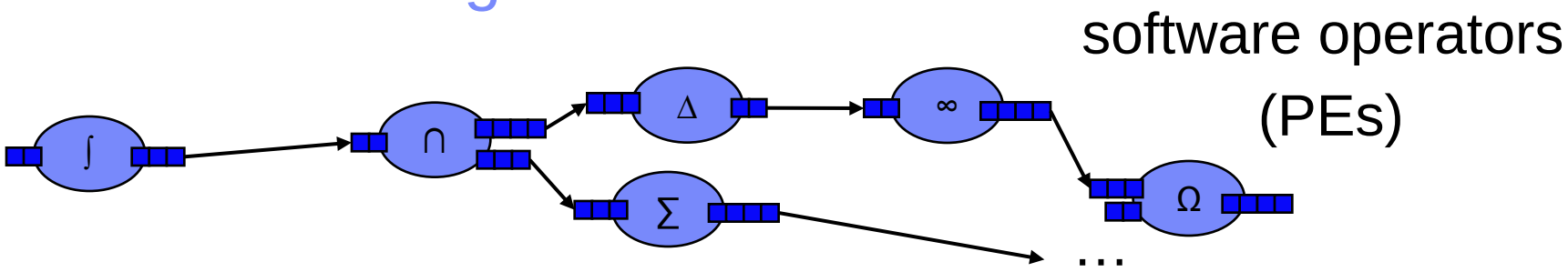
IBM Research

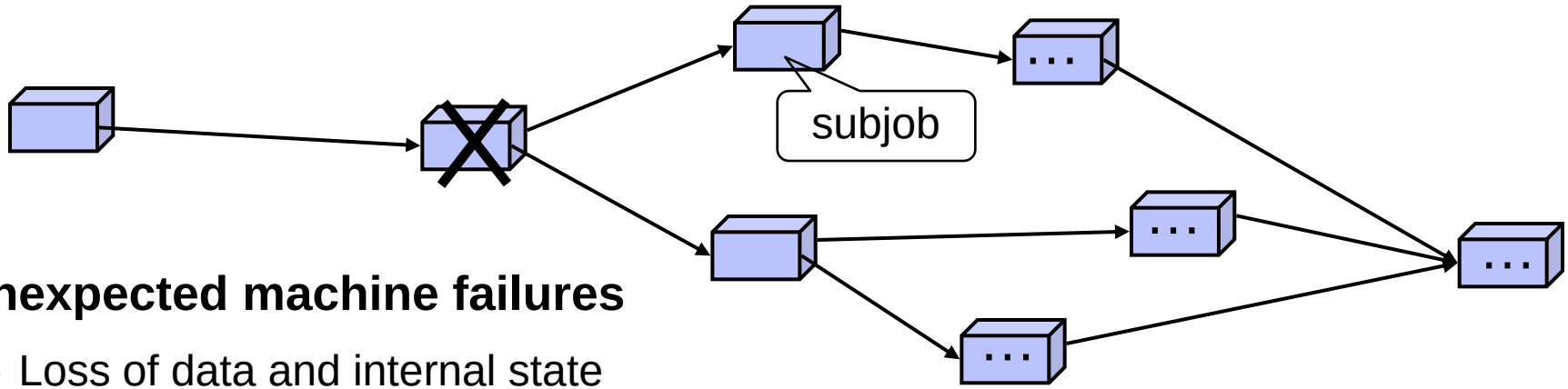
An Empirical Study of High Availability in Stream Processing Systems

Yu Gu, *Zhe Zhang*, Fan Ye, Hao Yang,
Minkyong Kim, Hui Lei, Zhen Liu

12/3/2009

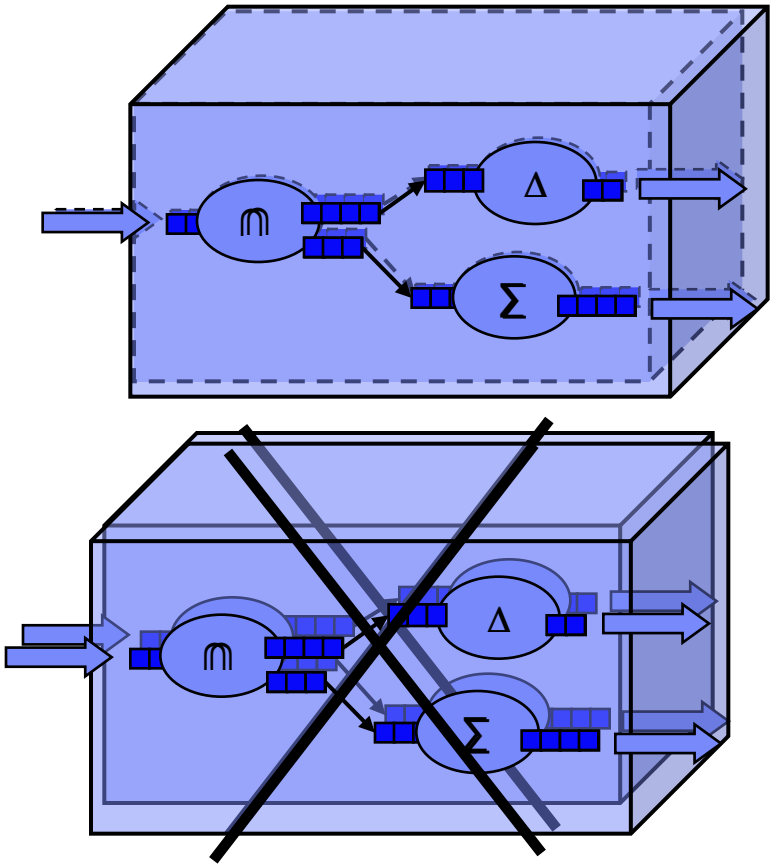
Stream Processing Model





- **Unexpected machine failures**
  - Loss of data and internal state
  - Disruption to normal processing
- **Challenge: how to preserve data / state and minimize disruption?**

