A-SAS : An Adaptive High-Availability Scheme for Distributed Stream Processing Systems

Hiroaki SHIOKAWA (University of Tsukuba, Japan)
Hiroyuki KITAGAWA (University of Tsukuba, Japan)
Hideyuki KAWASHIMA (University of Tsukuba, Japan)

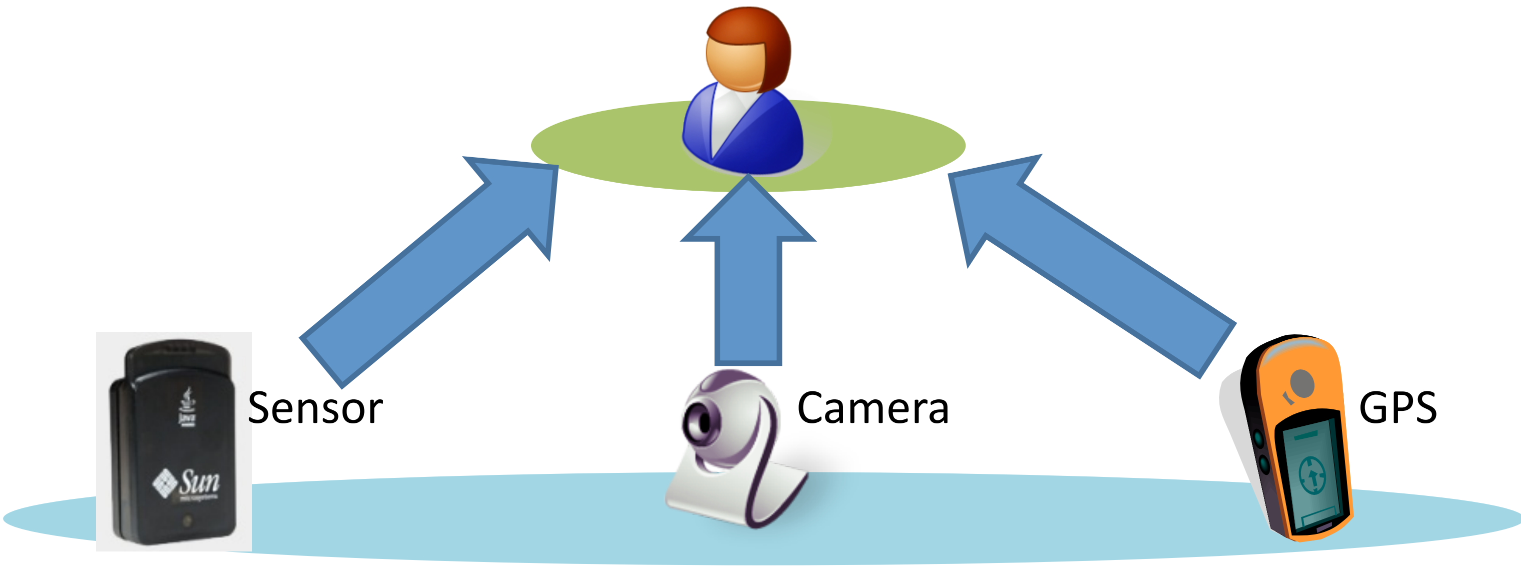
Outline

1. Background
2. Related Works
3. Proposed Scheme : Adaptive Semi-Active Standby
4. Evaluation
5. Conclusion and Future Works

Background

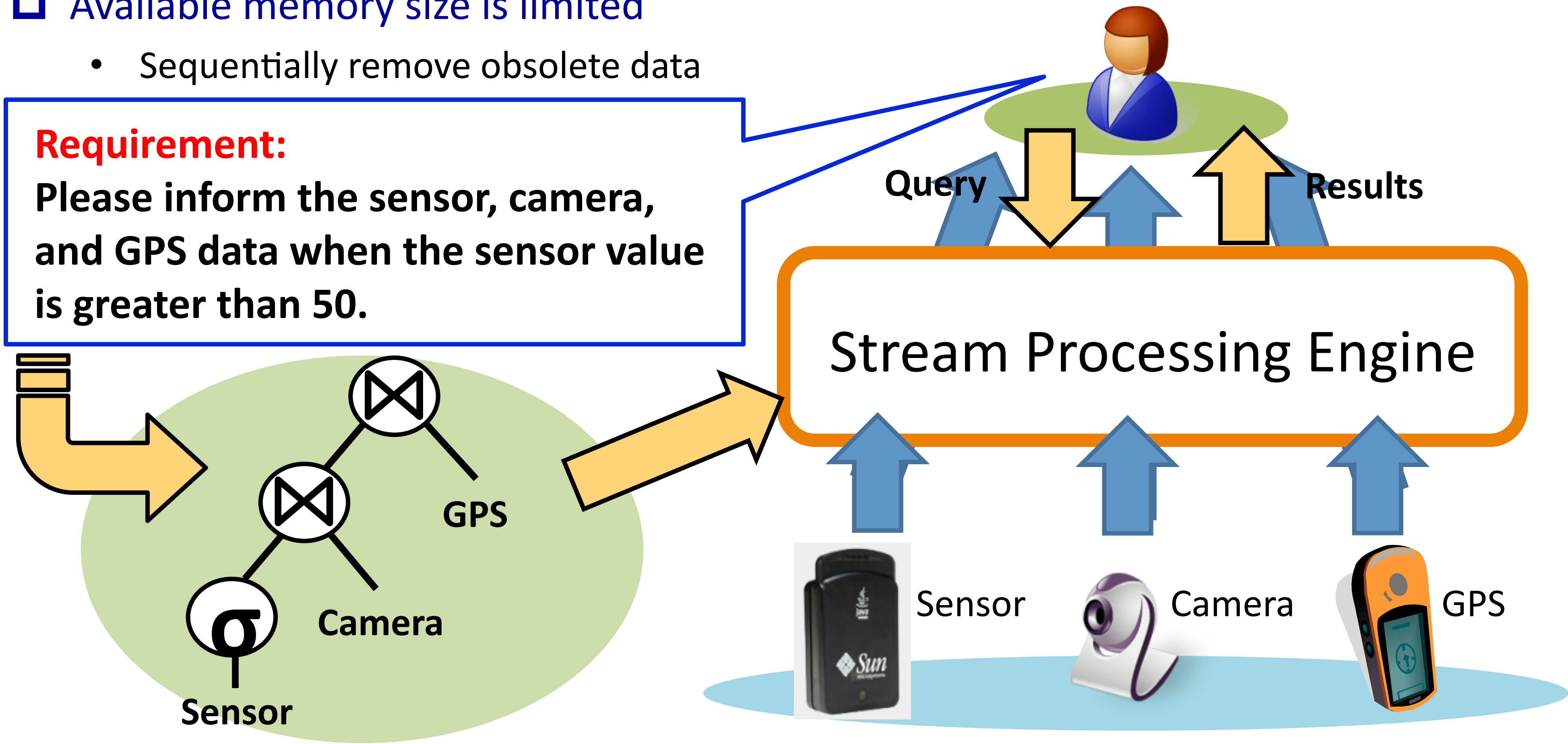
Background

- **A huge amount of <u>stream data</u> is available**
  - Data streams provide information changing over time
  - e.g. Sensor data, Camera data, GPS data, and so on…
- **A demand for query processing on stream data**
  - Filtering, Integration, Aggregation, and so on…



