UCI

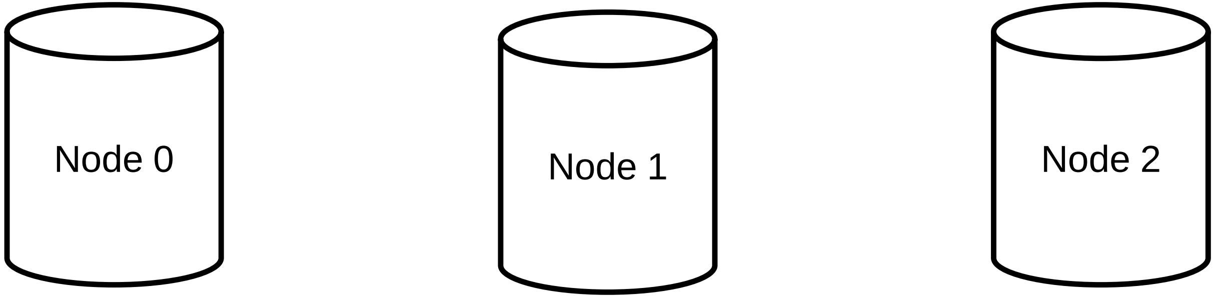
Improving Data Ingestion Performance in Apache AsterixDB