Existing approaches:
Active Standby vs. Passive Standby



Active Standby

Passive Standby

Basic Tradeoff between AS and PS

- **Active Standby**
  - Overhead: double processing load; at least double message load
  - Recovery delay: almost zero
- **Passive Standby**
  - Overhead: checkpoint messages
  - Recovery delay: failure detection + deploy new job + recover state

Motivation

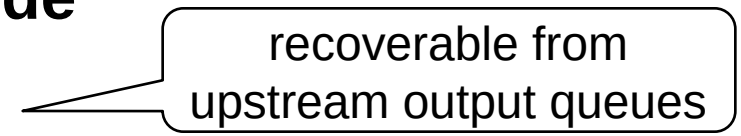
- **Tradeoffs of AS & PS not fully understood**
  - Only systematic comparison: [Hwang ICDE05]
    - Used a variant of PS with high overhead
    - Evaluated in simulations rather than real systems
- **Our contributions**
  - A *sweeping* checkpointing method
    - Reducing checkpoint overhead by one order of magnitude
    - Proof of consistency
  - A real prototype distributed stream processing system
  - Comprehensive and empirical evaluation of AS and PS

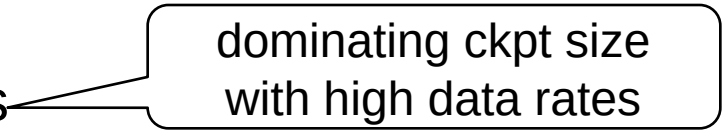
Outline

- Background and Motivation
- **Design and Implementation**
  - **Sweeping Checkpointing**
  - **System Architecture**
- Performance Evaluation
- Related Work
- Conclusions

Overview of Sweeping Checkpointing

- **What to include**
  - Internal states
  - Output queues
- **When to trim**
- **Checkpointing Multiple PEs**
- **Proof of consistency**



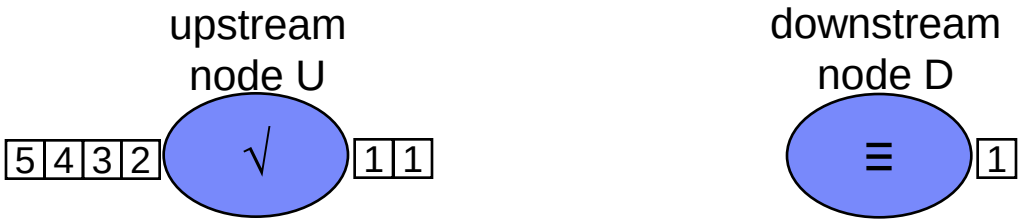


When to Trim

In U's output queue, only removing those packets that have been processed and checkpointed by D

When to Trim

In U's output queue, only removing those packets that have been processed and checkpointed by D

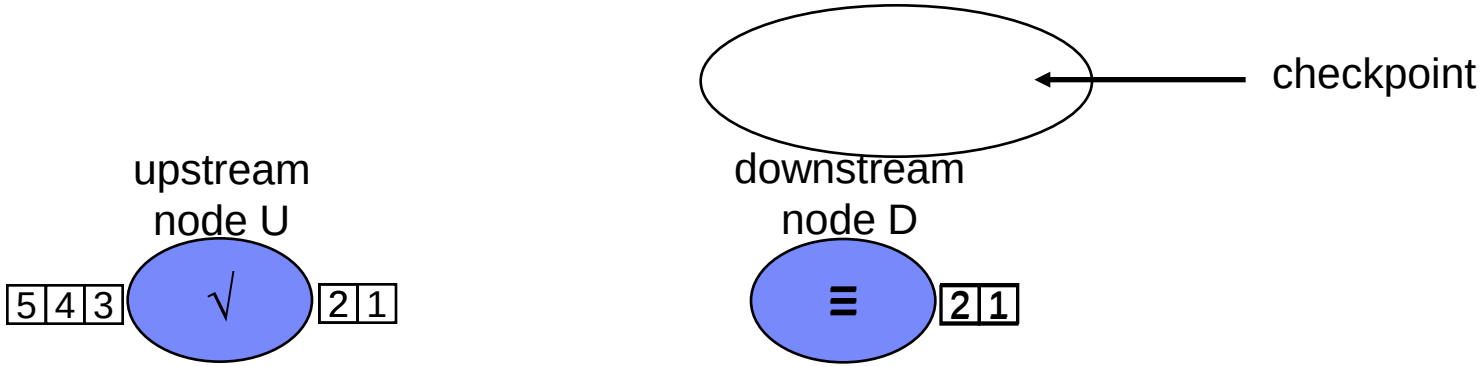


When to Trim

In U's output queue, only removing those packets that have been processed and checkpointed by D