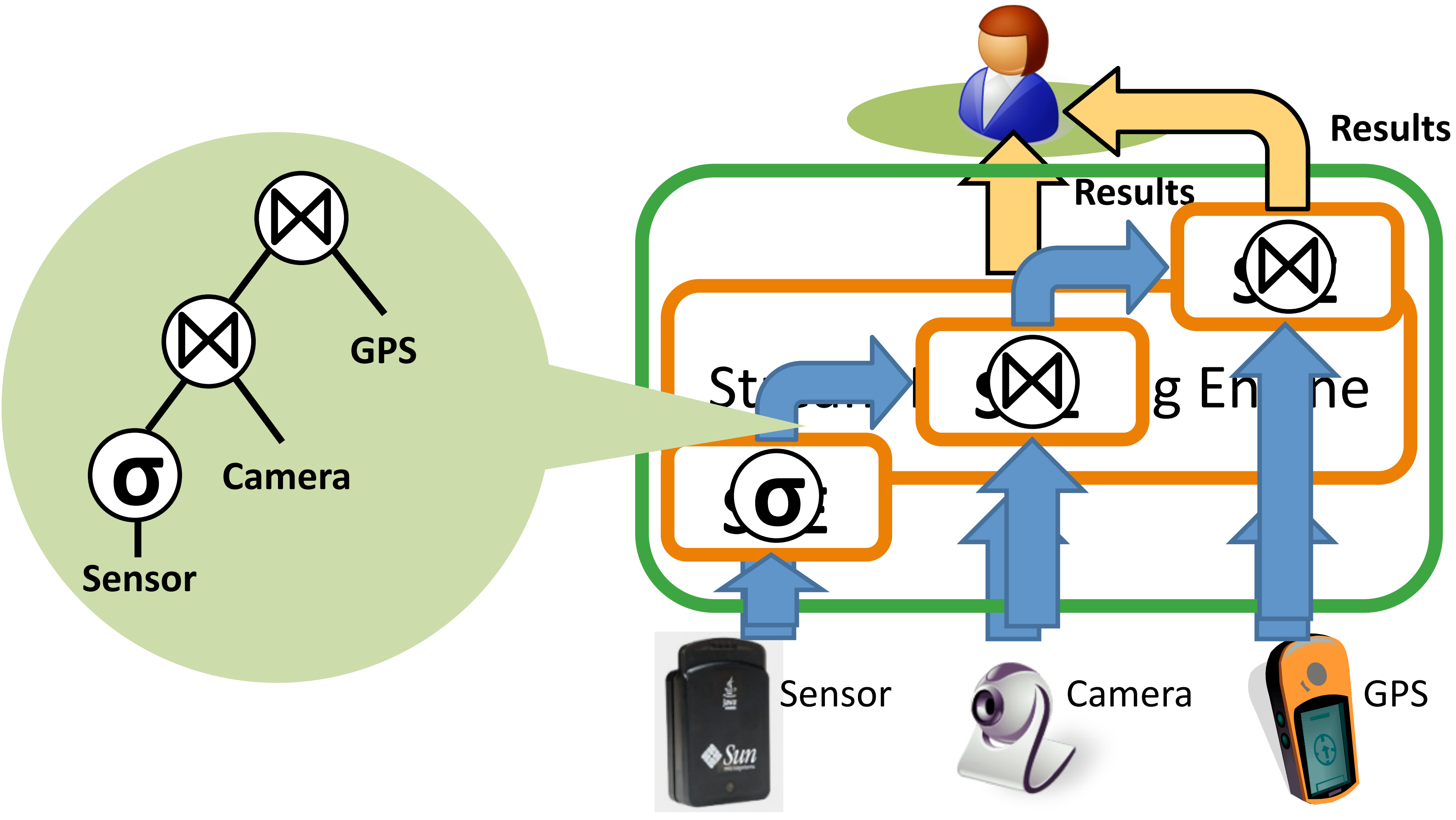
Stream Processing Engine (SPE)

- SPE is an infrastructure system for stream data processing
- Perform continuous query processing
  - A query is executed in a long time
- Available memory size is limited
  - Sequentially remove obsolete data



High-Availability Scheme for Distributed SPE

- To use data sources at remote sites or to reduce SPE’s loads, developers build a Distributed SPE (DSPE)

