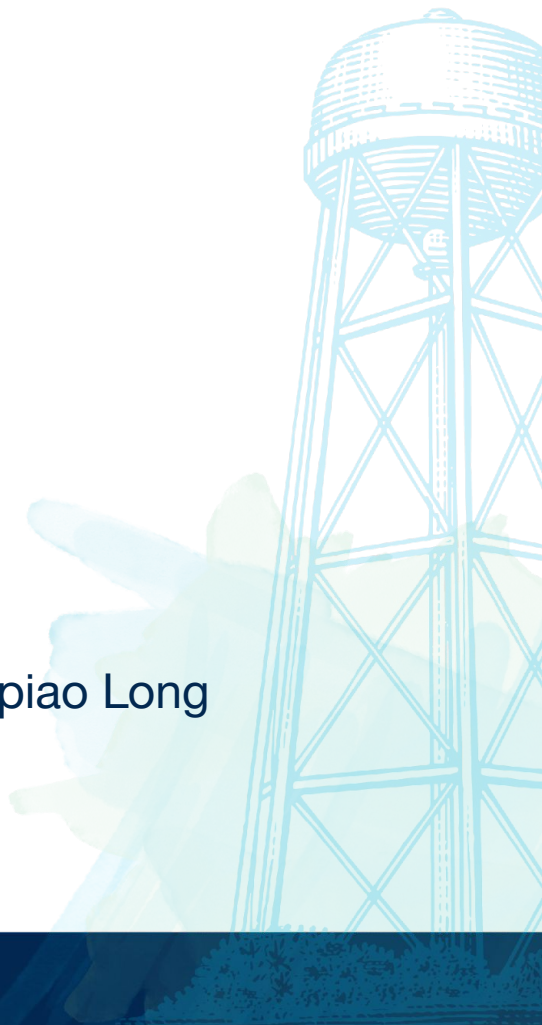


# Final Report

## RCC in NexRes

Presenter:

