**Deep Bodra**

**5801 1841**

**DBI Week 8(3/16 - 3/20)**

**Detailed explanation of the topics covered in video lecture**

**Video No 29**

1. Insight into Datapath’s GLA’s
   1. Query 1 Result
      1. 1TB data
      2. 150M tuples per second
      3. 55 sec query processing time [Oracle machine takes 140 seconds for the same query]
   2. No packing of tuples
      1. The query 1 has a lot of aggregate functions and this means that the values of attributes will be accessed in large numbers.
      2. This is what slows down most databases
      3. Datapath solves this problem by not packing the tuples into a different data structure because packing and unpacking of tuples is an overhead.
      4. What datapath does is it passes the attribute values of the tuples as arguments.
      5. This makes datapath faster than others
   3. Separation between policy and mechanism
      1. AddState merges the given states into a single state
      2. The act of merging is called mechanism
      3. When the states to be merged are small, AddState is called with a large number of states.
      4. When the states to be merged are large, AddState is called with a relatively smaller number of states.
      5. This decision is decided by another entity of datapath.
      6. This decision is called policy
      7. Thus policy and mechanism are separated
      8. This makes datapath faster than others
   4. Discussion on implementing TOPK using GLA’s (already covered in the previous weekly assignment)

**Video No 30**

1. Templates
   1. They allow you to write code generalized to data types
   2. But why not use virtual functions instead [on top of inheritance]?
   3. Because that level of indirection would incur you a performance hit
   4. Templates is equivalent to code generation for specific data types
2. Templated GLA’s
   1. GLA’s need to be written only once and the template along with macros will take care of different types
   2. C++ templates cannot be used for this purpose as they are very restrictive
   3. So, m4 is used for templating the GLA’s in datapath. For eg, templating input and output parameters, etc
   4. Application of Templated GLA
      1. Multiplexer GLA which takes multiple GLA’s as parameters
3. Brief discussion of code on GLA’s

**Video No 31**

1. Constant state in a GLA
   1. GLA is not a single instance
   2. It runs in parallel and once it has been executed the GLA must be able to figure out whether it should run again or not
   3. If it should be run again then the GLA must carry the some statistics/information with it(like in the case of clustering)
   4. GLA’s have a constant state that is updated at each iteration and carried over to the next iteration
   5. The external system has to set up the GLA at the start with respect to the state parameters and termination condition.
   6. The external system will not have to worry about the execution of GLA and thus GLA can operate independently

--THANK YOU--