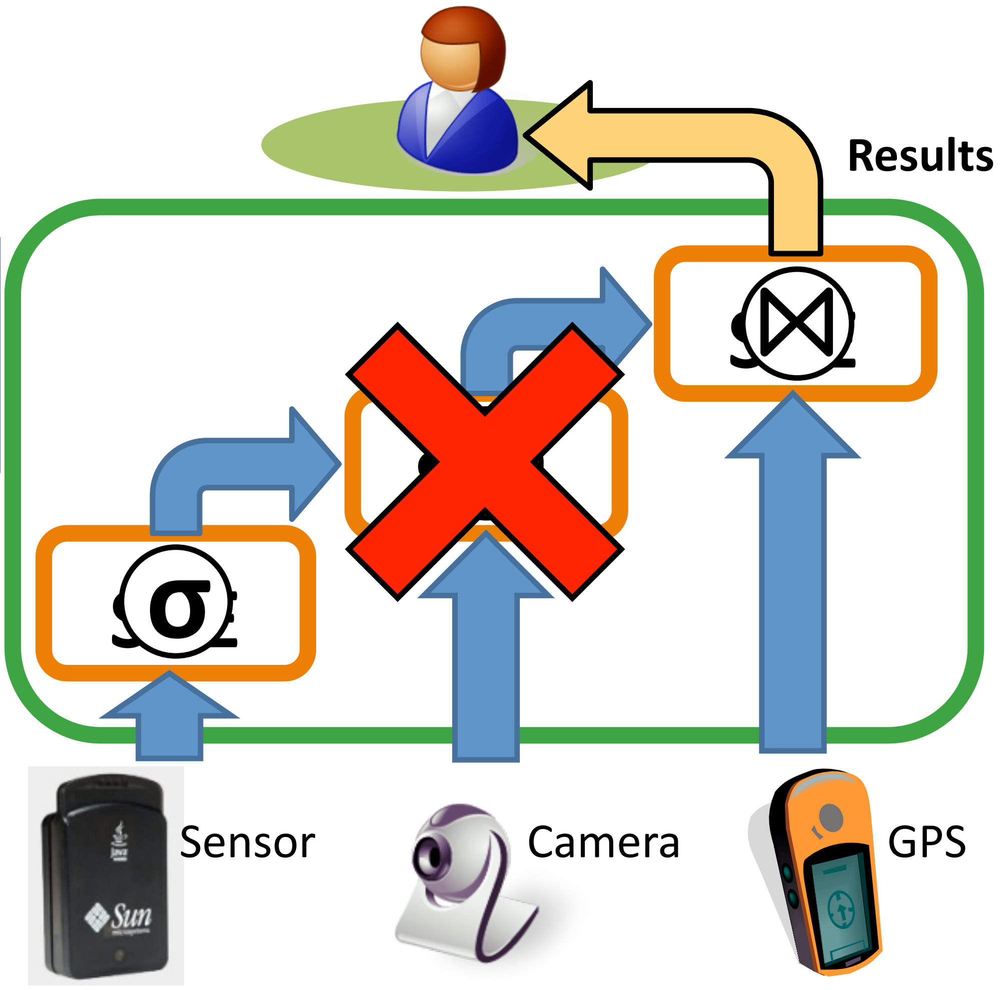


High-Availability Scheme for Distributed SPE

- To use data sources at remote sites or to reduce SPE's loads, developers build a Distributed SPE (DSPE)

Node failures can bring
1. suspension of the whole system
2. loss of a large amount of data

High-Availability Scheme must be considered for DSPE



Existing High-Availability Schemes[1]

[1] J. Hwang et al., "High-Availability Algorithms for Distributed Stream Processing," Proc. ICDE, 2005.

Active Standby

Downtime **Short**

Bandwidth usage **High**

Upstream Backup

Downtime **Long**

Bandwidth usage **Low**

Actual Environment

1) DSPE has two cost limitations: bandwidth usage and downtime.
2) Stream data properties (e.g. data rate, data size) dynamically change.

In existing schemes, they cannot balance the cost of bandwidth usage and downtime adaptively.

Objective

- Proposal of a new high-availability scheme named Adaptive Semi-Active Standby (A-SAS)

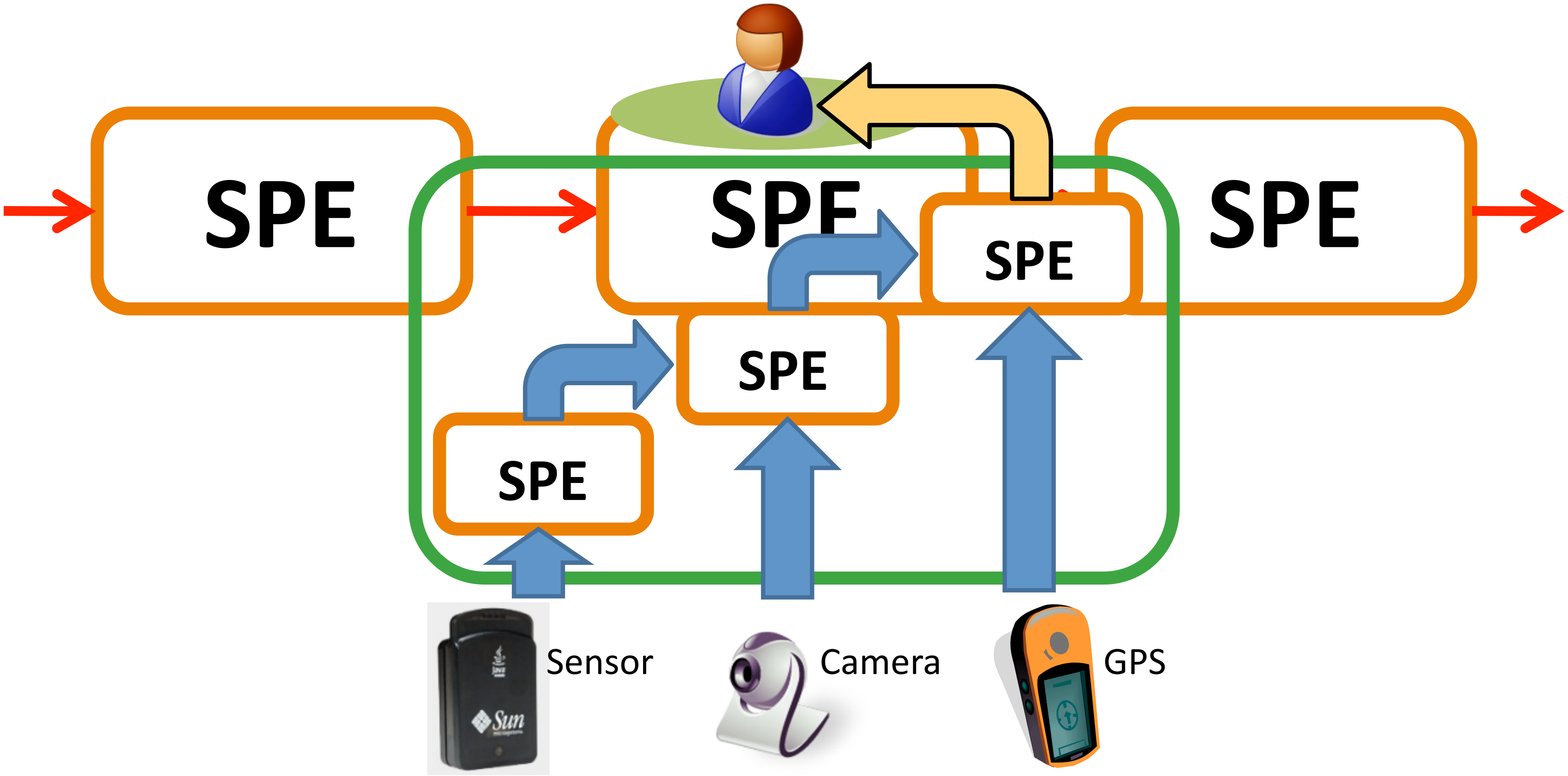
<Approach 1>
Estimates the running costs both recovery time and bandwidth usage.

