# Massive scale-out of expensive continuous queries E. Zeitler and T.Risch

**Presentation by Thomas Pasquier** 

# Stream splitting

# Splitstream

- splistream(stream s, s int q, function bfn, function rfn)
- user defines rfn (routing function)
- int rfn(int q, tupple t)
- user defines bfn (broadcast function)
- bool bfn(int q, tupple t)



#### Naive implementation



# Tree shapped implemenation: maxtree

#### Scalable Splitting of Massive Data Streams

Erik Zeitler, Tore Risch



Parasplit





# Evaluation: network bound



#### Window router stream rate



If w large enough bound by the network However, performance decrease when p large (author state reason unknown)

# **Evaluation parasplit\***



Le

## **Comparison different solutions**



# Cost model and heuristic

### Cost model for Window router

Cpr = cr + cs + ce

- cr : read cost
- cs : split cost
- ce : emit cost



# Cost window splitter

Cps = crw + cs (o+r+q.b) + ce(r+q.b)



# Cost model for query processor





- Cpr = crw + cs + ce
- Cps = crw + cs (o+r+q.b) + ce(r+q.b)
- Cpq = cr + p(cp+cm) + O

# Heuristic for estimating p

- We search p such that  $p \cdot \Phi_{PS} \ge \Phi_{D}$
- Assume:

- $\circ$  1% broadcast tuples
- $\circ$  0% omitted
- crw = 0
- Cps = crw + cs (o+r+q.b) + ce(r+q.b)

 $\hat{C}_{PS} = \left(cs + ce\right) \cdot \left(0.99 + 0.01 \cdot q\right)$ 

• We estimate cs + ce by measuring the maximum steam rate

$$\hat{p} = \left[\frac{\Phi_D}{\Phi_{PS}^{(1)}} \cdot \left(0.99 + 0.01 \cdot q\right)\right]$$

We can the desired steam rate



$$\eta = \frac{p \cdot C_{PS}}{C_{PR} + p \cdot C_{PS} + q \cdot C_{PQ}^{(O=0)}}$$

 Measurement of the additional work incurred by executing parasplit in comparison to executing a window splitter in a single process



# **Evaluation efficiency**



Efficiency

#### **Related** publications

 Event-based Systems: Opportunities and Challenges at Exascale, Brenna et al., 2009

 stream splitting shown to be a bottleneck

 MapReduce Online, Condie et al., 2010

 does not handle scalable stream splitting

# Thank you

Questions ?