Dakai Kang, Musheng He, Zizhong Li, Xiaoxing Chen, Piaopiao Long



# Content

- **Problem Statement**
- **Implementation**
- **Experiments and Results**



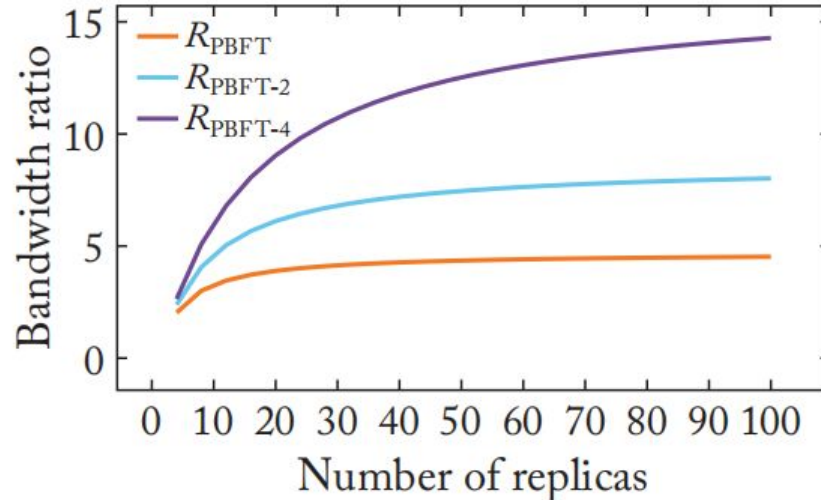
# Part 1

# Problem Statement



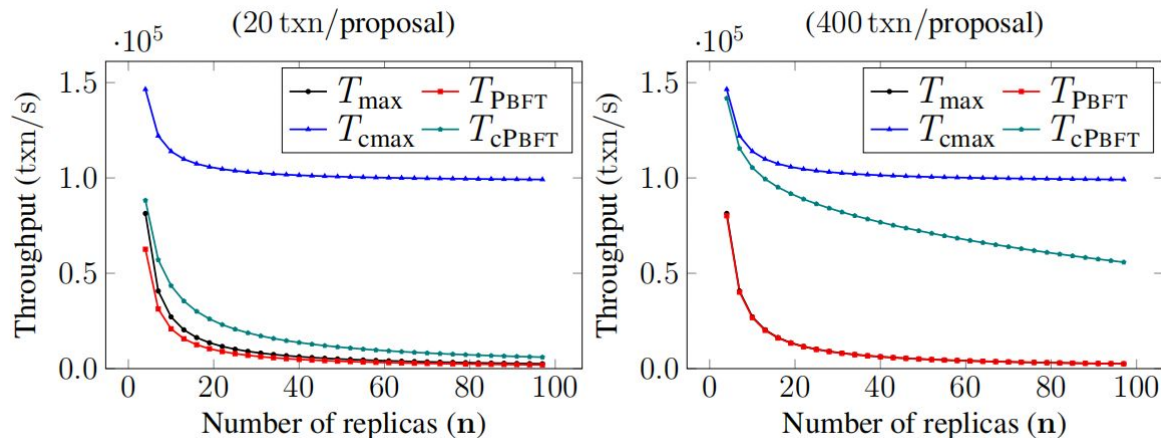
# Problem Statement - RCC

- In primary-backup protocols, like PBFT, primary broadcasts client requests.
- Unbalanced overhead of sending messages.
- Bottleneck: Outgoing bandwidth of Primary



# Problem Statement - RCC

- Distribute the overhead of broadcasting client requests
- Running concurrent instances



# Problem Statement - ResDB vs NexRes

- ResDB assigns only one thread to one instance.
- The actual bottleneck of PBFT in ResDB: **CPU utilization.**
  
- NexRes assigns multiple threads to one instance.
- NexRes is able to saturate the primary's network bandwidth in PBFT.
- NexRes meets the prerequisite for leveraging RCC.

# Part 2

# Implementation



# Implementation

- **Configuration**

- Specify the number of instances and the primaries

- **Message**

- *instance*: identify the instance of a proposal

- **Client**

- Send client requests to all primaries evenly



# Implementation

- **Primary**

- indicate its instance when broadcasting proposals

- **Backup Replica**

- When receiving a proposal of instance  $i$ , check if it is exactly from the primary of instance  $i$

# Implementation

- **Total Ordering**

- Before executing committed client requests, RCC orders the committed requests between different instances in the same round.
- Simplest one: from the lowest instance to the highest one.

# Part 3

## Experiments and Results



# Experiments and Results

- **Setup**

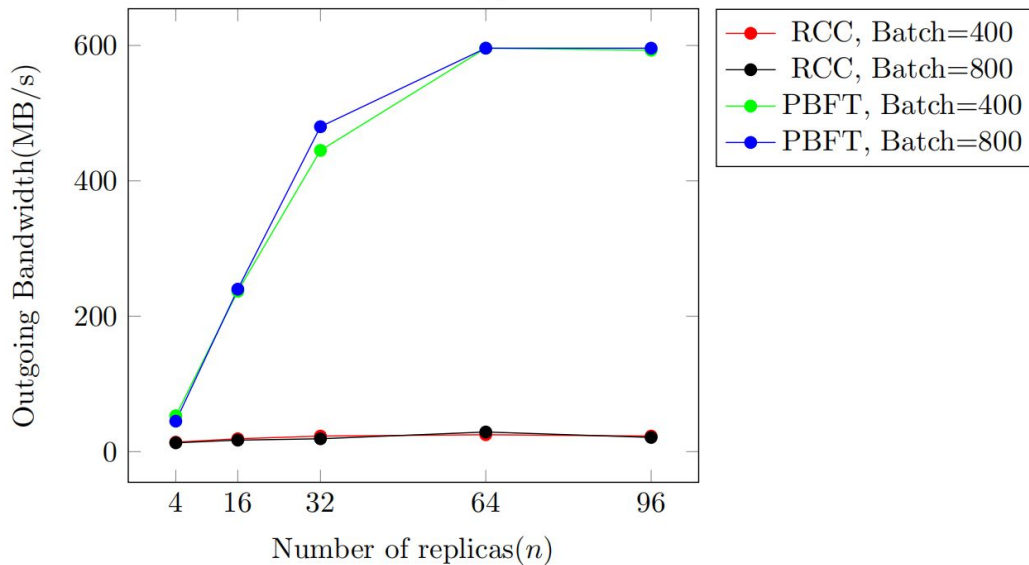
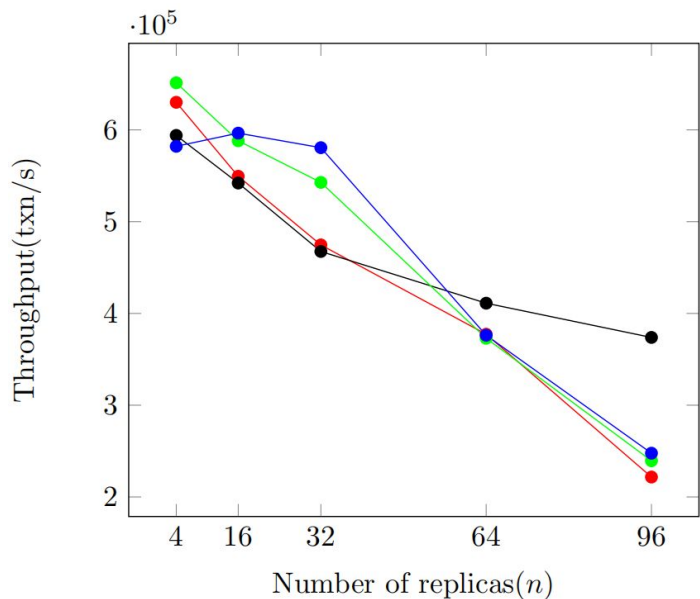
- AWS EC2
- Instance Type: t3.2xlarge; 8-vCPUs; 32-GiB Memory
- Up to 96 replicas and 4 clients