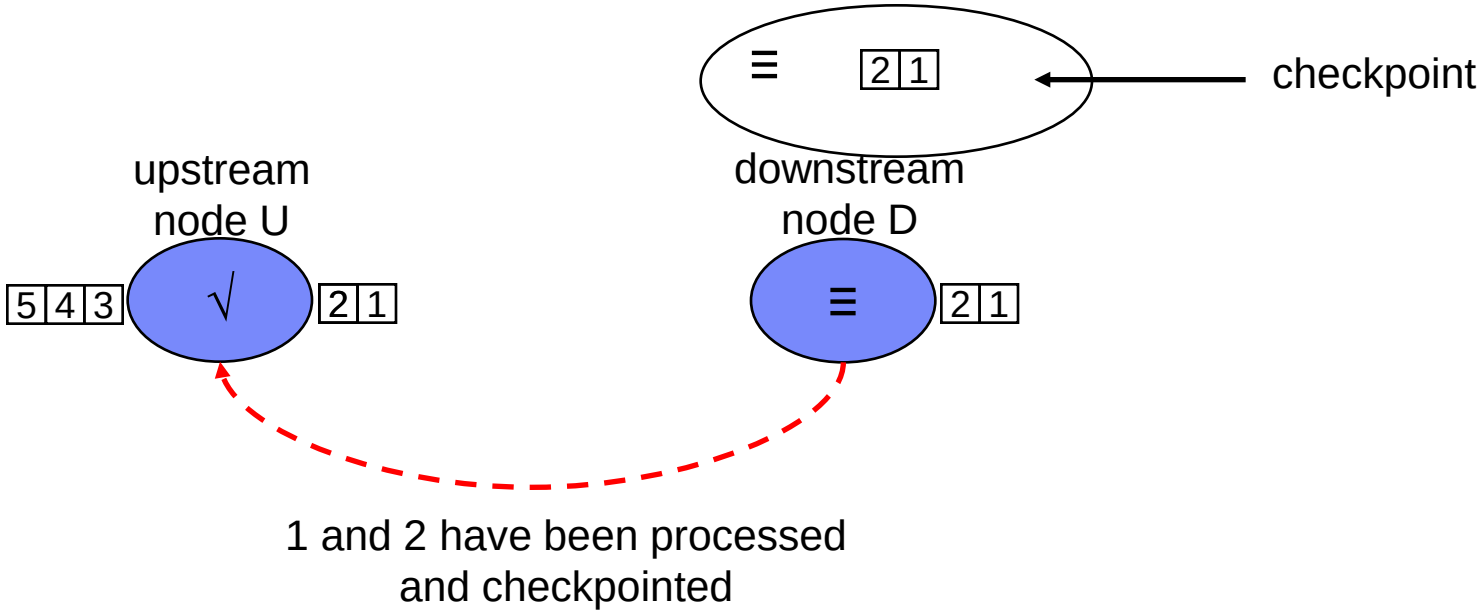
When to Trim

In U's output queue, only removing those packets that have been processed and checkpointed by D



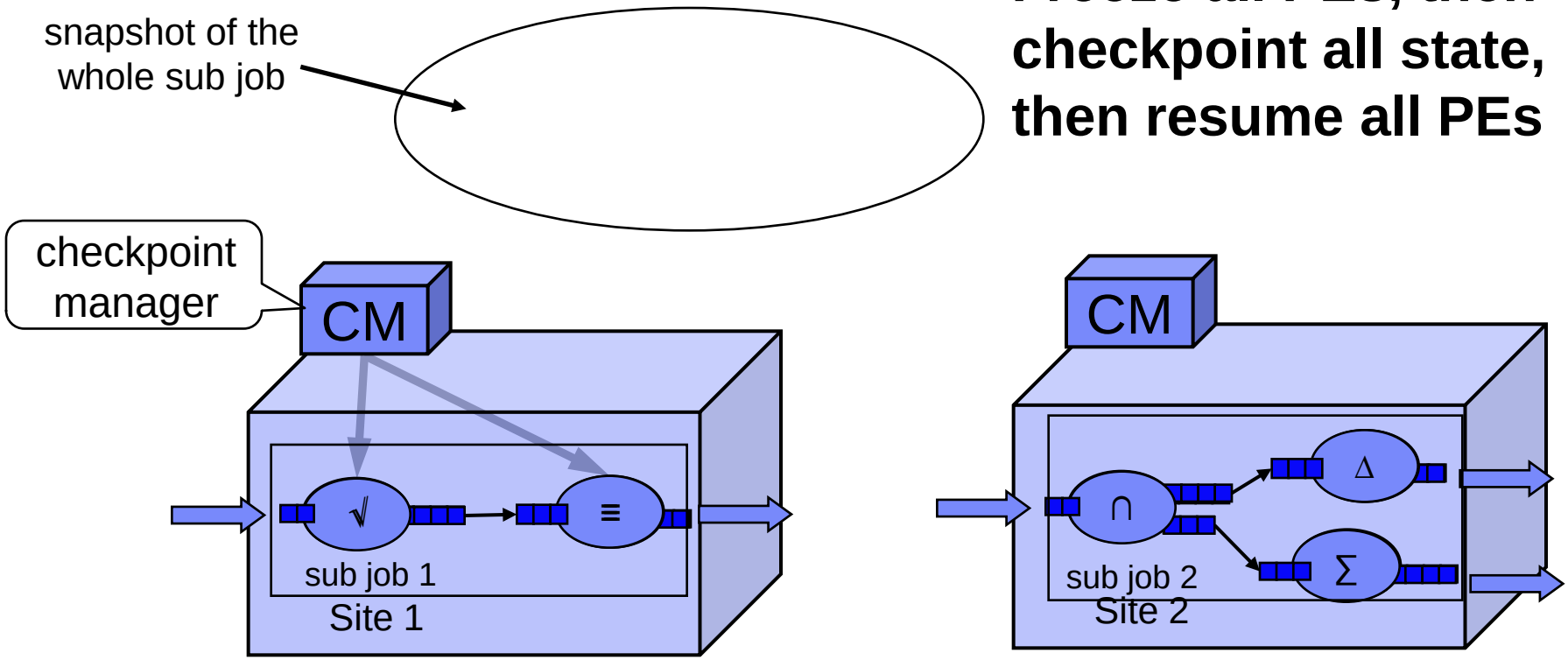
When to Trim

In U's output queue, only removing those packets that have been processed and checkpointed by D



