COP 6726: Database Systems Implementation

Spring 2018

Weekly Assignment 8

20-03-2018:

* For column store the performance is 1000x faster than row store
* No SQL 2005-2015
* Not Only SQL hides behind scalability
* Many Column stores can process simple aggregates easily
* They fail in complicated datamining requirements
* Grokit was designed to specifically survive these situations
* Row Store DBs – no thought was given for user defined aggregates
* Row Store even if they support User Defined aggregates typically restrict it to one-page size whereas in Grokit they could 200 GB
* You are also limited to typically one return row
* If everything fails, do string processing in Hadoop
* Grokit takes 3 seconds to start a query
* Generalized Filters
  + Filter can be applied to a row
  + There is only a limited amount of fancy things you can do to a filter.
* Generalized Transformers
  + I want to take 4 column values and combine them in complicated ways and put the result in 5th row.
  + It’s hard in DS to stream this data at fast speed outside.
* GLADE Paper in 2009
* Generalized Input (GI)
  + Load in parallel
* GIST Generalized Iterative State Transformer
* Vectorized Processing

22-03-2018

* Iterator Model
* Lots of DBs use it, Its not fast, Zero Parallelization
* Operators can talk to each other with knowing each other at all.
* Group By is a blocking op, means you cant do other thing until the very end of group by
* Our project is a threadsafe way to communicate between two operators
* We are doing bottom up
* Its impossible to tell the spread if process in parallelism.
* MPP – Multiple processor parallelism
* MPP’s biggest issue is you can virtually never do perfect balancing.