# Experiments and Results

- **Experiment 1 - Scalability**

- Test and record the throughput and outgoing bandwidth of RCC and PBFT with different number of replicas.
- The number of replicas is set to be 4, 16, 32, 64 and 96;
- the number of batch size is set to be 400 and 800.

# Experiments and Results

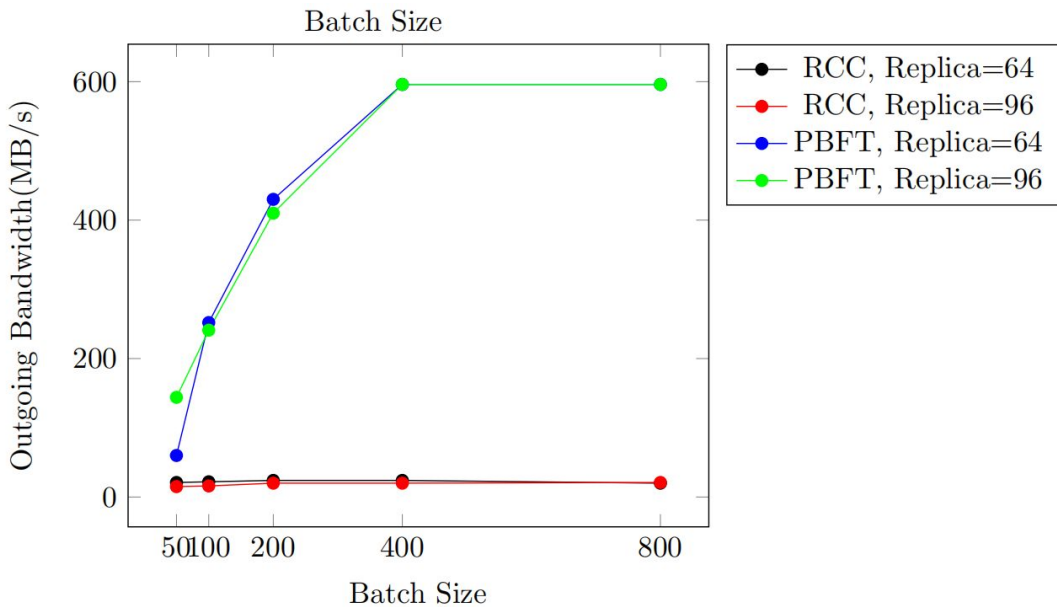
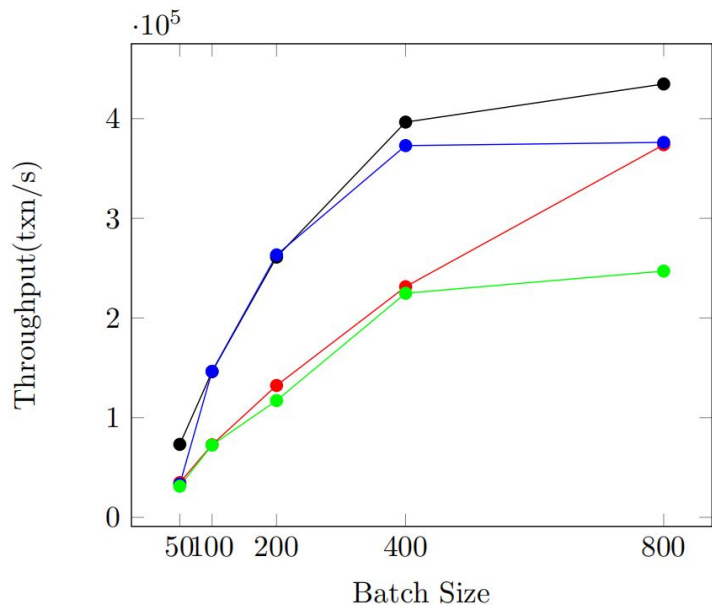


