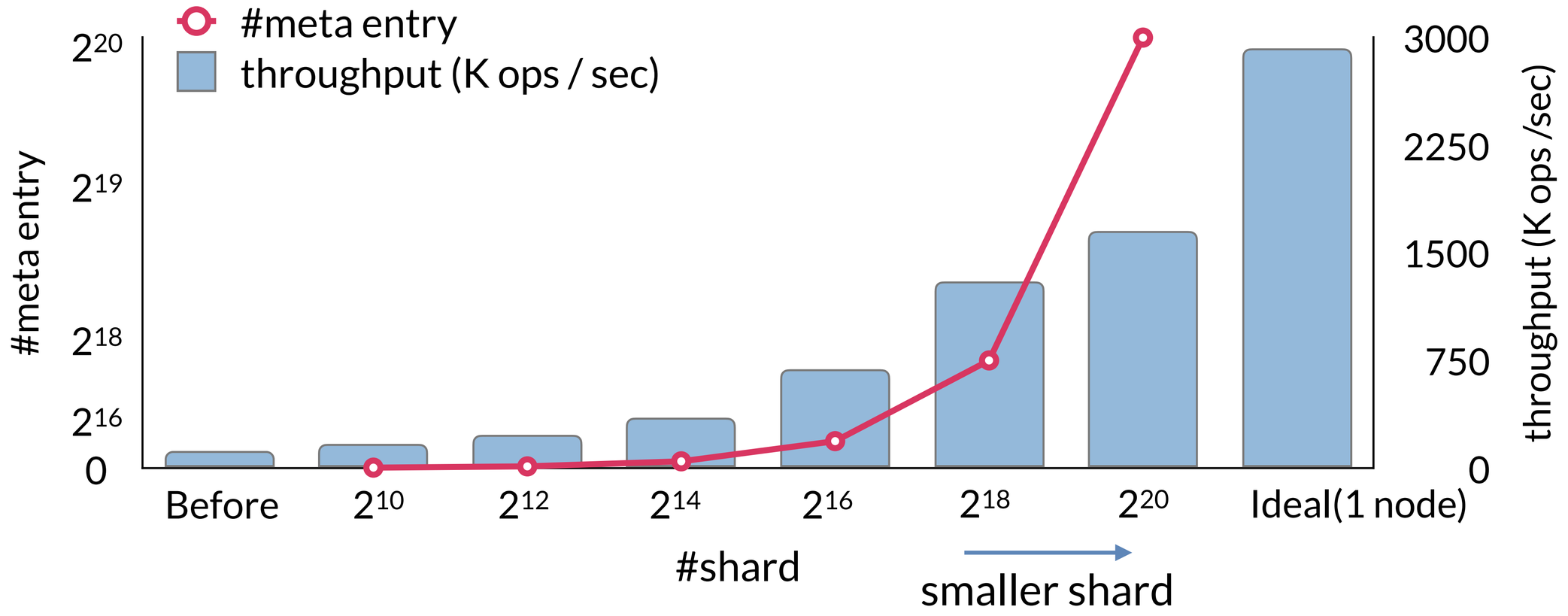
Each node has to cache **entire** metadata.