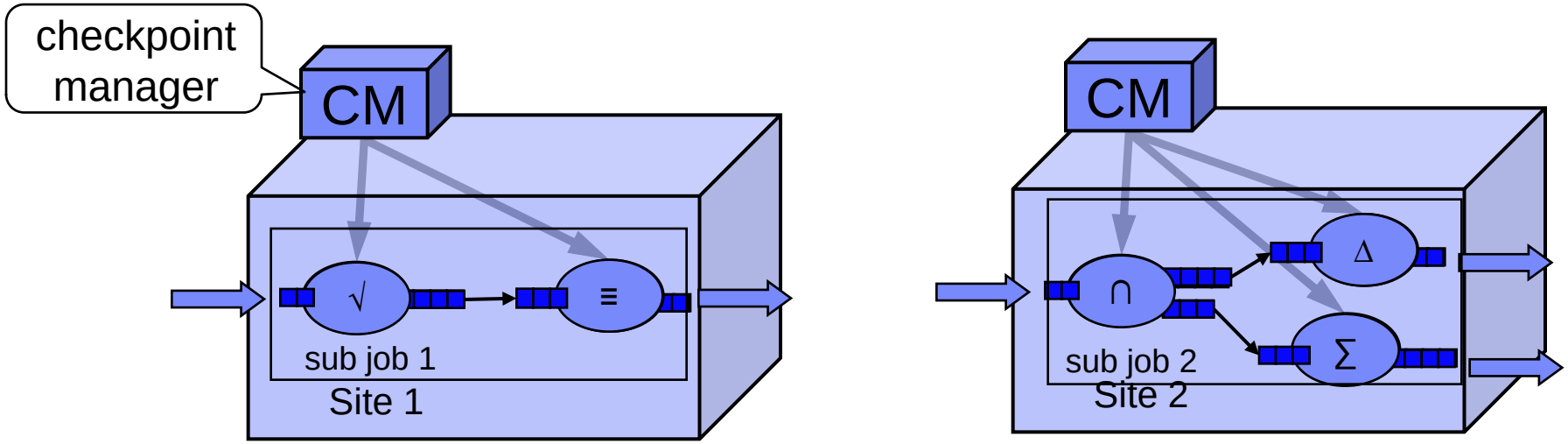
Checkpointing Multiple PEs – Synchronous

- **Freeze all PEs, then checkpoint all state, then resume all PEs**



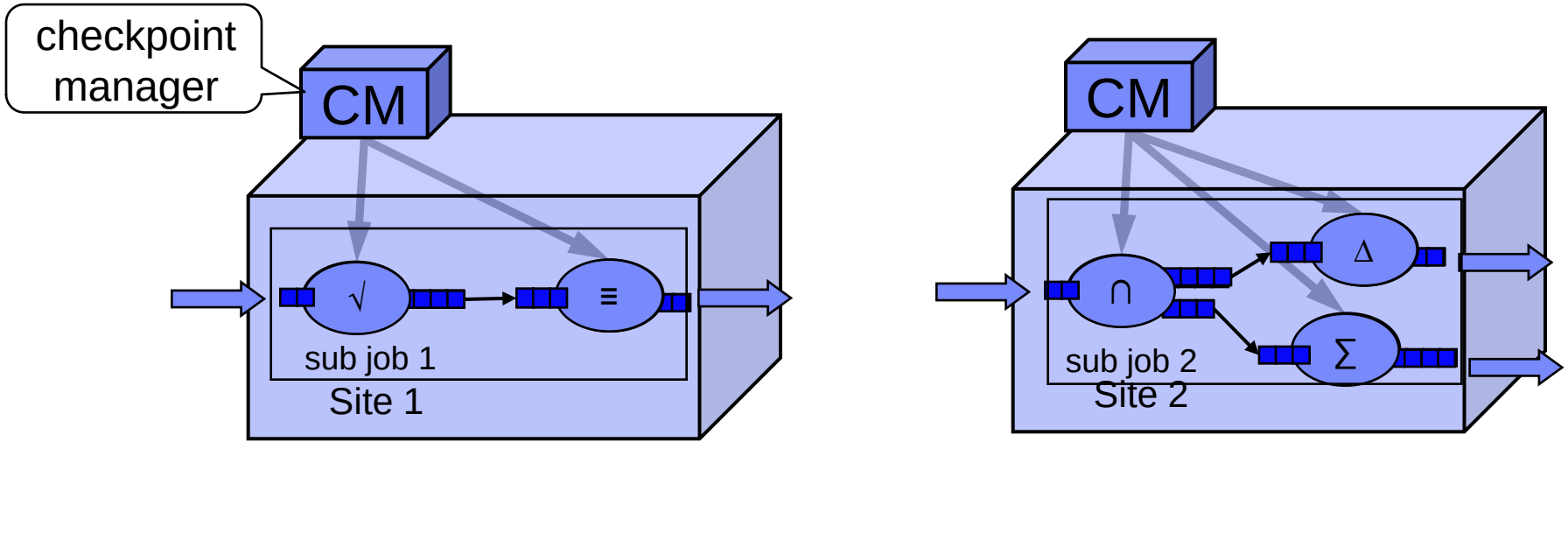
Checkpointing Multiple PEs – Individual

- **Freeze / checkpoint / resume each PE individually**



Checkpointing Multiple PEs – Sweeping

- **Checkpoint a PE immediately after receipt of acknowledgement and output queue trimming**