<Approach 2>
Balances the costs of recovery time and bandwidth usage adaptively to meet their limitations.

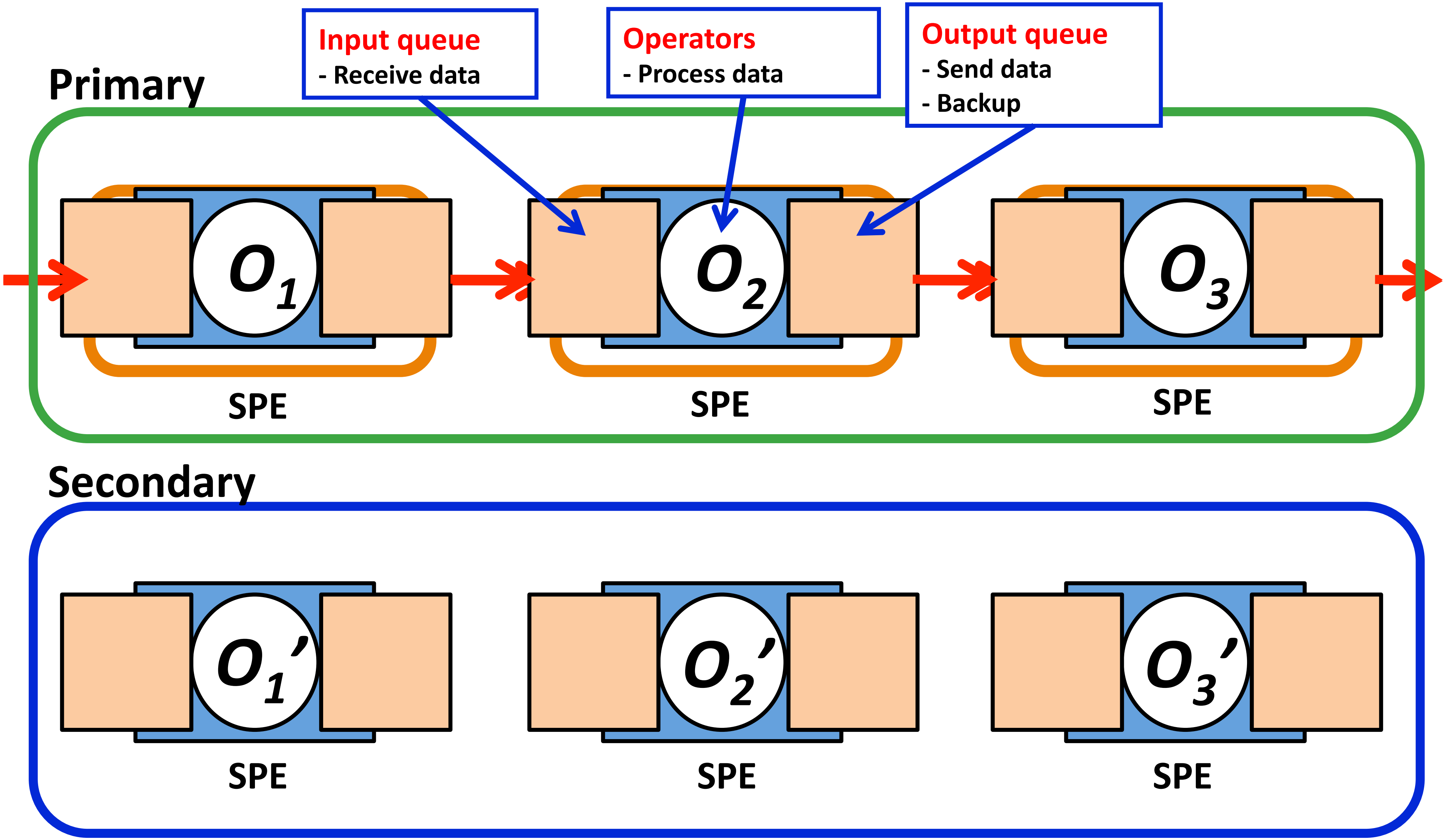
- This contribution frees system developers from difficult system tuning for high-availability.