Qiyang He
Southern University of Science and Technology

Outline

- Introduction
- Benchmark

Use Case

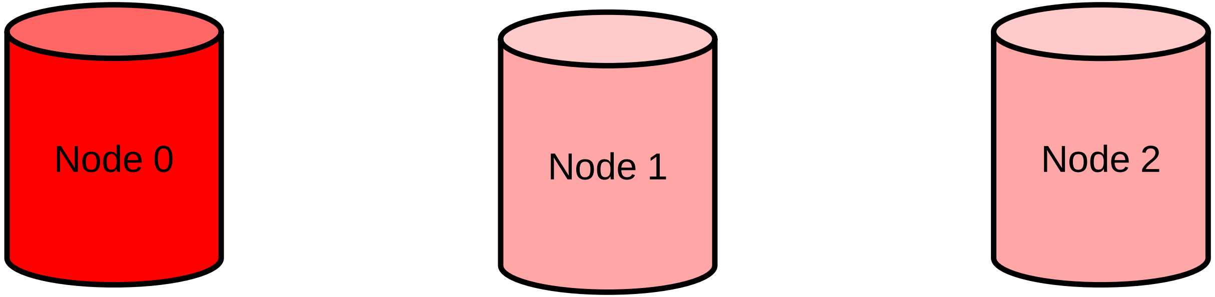


Node 0

Node 1

Node 2

Use Case



Node 0

Node 1

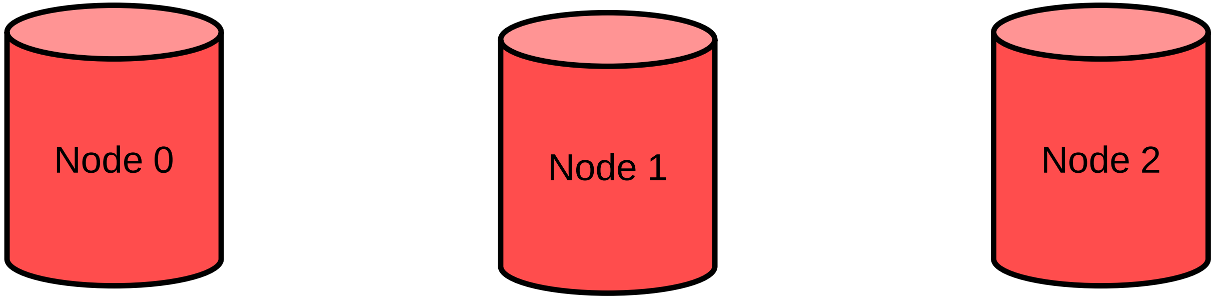
Node 2

Parse: 100%
Store: 33.3%

Parse: 0%
Store: 33.3%

Parse: 0%
Store: 33.3%

Use Case



Node 0

Node 1

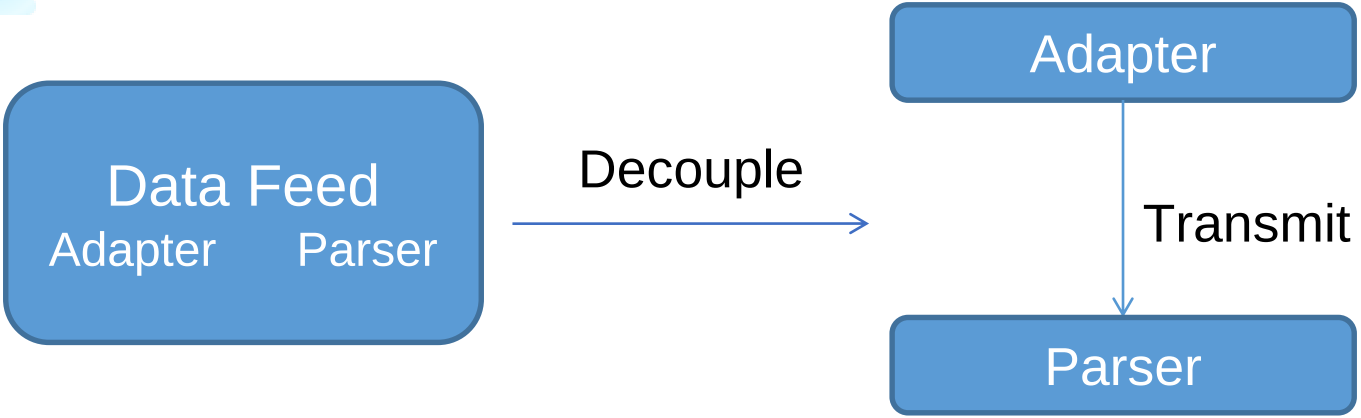