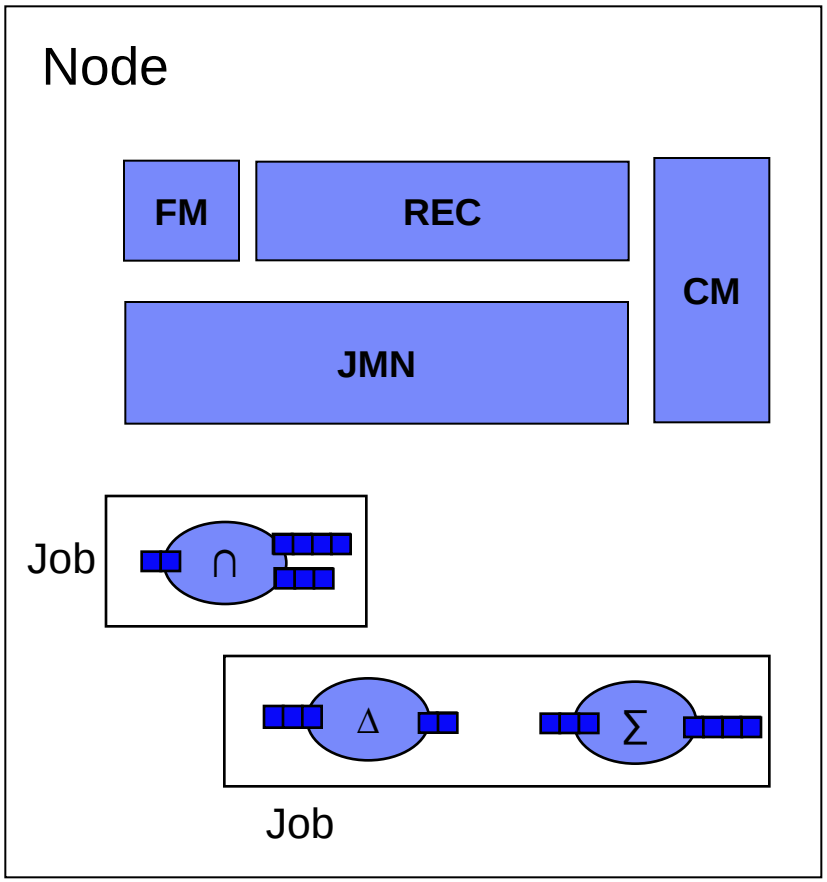
Sketch of Proof for Consistency

- **Scenario: single node failure (N_i)**
  - Actions for recovery
    - Recovering operator state
    - Recovering input queue from output queues of upstream
    - Reprocessing affected elements

only trimmed to reflect latest checkpoint of N_i

- **Scenario: multiple concurrent node failures**
  - Actions for recovery
    - Finding and recovering most upstream failed node
    - Recovering other nodes recursively

System Architecture



- **Remote Execution Coordinator**
  - manage HA protection for distributed jobs
- **Job Management**
  - manage job deployment
- **Checkpoint Manager**
  - manage checkpoint tasks according to assigned checkpoint mechanism
- **Failover Manager**
  - monitor other nodes and initiate recovery
- **Jobs and Processing Nodes**
  - take data from upstream, execute processing tasks, and send results to downstream
- **Features:**
  - A distributed job consists of multiple subjobs, each of which can choose its own specific HA mechanism (AS, PS)
  - The system coordinates the deployment and protection of subjobs among all machines