Related Works

System Model

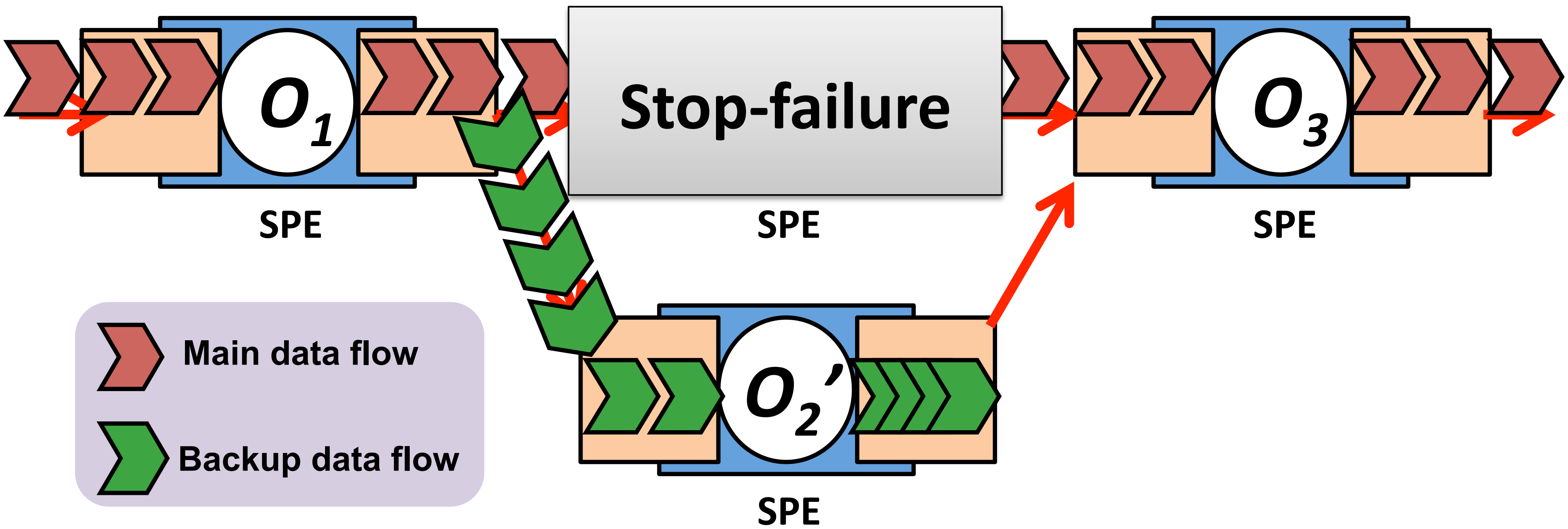


System Model



Related Work : Active Standby

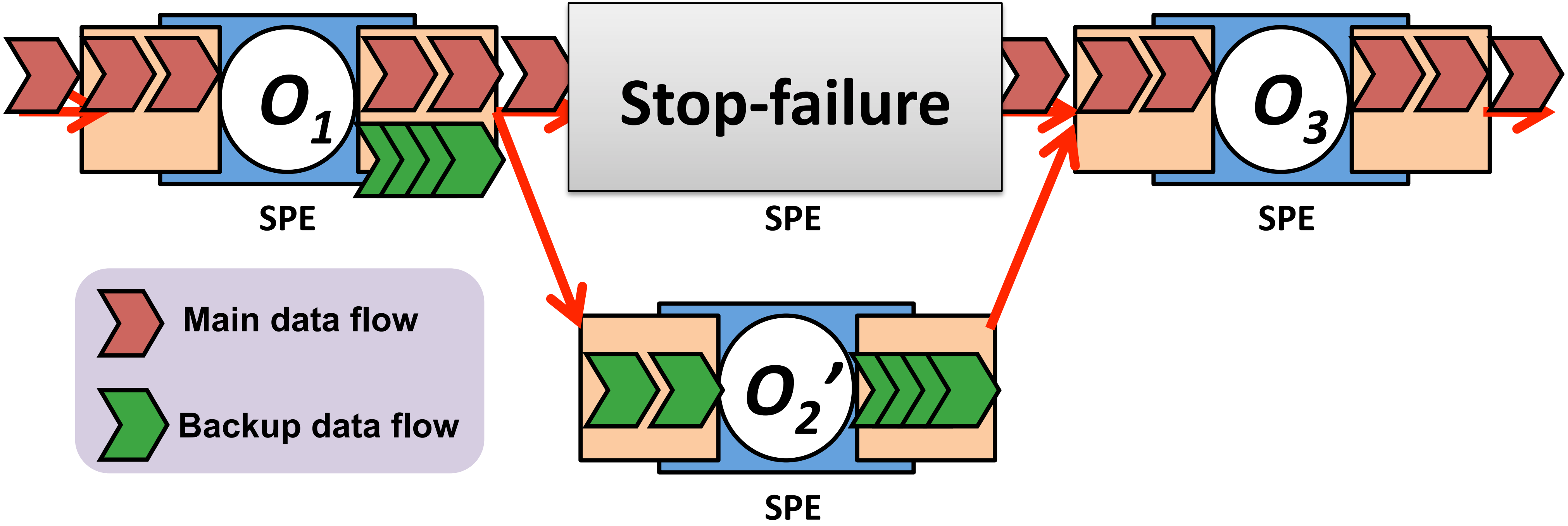
- Each secondary receives processing results from the upstream primary and processes them in parallel with the primary.



The backup data are sent in parallel with the main data: **High bandwidth usage**
Secondary always has the same state as its primary: **Short recovery time**

Related Work : Upstream Backup

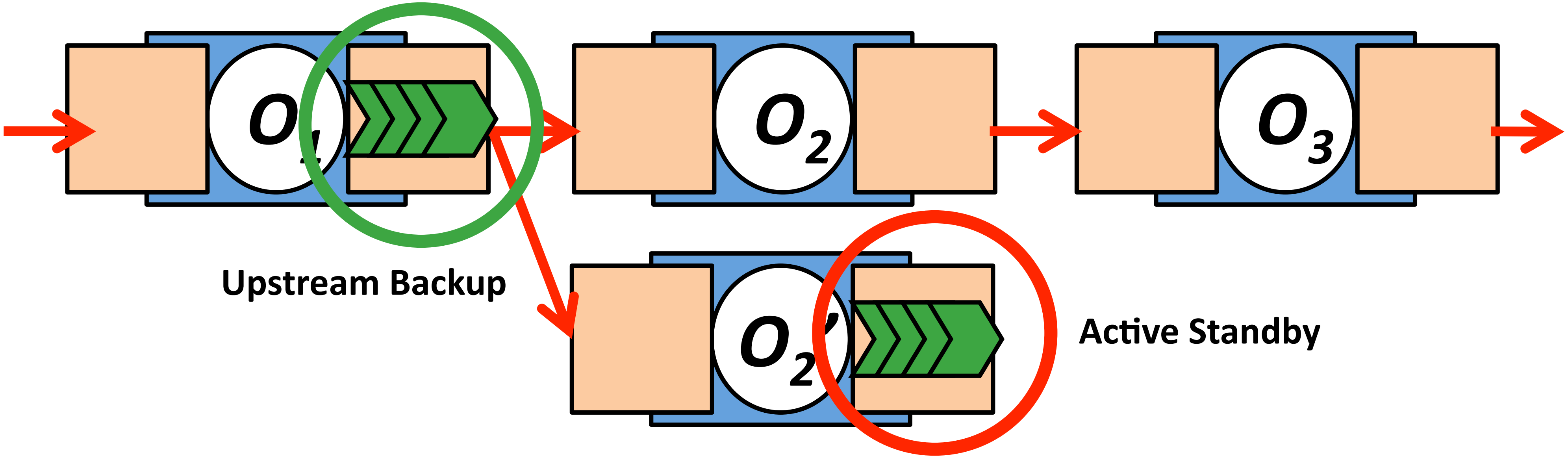
- All backup data are saved in the upstream primary's output queue.



