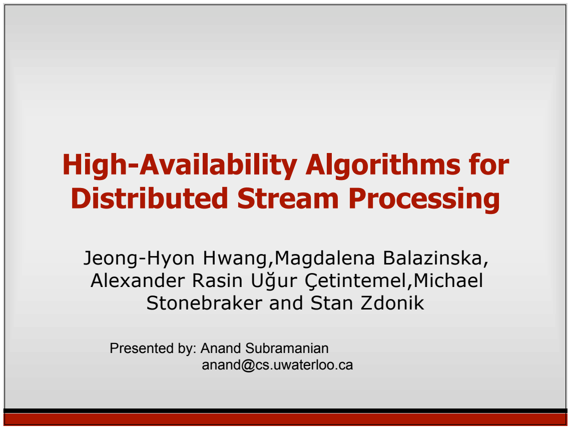
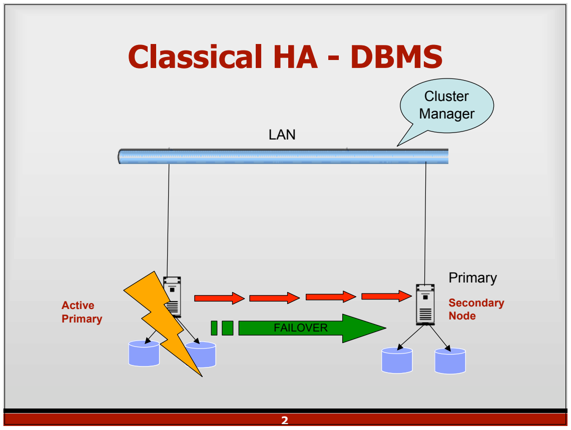


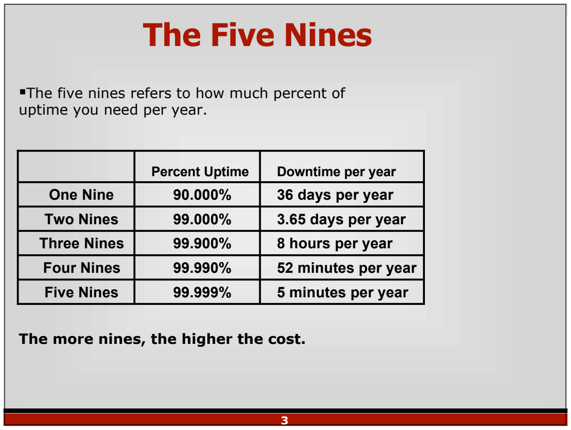
High-Availability Algorithms for Distributed Stream Processing

Jeong-Hyon Hwang,Magdalena Balazinska, Alexander Rasin Uğur Çetintemel,Michael Stonebraker and Stan Zdonik

Presented by: Anand Subramanian
anand@cs.uwaterloo.ca



Classical HA - DBMS

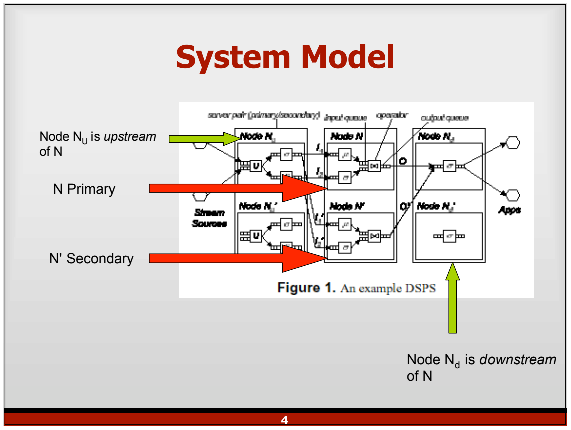


The Five Nines

- The five nines refers to how much percent of uptime you need per year.

	Percent Uptime	Downtime per year
One Nine	90.000%	36 days per year
Two Nines	99.000%	3.65 days per year
Three Nines	99.900%	8 hours per year
Four Nines	99.990%	52 minutes per year
Five Nines	99.999%	5 minutes per year

The more nines, the higher the cost.