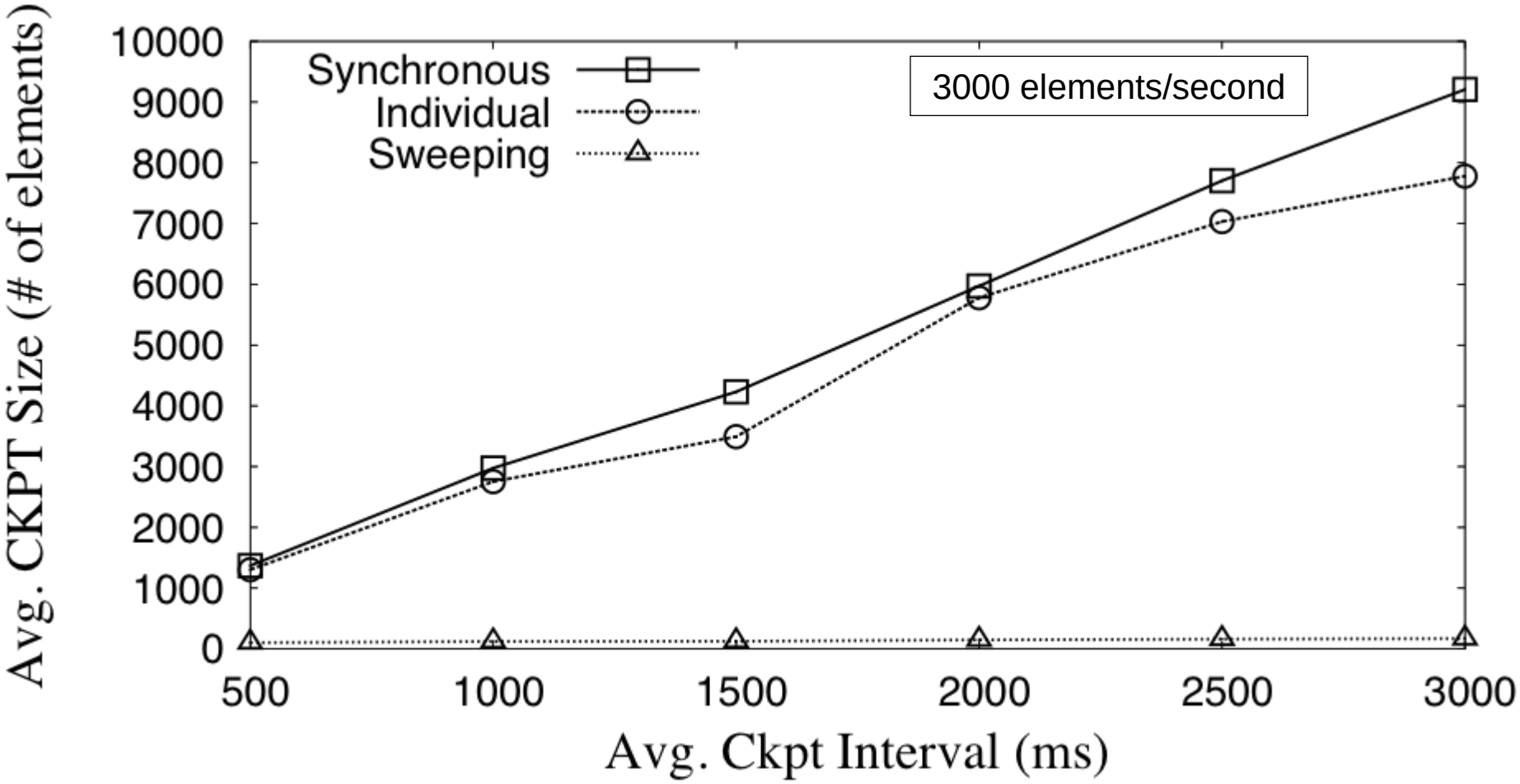
Outline

- **Background and Motivation**
- **Design and Implementation**
- **Performance Evaluation**
  - Experiment Setup
  - Overhead and Delay Results
- **Related Work**
- **Conclusions**

Experiment Setup

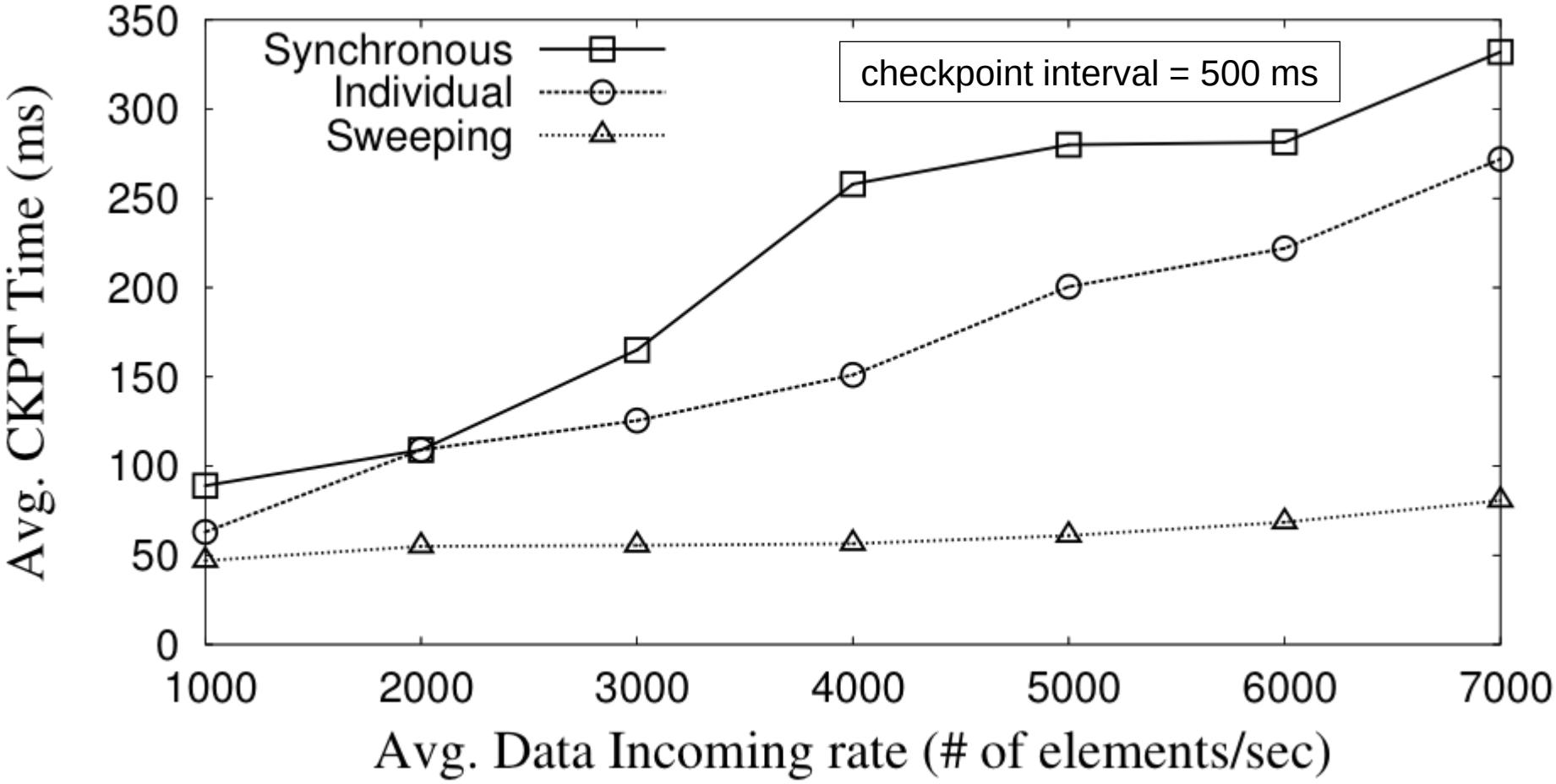
- **Testbed: a cluster environment**
  - Dual Xeon 3.06GHz CPUs, 800MHz, 512KB L2 caches, 4GB memory, 80GB disk
  - 1Gbps LAN
  - A distributed job containing 4 subjobs, each having 2 processing nodes running on one machine
- **Metrics**
  - Recovery delay
  - Message overhead

