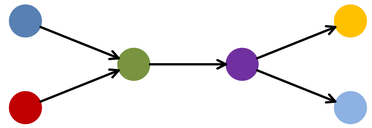
An Optimizer for Distributed Intra-Node-Parallel Streaming Systems

Matthias J. Sax[#12], Malu Castellanos[+2], Qiming Chen[+2], Meichun Hsu[+2]

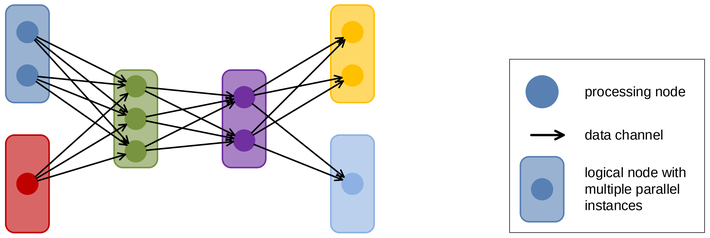
Streaming and Parallel Dataflow Execution

- *Data Intensive Computing.*
- Requires low response time, up to (near) real-time analysis.
- MapReduce does not fit (batch system).
- New class of intra-node parallel streaming systems address this problem: e.g., Storm, S4, Muppet.

Input is a dataflow specified as directed acyclic graph (DAG):

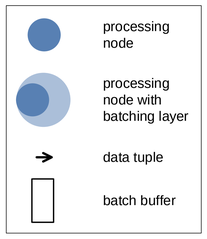


Dataflow (called Topology in Storm) is executed in a parallel manner:

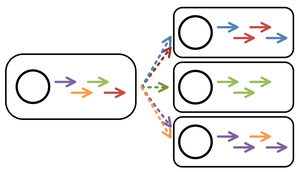


Batching in Streaming Systems

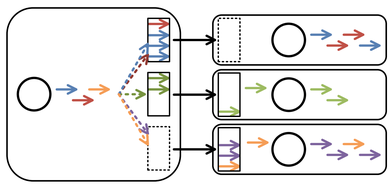
- Sending data tuple-by-tuple results in high network overhead.
- Tuple batching can increase throughput.

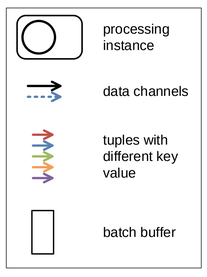


Key-based data distribution (w/o batching):



Novel batching schemas for intra-node parallelism:





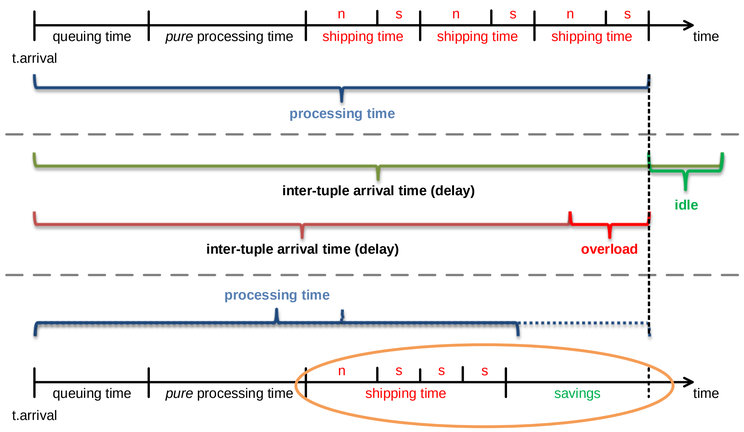
[#]Humboldt-Universität zu Berlin
[+]Hewlett-Packard Laboratories

[1]mjsax@informatik.hu-berlin.de
[2]{firstname.lastname}@hp.com

Cost Model for *Batch Size* and *Degree of Parallelism*

Optimizing Batch Size:

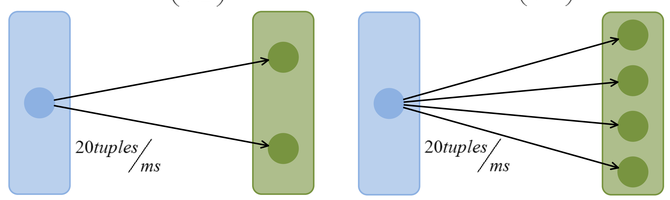
- batching reduces network overhead *n*
- n is shared over multiple tuples, each with payload *s*



Optimizing *Degree of Parallelism (dop)*:

- increasing dop reduces load on single node

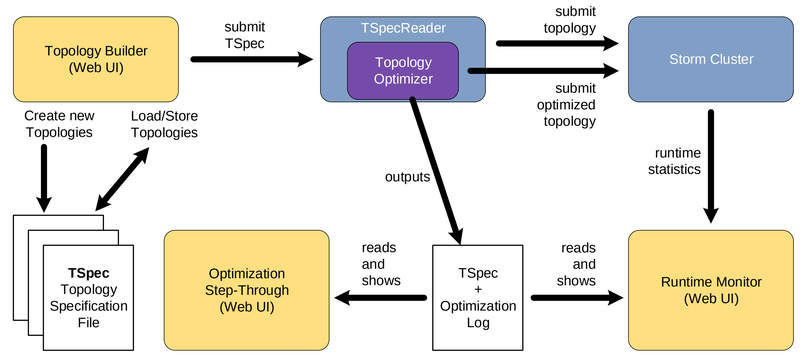
$$delay = \left(\frac{20}{ms}\Big/2\right)^{-1} = 0.1ms \qquad delay = \left(\frac{20}{ms}\Big/4\right)^{-1} = 0.2ms$$



Topology Optimization Algorithm

1: $P \leftarrow$ all source nodes
2: **while** P is not empty **do**
3: **for all** $p \in P$ **do**
4: **if** input latency is smaller than act. proc. time **then**
5: $b \leftarrow$ calc batch size to reduce ship. time
6: **if** $b > B_{max}$ **then** $b = B_{max}$
7: increase dop of p to increase l_i
8: **end if**
9: **end if**
10: calculate output rate r_o
11: **end for**
12: $C \leftarrow$ all unprocessed nodes with known input rate r_i
13: **for all** $c \in C$ **do**
14: $\text{dop}_c \leftarrow$ calculate dop such that $l_i > ppt$
15: **end for**
16: $P \leftarrow$ all $c \in C$ with outgoing edges
17: **end while**

Demo Architecture



Evaluation