Massive scale-out of expensive continuous queries

E. Zeitler and T. Risch

Presentation by Thomas Pasquier
Stream splitting
Splitstream

- `splitstream(stream 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 shaped implementation: maxtree

Scalable Splitting of Massive Data Streams
Erik Zeitler, Tore Risch

reduce bottleneck, but still problematic
Parasplit

Window router, distribute entire windows

Window splitter
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*

Less degradation when using parasplit*
Comparison different solutions

[Graph showing comparison of different solutions]

- Parasplit*
- Parasplit
- Maxtree
- Fsplit

Max stream rate [Mbps]

q
Cost model and heuristic
Cost model for Window router

\[ \text{Cpr} = \text{cr} + \text{cs} + \text{ce} \]

- cr : read cost
- cs : split cost
- ce : emit cost
Cost window splitter

\[ \text{Cps} = \text{crw} + \text{cs} (\text{o}+\text{r}+\text{q.b}) + \text{ce}(\text{r}+\text{q.b}) \]

- crw: read cost per window
- cs: split cost per tuple
- ce: emit cost per tuple
- o: omit %
- r: routing %
- b: broadcast %
- o + r + b = 100%
Cost model for query processor

\[ C_{pq} = cr + p(cp+cm) + O \]

- \( cr \): read cost per tuple
- \( cp \): poll cost
- \( cm \): merge cost
- \( O \): cost for executing the query and emitting the results
Cost model for parasplit

- $C_{pr} = crw + cs + ce$
- $C_{ps} = crw + cs (o+r+q.b) + ce(r+q.b)$
- $C_{pq} = cr + p(cp+cm) + O$
Heuristic for estimating p

- We search p such that \( p \cdot \Phi_{ps} \geq \Phi_D \)
- Assume:
  - 1% broadcast tuples
  - 0% omitted
  - \( crw = 0 \)
- \( C_{ps} = crw + cs \cdot (o+r+q.b) + ce(r+q.b) \)
- We estimate \( cs + ce \) by measuring the maximum steam rate
- 
  \[
  \hat{c}_{ps} = (cs + ce) \cdot (0.99 + 0.01 \cdot q)
  \]
- We can then estimate \( p \), given the desired steam rate
Efficiency

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

- Measurement of the additional work incurred by executing parasplit in comparison to executing a window splitter in a single process
- Useful work:
  - \( p \cdot C_{Ps} \)
- Overhead:
  - \( C_{Pr} \)
  - \( q \cdot C_{PQ} \) with \( O=0 \)
Evaluation 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 ?