Only after the failure, upstream primary sends backup data : **Low bandwidth usage**
Secondary must reprocess all backup data : **Long recovery time**

Proposed Scheme:
Adaptive Semi-Active Standby

Basic scheme : Semi-Active Standby



Each scheme has its own feature depended on WHERE and WHEN the backup data are save

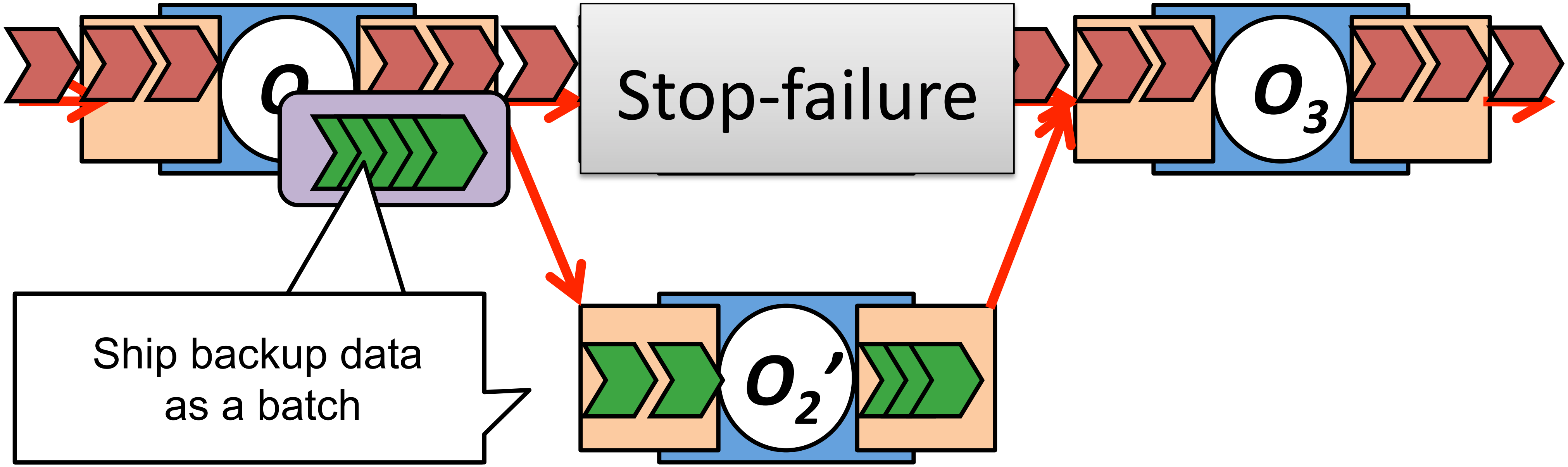
【Basic idea】

To realize the balance of bandwidth and recovery time, we decide the optimal backup data allocation by controlling the timing of shipping backup data.

Basic Scheme : Semi-Active Standby

Each primary ships all backup data in its output queue, when its output queue length reaches the limit which is given by developer.

A block of the backup data shipped from primary : **Batch.**
The limit of the queue length : **Batch size.**

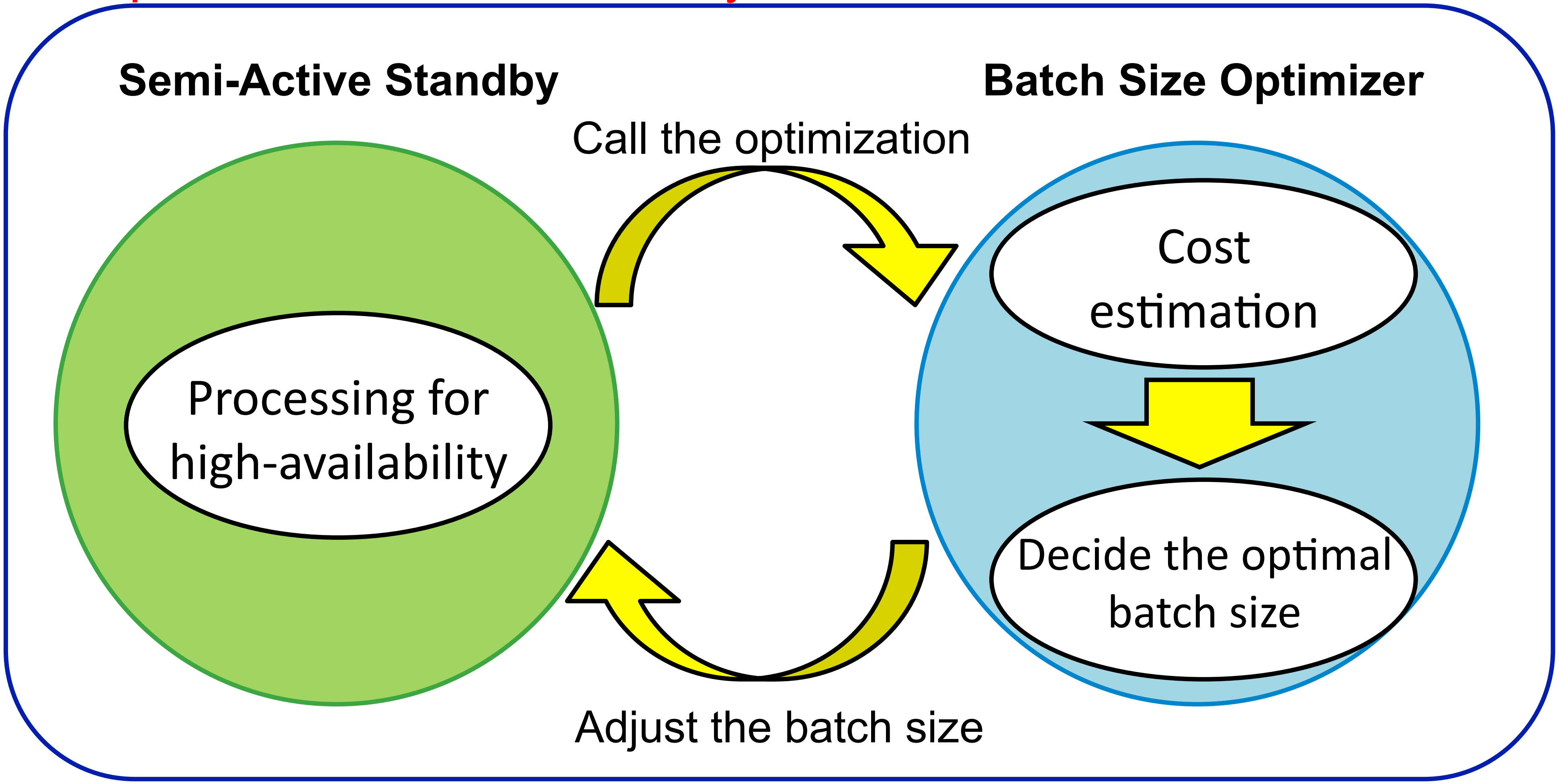


Increase the batch size : bandwidth usage → High, recovery time → Short
Decrease the batch size : bandwidth usage → Low, recovery time → Long