Avg. Checkpoint Queue Size Comparison



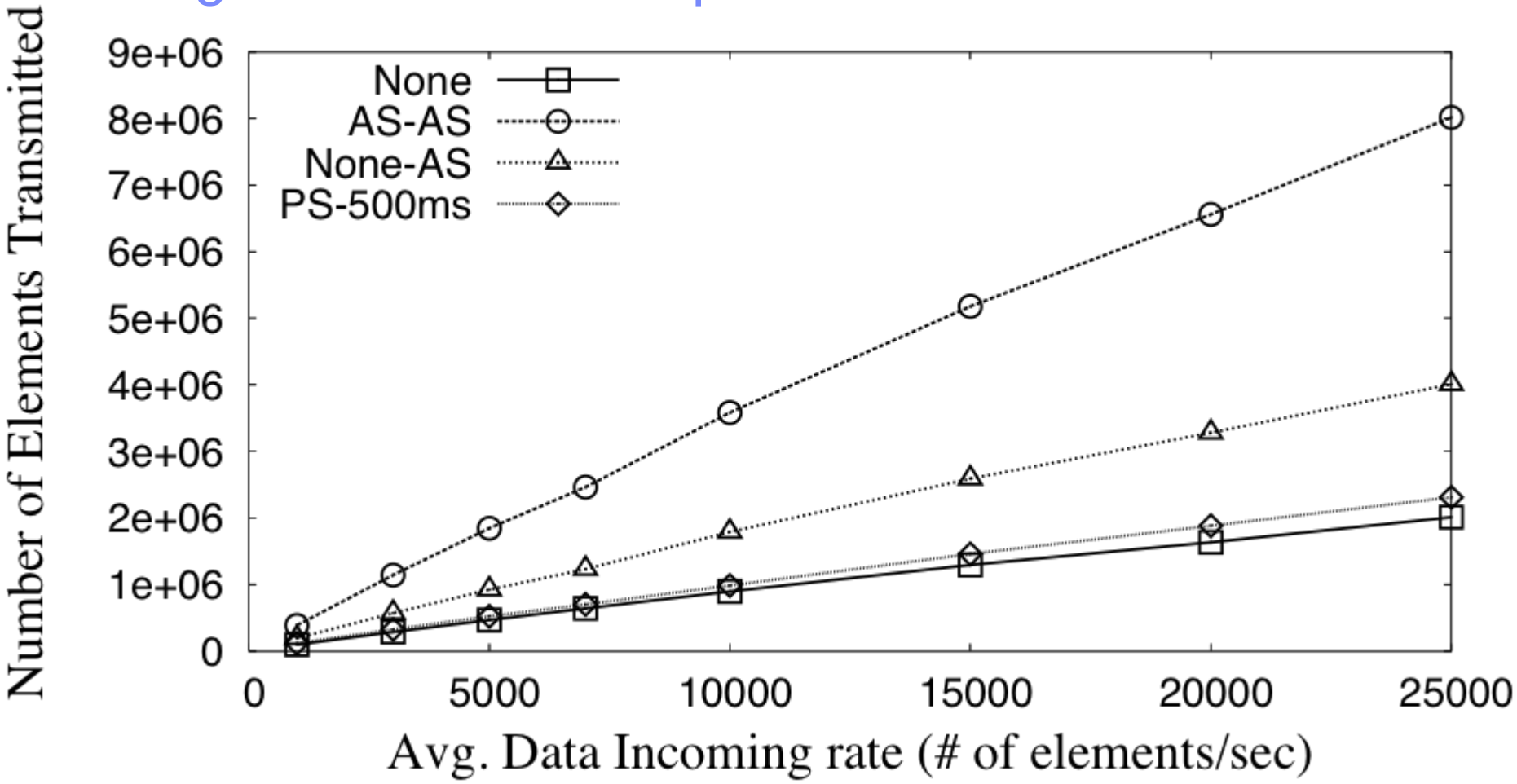
Sweeping reduces checkpoint size by about 96%