Node 2

Parse: 33.3%
Store: 33.3%

Parse: 33.3%
Store: 33.3%

Parse: 33.3%
Store: 33.3%

My work



Data Feed

Adapter Parser

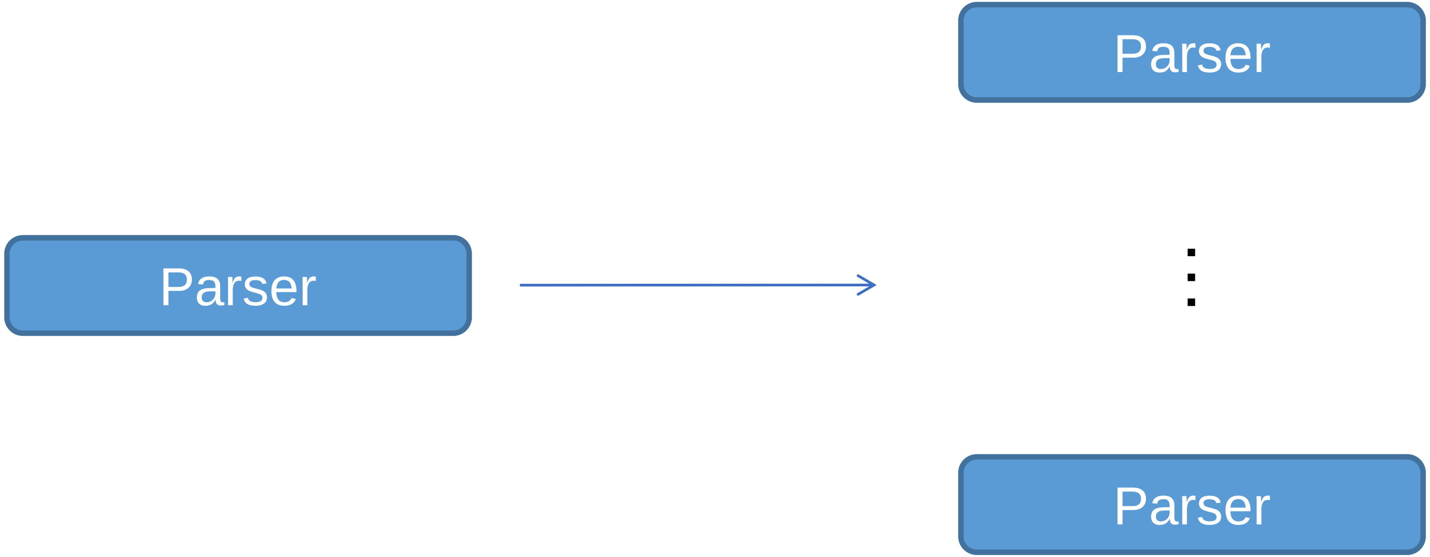
Decouple

Adapter

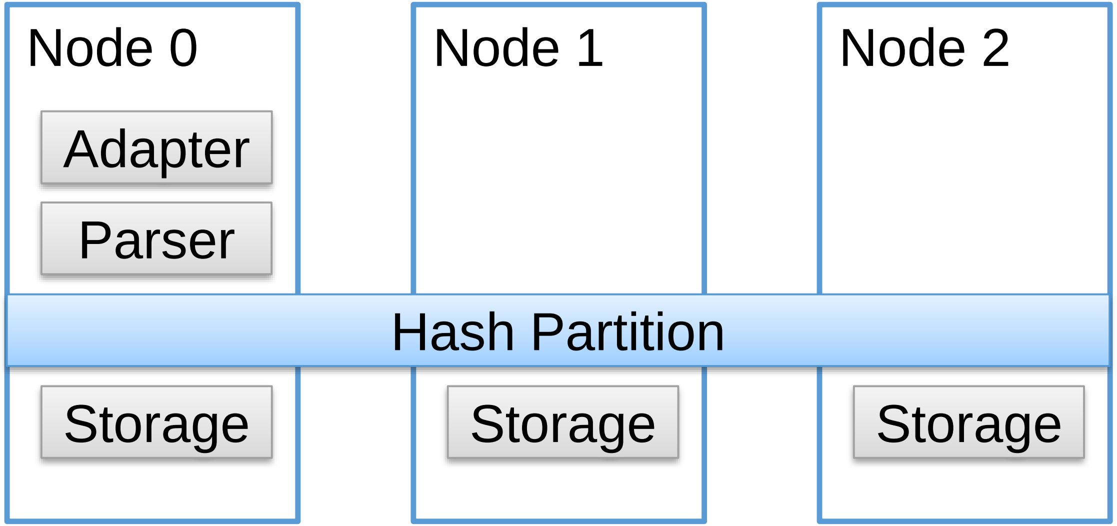
Transmit

Parser

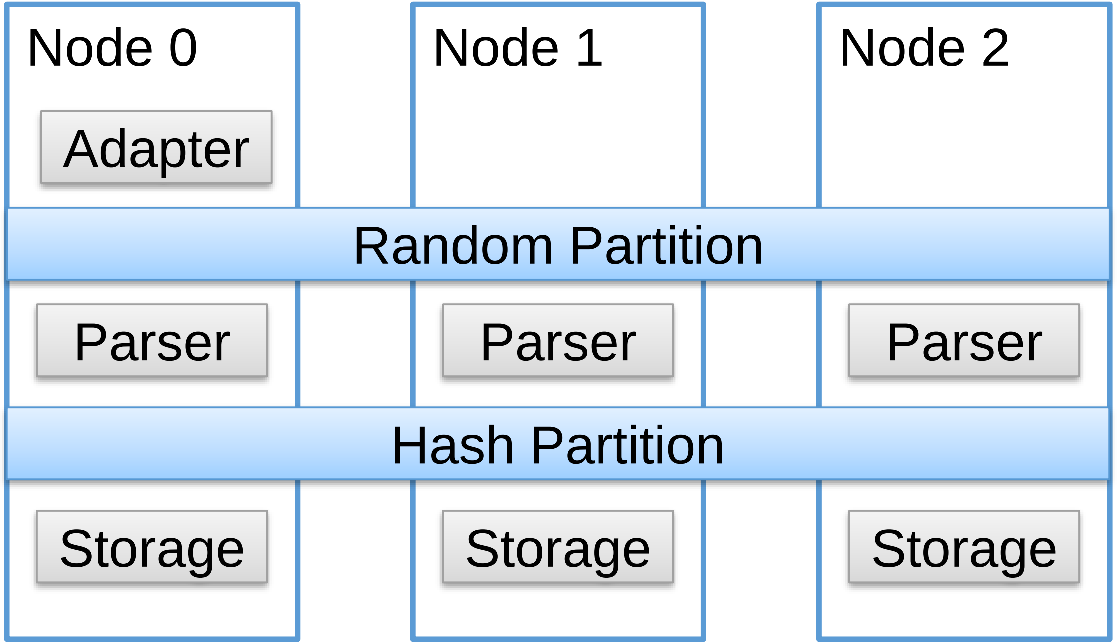
My work



Existing Framework



Proposed Framework



Restriction

Primary Key

1. {“name”:”Alice”,”friends_count”:18,”followers_count”:4941}
2. {“name”:”Bob”,”friends_count”:445,”followers_count”:22649}
3. {“name”:”Alice”,”friends_count”:18,”followers_count”:4}
4. {“name”:”Bob”,”friends_count”:455,”followers_count”:22649}

Parallel Ingestion:

1 and 2
1 and 4
3 and 2
3 and 4

Experiment Settings

- Processor: i7-5575 CPU @ 3.30GHz (2 cores)
- Memory: 16GB
- Dataset: 10M tweets (each about 377Bytes)