Overview of the proposed scheme
Adaptive Semi- Active Standby (A-SAS)

- Extend SAS

Adaptive Semi-Active Standby



Cost Estimation with Cost Models

- Calculate the estimated bandwidth usage *BW** and estimated recovery time *Rtime**

B_{size} means current batch size

Cost model for bandwidth usage

Bandwidth usage ＝ bytes the primary sends in a unit time

$$BW^* = B_{send} \sum_{i=1}^{B^*_{size}} |tuple_i|$$

B_{send} :	Average number of batches sent per time unit
$\sum_{i=1}^{B^*_{size}} \lvert tuple_i \rvert$:	Data size per unit of batch

This model can cope with changes in the data rate and data size.

Cost model for recovery time

Recovery time ＝ the total time for resending and reprocessing backup data

$$R^*_{time} = B_{size}(S_{time} + P_{time})$$

S_{time} :	Average time of resending per unit data
P_{time} :	Average time of reprocessing per unit data

This model can cope with changes in the network delay and processing time

Decide the Batch Size

☐ Adjust the batch size using the result of estimations

Define bandwidth usage and recovery time limits set by developer as BW and R_{time}.

B_{delta} is the tuning range of the batch size.

＜Batch size optimization＞

1) Calculate the estimated bandwidth usage and recovery time from cost models

2) Decide the batch size

- a) If estimated bandwidth $> BW$ and estimated recovery time $< R_{time}$ then
  The batch size is incremented by B_{delta}, and return to 1).
- b) If estimated bandwidth $< BW$ and estimated recovery time $> R_{time}$ then
  The batch size is decremented by B_{delta}, and return to 1).
- c) If estimated bandwidth $> BW$ and estimated recovery time $> R_{time}$ then
  Execute a) or b) which has higher priority, return to 1).
- d) If estimated bandwidth $< BW$ and estimated recovery time $< R_{time}$ then
  Do nothing

Evaluation

Overview

Objective

- Evaluate and compare the performance of A-SAS when stream data properties change.

Comparison schemes

- Active Standby(AS) and Upstream Backup (UB)

Metrics

Bandwidth usage and Recovery time

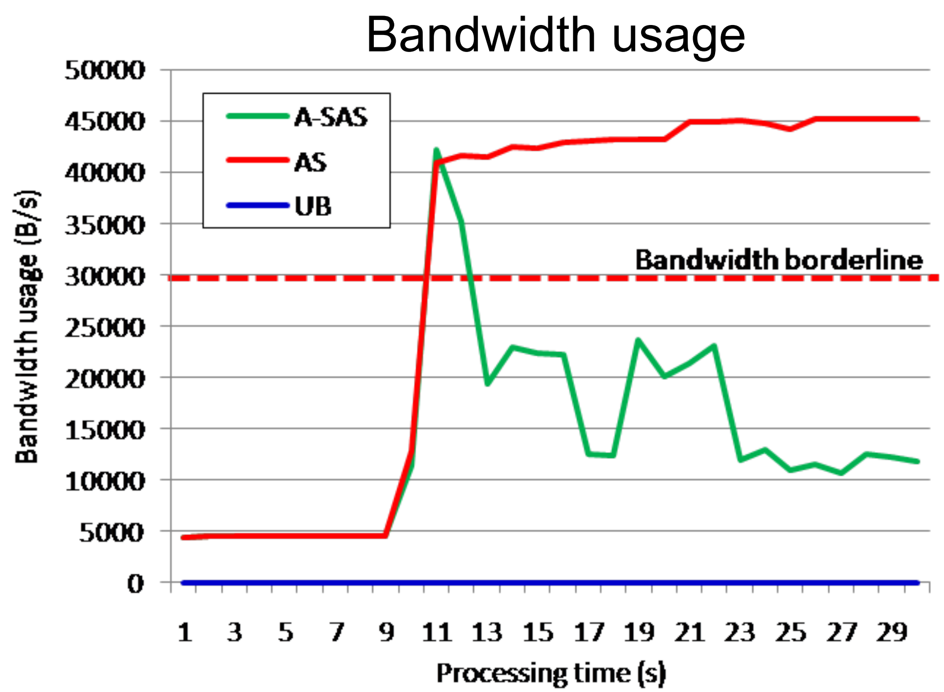
Other parameters of the evaluation

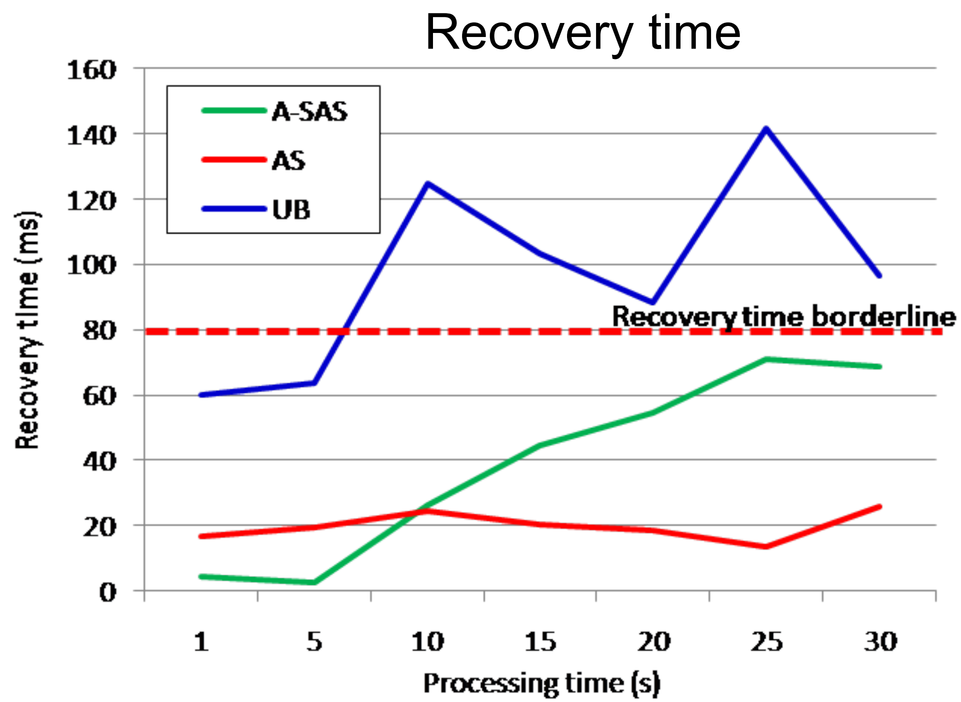
- Batch size at the start time is 1
- Tuning ranges of the batch size is 5
- Optimization interval is 10ms
- Recovery time has priority regarding optimization

Evaluation 1 : Comparison of Changes in Data Rate

- Set up the data rate as 100 tuples/s.
- After 10 seconds have passed, increase the rate to 1000 tuples/s.