Shard-based Migration: Dilemma

graph	RMAT_26
#key value	54 M
#node	8

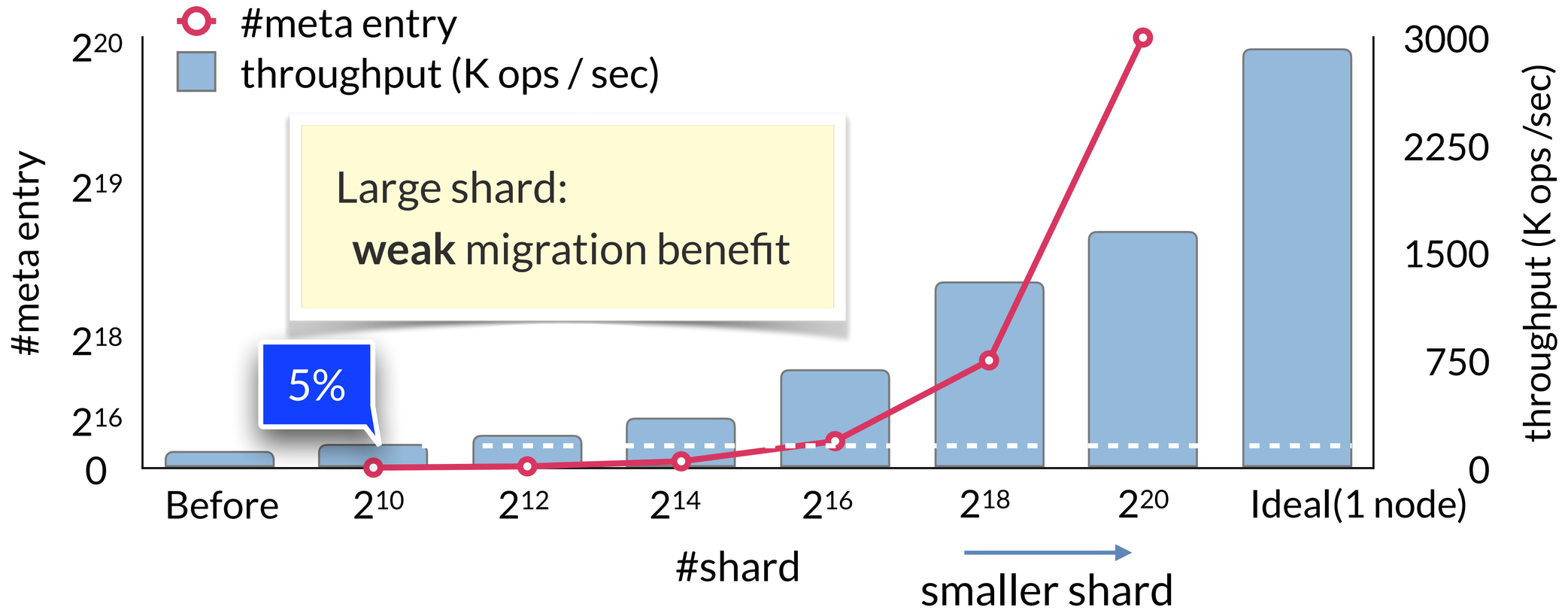
Each node has to cache **entire** metadata.



Shard-based Migration: Dilemma

graph	RMAT_26
#key value	54 M
#node	8

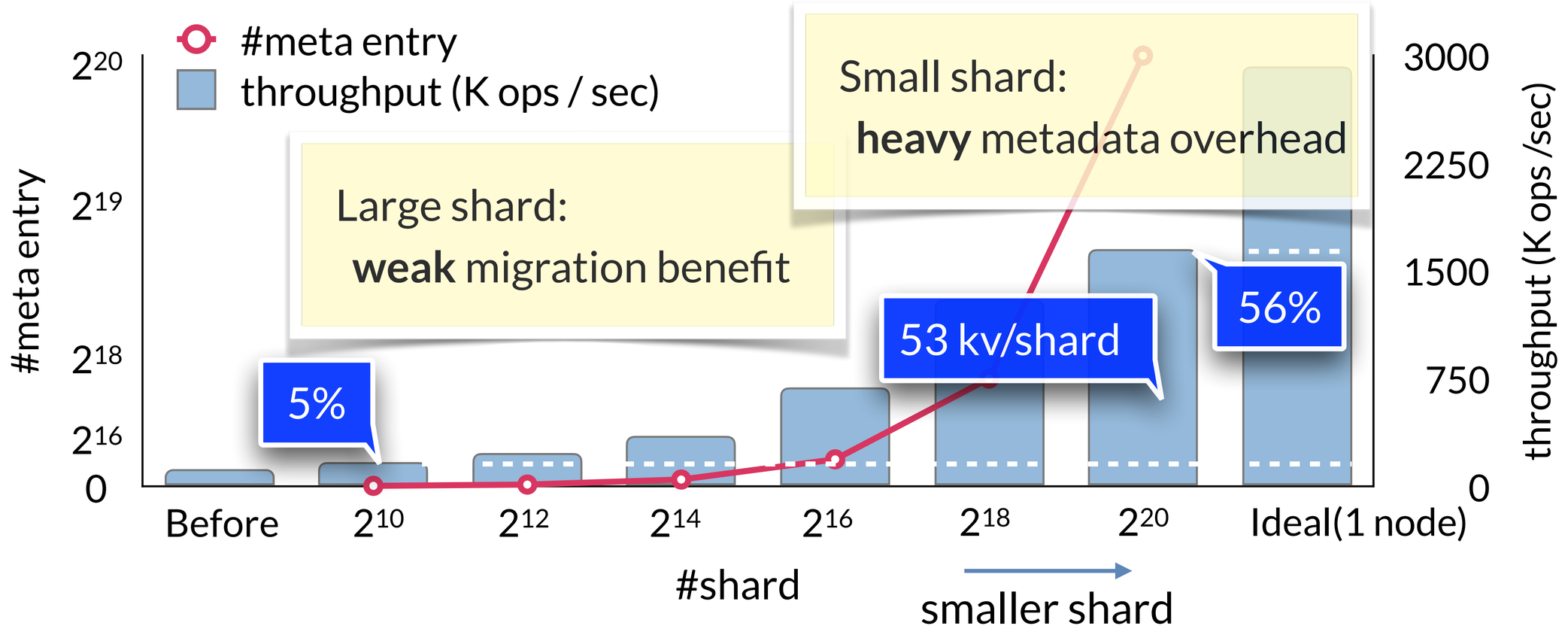
Each node has to cache **entire** metadata.