System Model

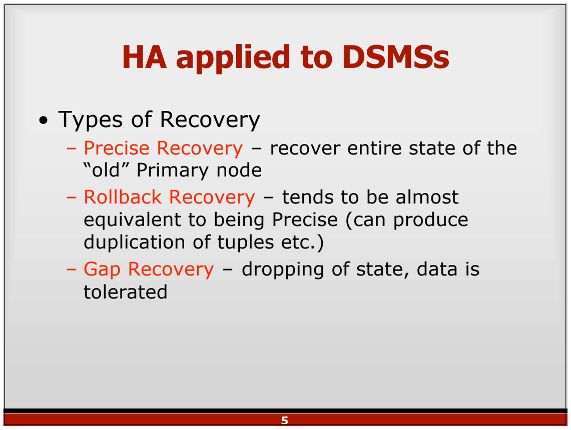
Node N_U is *upstream* of N

N Primary

N' Secondary

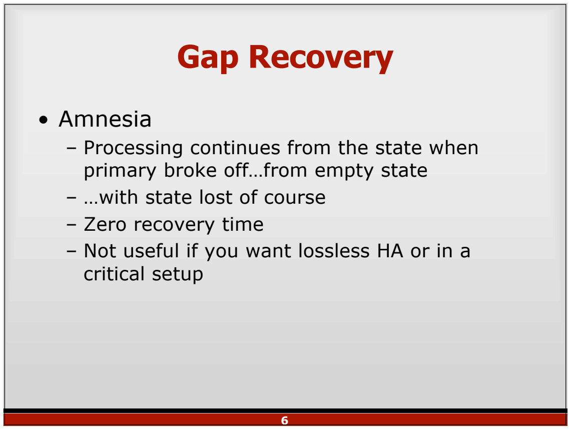
Figure 1. An example DSPS

Node N_d is *downstream* of N



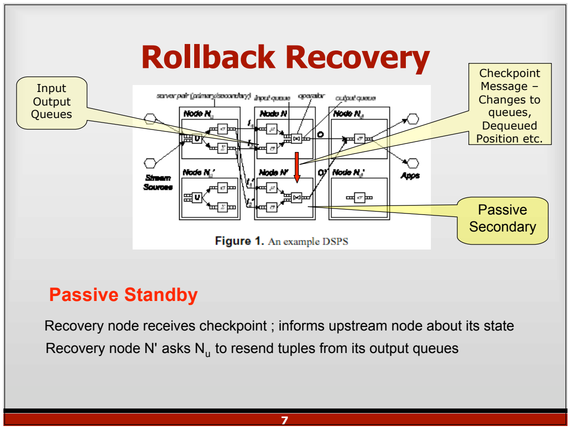
HA applied to DSMSs

- Types of Recovery
  - Precise Recovery – recover entire state of the “old” Primary node
  - Rollback Recovery – tends to be almost equivalent to being Precise (can produce duplication of tuples etc.)
  - Gap Recovery – dropping of state, data is tolerated



Gap Recovery

- Amnesia
  - Processing continues from the state when primary broke off…from empty state
  - …with state lost of course
  - Zero recovery time
  - Not useful if you want lossless HA or in a critical setup



Rollback Recovery

Figure 1. An example DSPS

Input Output Queues

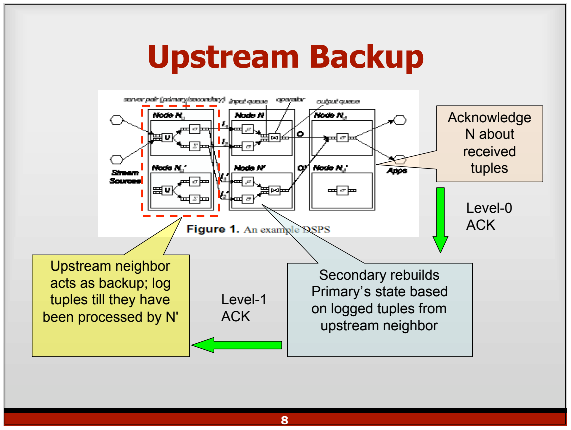
Checkpoint Message – Changes to queues, Dequeued Position etc.