# Experiments and Results

- **Experiment 2 - Batching**

- Test and record the throughput and outgoing bandwidth of RCC and PBFT of different batchsize.
- The number of batch size is be set to 50, 100, 200, 400 and 800; the number of replicas is set to be 64 and 96.

# Experiments and Results



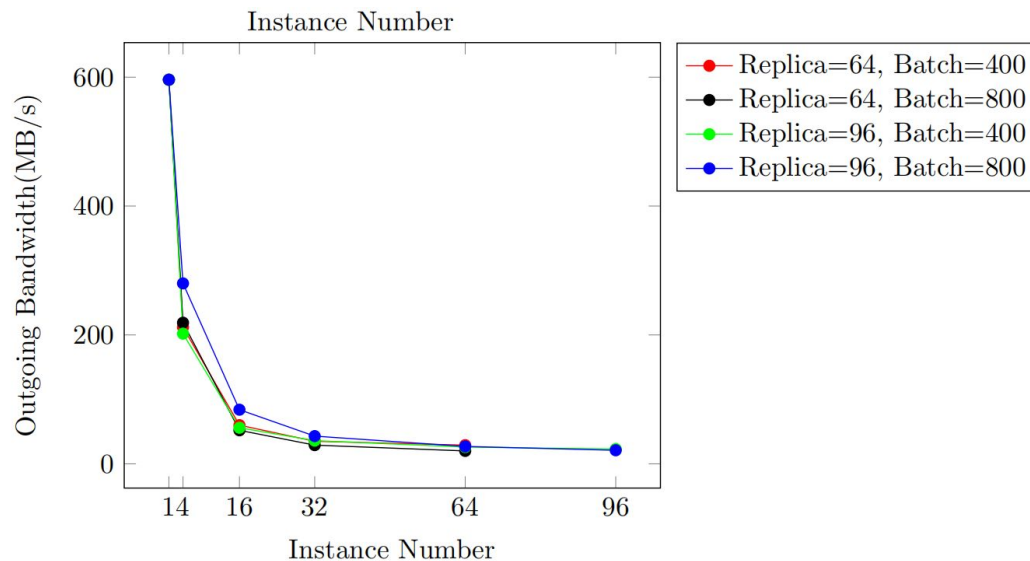
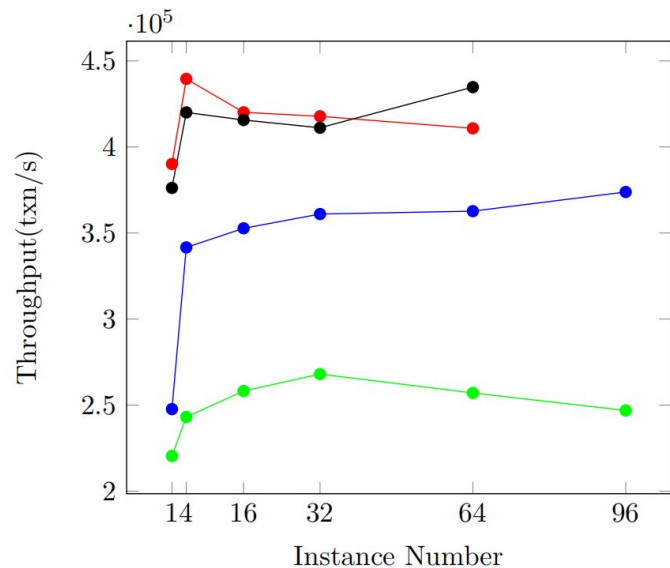


# Experiments and Results

- **Experiment 3 - Concurrency**

- Test and record the throughput and outgoing bandwidth of RCC with 64 or 96 replicas with different numbers of instances
- The number of batch size is set to 400 and 800; the number of replica is set to be 64 and 96.

# Experiments and Results



# Final Report

## RCC in NexRes

### Q & A

Presenter:

Dakai Kang, Musheng He, Zizhong Li, Xiaoxing Chen, Piaopiao Long