Shard-based Migration: Dilemma

graph	RMAT_26
#key value	54 M
#node	8

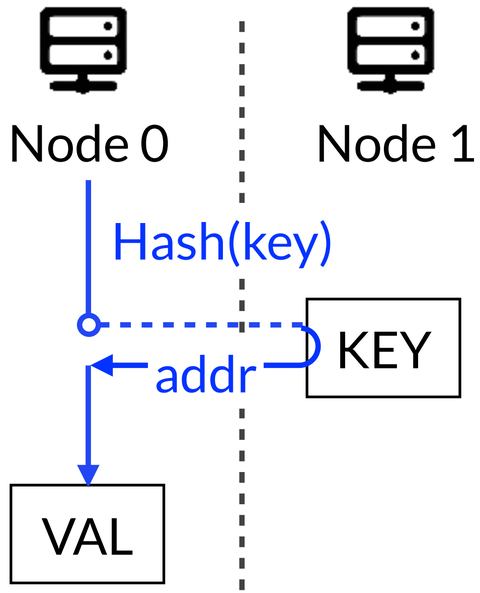
Each node has to cache **entire** metadata.



Design Overview

Pragh uses ***split live migration*** for graph traversal.

Split live migration: migrates values while keeps keys stationary

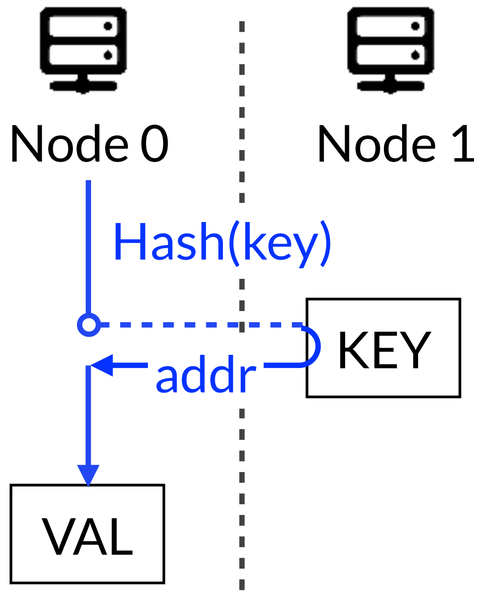


Design Overview

Pragh uses ***split live migration*** for graph traversal.

Split live migration: migrates values while keeps keys stationary

- Great migration benefits
- No metadata overhead

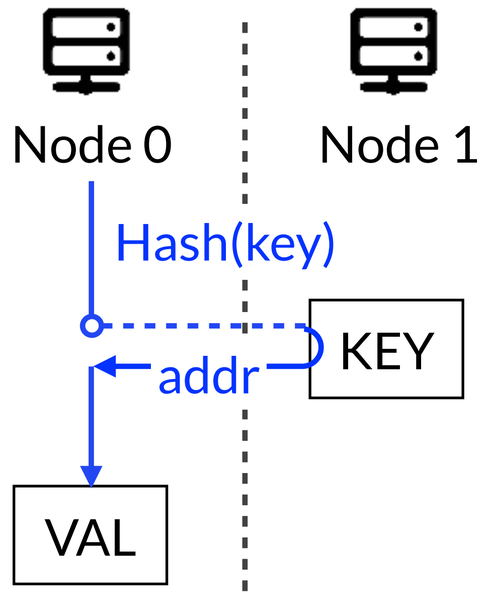


Design Overview

Pragh uses ***split live migration*** for graph traversal.

Split live migration: migrates values while keeps keys stationary

- Great migration benefits
- No metadata overhead
- Throughput: **19X**
- Eliminate remote access: **97%**
- Port to Wukong (OSDI'16): **2.53X**



Design Overview

Pragh uses ***split live migration*** for graph traversal.

Split live migration: migrates values while keeps keys stationary

How to achieve fully-localized split migration ?

How to migrate over evolving graphs?

How to support fine-grained and lightweight monitoring ?

- Port to Wukong (OSDI'16): **2.53X**

VAL

IPADS
INSTITUTE OF PARALLEL
AND DISTRIBUTED SYSTEMS

ATC 2019, 4:35 PM, Track II, on July 11th