Passive Secondary

Passive Standby

Recovery node receives checkpoint ; informs upstream node about its state

Recovery node N' asks N_u to resend tuples from its output queues



Upstream Backup

Figure 1. An example DSPS

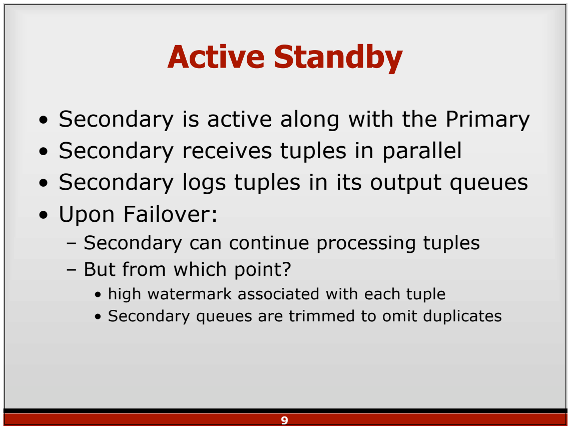
Acknowledge N about received tuples

Level-0 ACK

Upstream neighbor acts as backup; log tuples till they have been processed by N'

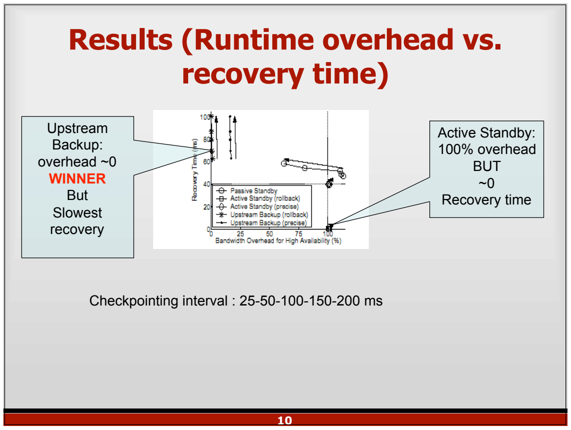
Level-1 ACK

Secondary rebuilds Primary's state based on logged tuples from upstream neighbor



Active Standby

- Secondary is active along with the Primary
- Secondary receives tuples in parallel
- Secondary logs tuples in its output queues
- Upon Failover:
  - Secondary can continue processing tuples
  - But from which point?
    - high watermark associated with each tuple
    - Secondary queues are trimmed to omit duplicates

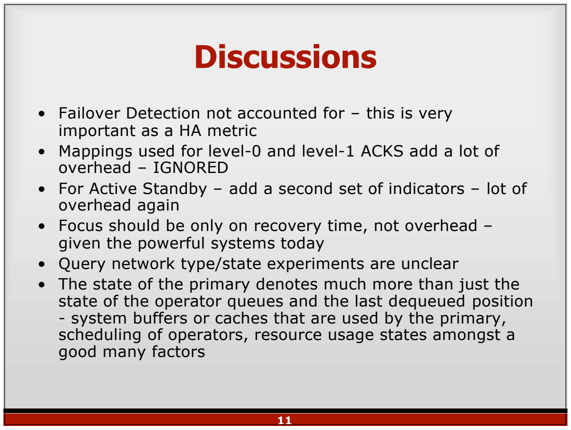


Results (Runtime overhead vs. recovery time)

Upstream Backup: overhead ~0 **WINNER** But Slowest recovery

Active Standby: 100% overhead BUT ~0 Recovery time

Checkpointing interval : 25-50-100-150-200 ms



Discussions

- Failover Detection not accounted for – this is very important as a HA metric
- Mappings used for level-0 and level-1 ACKS add a lot of overhead – IGNORED
- For Active Standby – add a second set of indicators – lot of overhead again
- Focus should be only on recovery time, not overhead – given the powerful systems today
- Query network type/state experiments are unclear
- The state of the primary denotes much more than just the state of the operator queues and the last dequeued position - system buffers or caches that are used by the primary, scheduling of operators, resource usage states amongst a good many factors