▪Set up the limit of bandwidth usage as 30KB/s, and limit of recovery time as 80ms



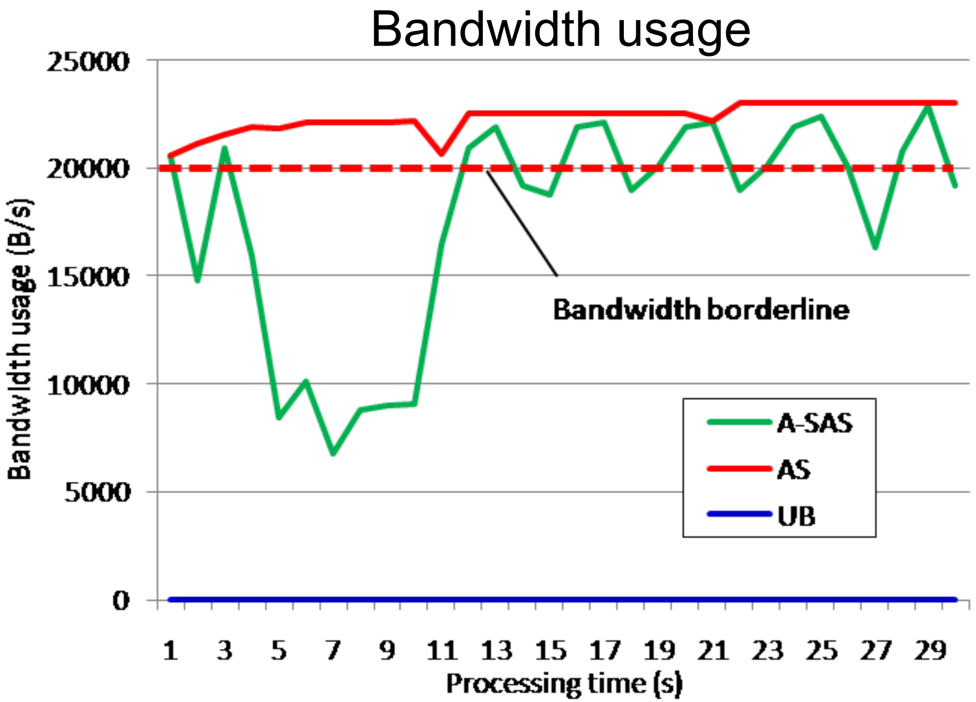


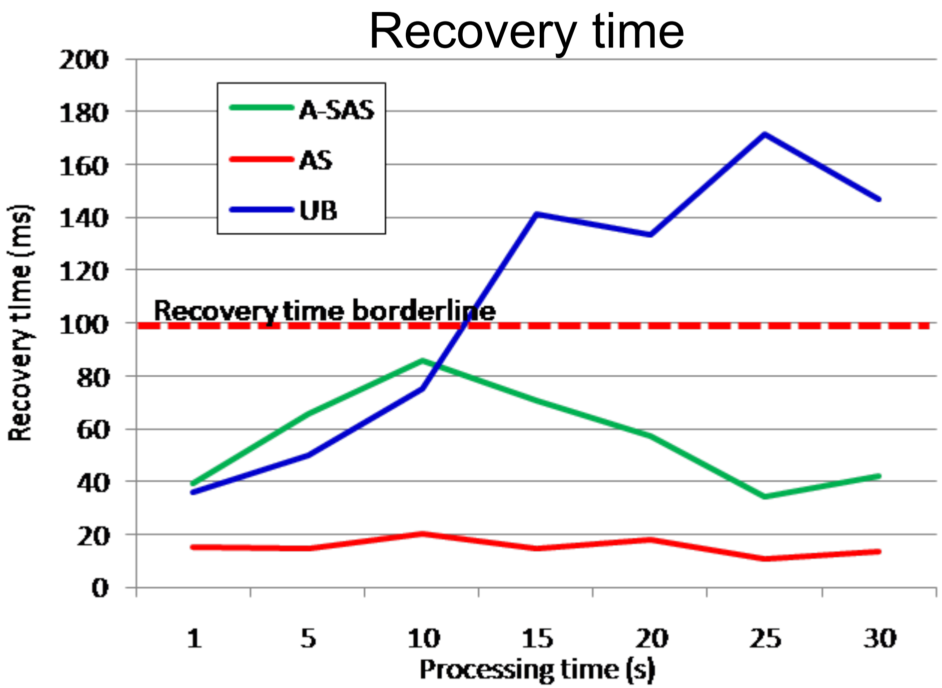
▪After the first 12 seconds, A-SAS meet the bandwidth usage limitation.
▪A-SAS is also able to meet the recovery time limitation.

Evaluation 2 : Comparison of Changes in Rerocessing Time

- Set up the average time of reprocessing per unit data as 0.05ms.
- After 10 seconds have passed, increase the average time of reprocessing to 1ms.

・Set up the limit of bandwidth usage as 20KB/s, and limit of recovery time as 100ms





・After the increasing, A-SAS sets a priority on recovery time, because A-SAS cannot meet both limitations in this setting.

・Because of above the reason, bandwidth fluctuates between upper and lower sides of limitation.

Conclusions and Future Work

Conclusions and Future Work

- We propose Adaptive Semi-Active Standby, a new high-availability scheme that can adaptively balance costs of bandwidth usage and recovery time.
- Experiments clearly showed that Adaptive Semi-Active Standby exhibited advantages compared with existing schemes Active Standby and Upstream Backup.
- Future works
  - More sophisticated high-availability schemes for processing multiple data streams.
  - Implementation of the proposed scheme in a large scale DSPE environment.

筑波大学
University of Tsukuba

Thank you for your Attention.