Benchmark

2 Nodes:

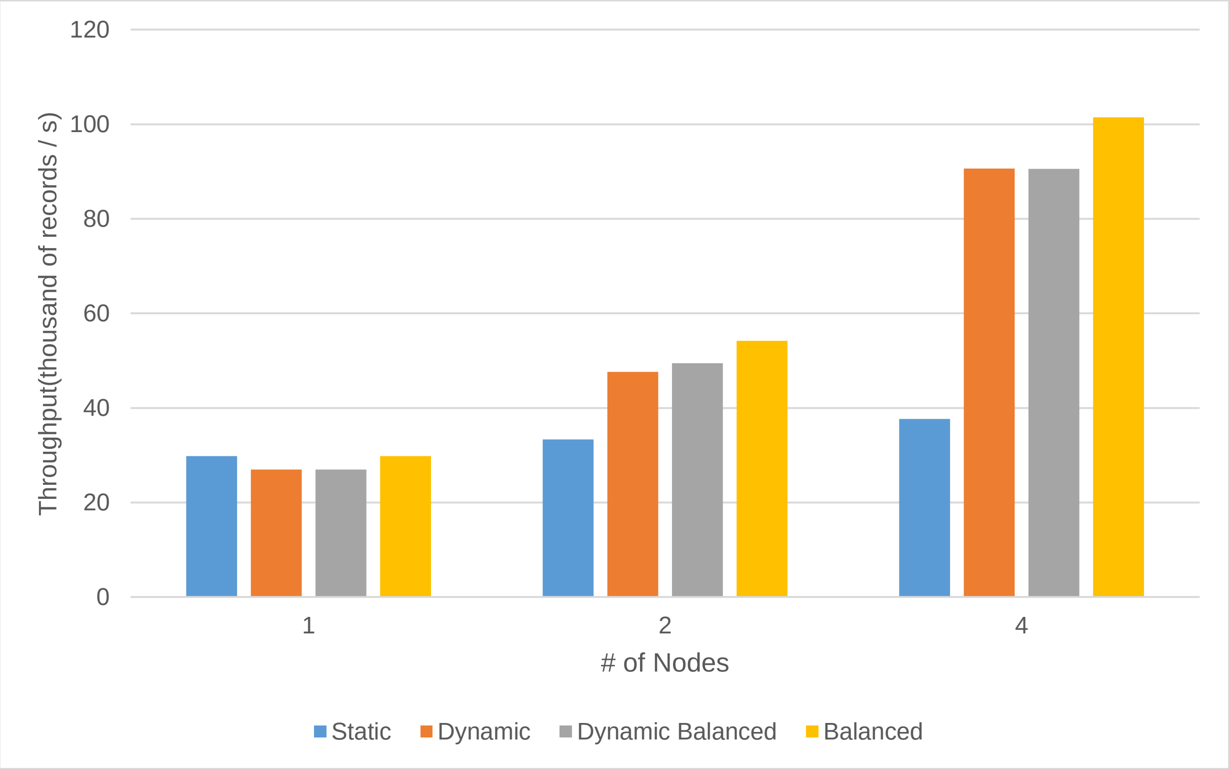
+42.8%   Current Framework

-12.2%   Ideal Framework

4 Nodes:

+140.9%   Current Framework

-10.7%   Ideal Framework



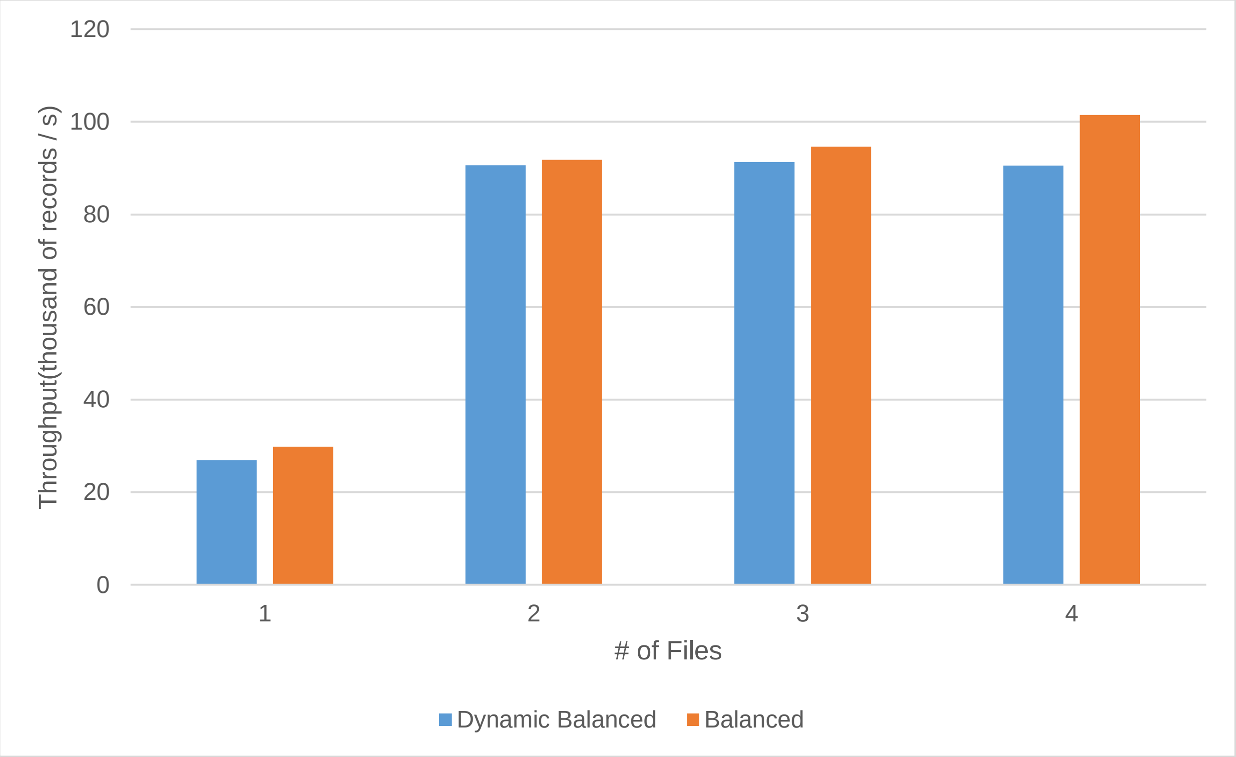
Benchmark

Dynamic:

No difference after 2 files

Balanced:

Tiny improvement after 2 files



Thank you