Checkpoint Time Comparison



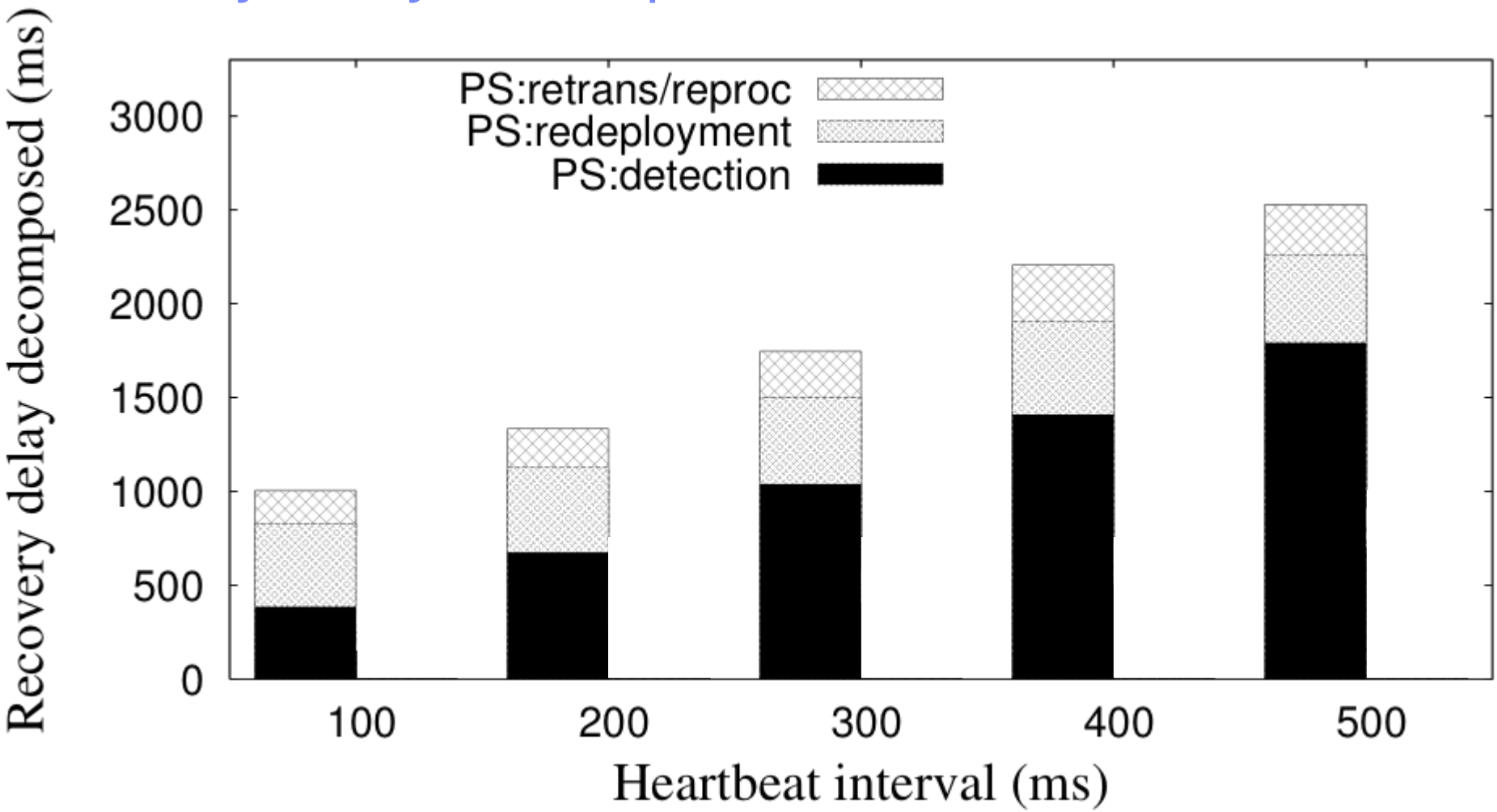
Sweeping reduces checkpoint time by about 75%

Message Overhead Comparison



AS-AS incurs almost 4 times message overhead vs. PS

Recovery Delay Decomposition



Detection delay becomes dominant with large heartbeat interval

Outline

- Background and Motivation
- Design and Implementation
- Performance Evaluation
- **Related Work**
- Conclusions

Related Work

- **Borealis**
  1. "*Fault tolerance in the Borealis distributed stream processing system*" (SIGMOD '05)
     - A variant of AS
     - Achieving flexible trade-off between availability and consistency by introducing tentative data concept
  2. "*Fast and reliable stream processing over wide area networks*" (ICDE '07)
     - A variant of AS
     - Most expensive variant; upstream sending to all downstream replicas
     - No switch required when failure occurs
  3. "*A cooperative, self-configuring high-availability solution for stream processing*" (ICDE '07)
     - A variant of PS
     - Novel checkpoint scheduling and backup assignment
     - Balances recovery load over multiple servers
  4. "*Borealis-R: a replication-transparent stream processing system for wide-area monitoring applications*" (SIGMOD '08)
     - A variant of AS
     - Same technique as in [2]
     - Novel mechanism to allow replicas execute without coordination but still produce consistent results

Related Work

- **System S**
  5. *"Towards automatic fault recovery in System-S" (ICAC '07)*
     - Checkpoint state
     - Recovery of JMN, not jobs
  6. *"Failure recovery in cooperative data streaming analysis" (ARES '07)*
     - How to select a backup site on demand, not recovery technique
  7. *"Online failure forecast for fault-tolerant data stream processing" (ICDE '08)*
     - Prediction of potential failures, a monitoring technique
     - Leverages varies system metrics (system productivity, available CPU, etc.) to predict failures before they occur

- **Comparison of AS and PS**
  8. *"High-availability algorithms for distributed stream processing" (ICDE '05)*
     - Valuable summaries of basic tradeoffs
     - PS variant has large overhead
     - Evaluation mainly based on simulations

Conclusions

- **Fundamental tradeoffs between AS and PS in stream processing systems not fully understood**
- **Our contributions**
  - A novel sweeping checkpoint mechanism
  - Proof of consistency
  - System implementation and empirical evaluation
- **Performance results providing valuable insights**
  - Importance of queue trimming in checkpointing
  - Decomposition of recovery delay

Questions?

Zhe Zhang

zhezhang@ornl.gov

http://users.nccs.gov/~zzhang3/pubs/empirical-middleware09.pdf