Solution for CIS 670 Assignment One

1. *Give a real life example of Big data application that has not been mentioned in the slides. (The slides mentioned CERNs Large Hadron Collider (LHC), Earth Scope, etc). Explain why your example have three Vs as challenges? If your example does not have three Vs, you need to give more than one example to cover them (e.g., one example has the Volume and Velocity challenges, and another example has the Volume and Variety challenges).*

Answer: For example, The National Retail Data Monitor (NRDM) at the University of Pittsburgh monitors sales of over-the-counter (OTC) healthcare products to identify disease outbreaks as early as possible (**Velocity**). The goal of the NRDM project has been to bring this new type of public health surveillance into existence as quickly as possible to meet the nation's need for the early detection of bioterrorism as well as naturally occurring disease outbreaks.

Since its activation with public health departments in December 2002, the number of retail pharmacy, grocery, and mass merchandise operations (**Variety**) that participate in the NRDM has grown to more than 21,000 stores (**Volume**), from among the nation's top thirteen chains. More than 800 public health officials, across 49 states, the District of Columbia, and Puerto Rico, have had access to the system via protected user accounts.

2. *What are the relationships between parallel databases and Big data with respect to three V's?*

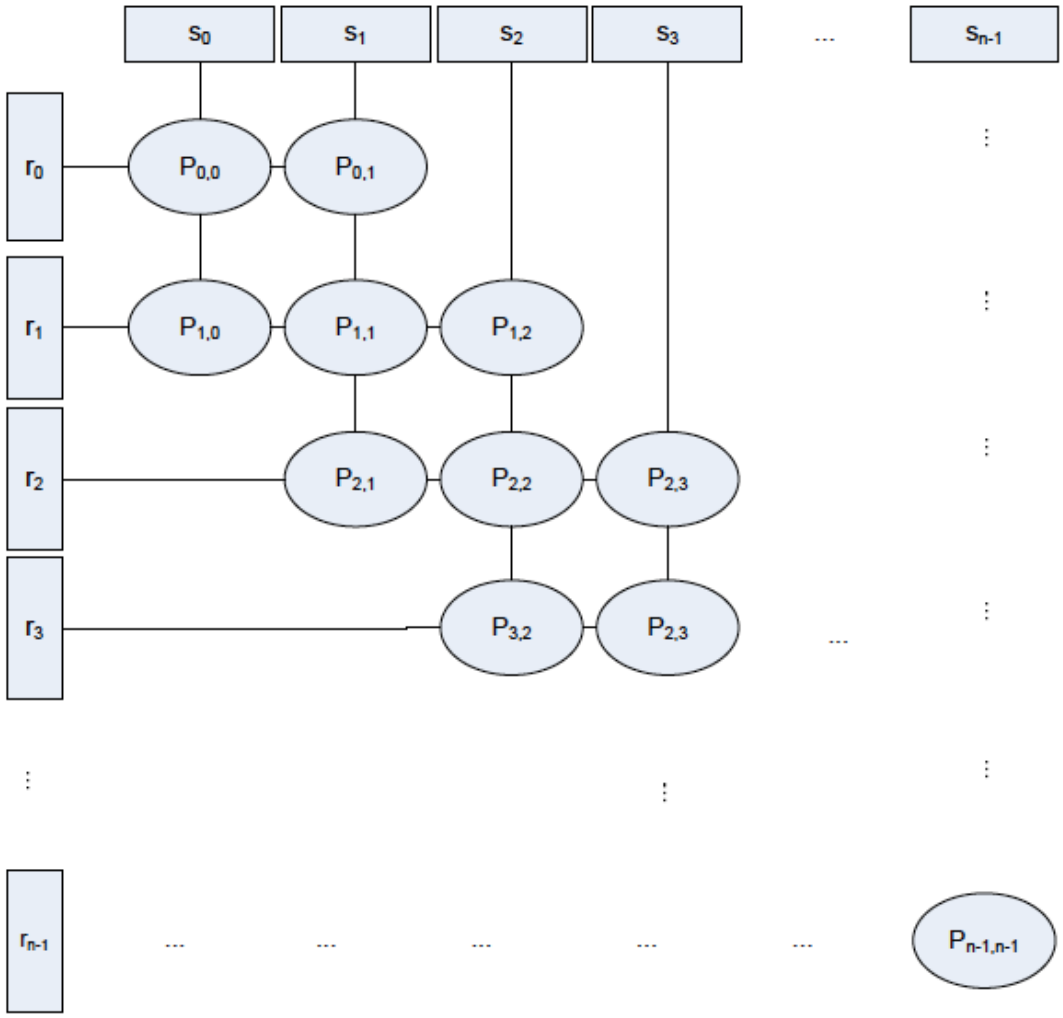
Answer: With regarding to big data, parallel database can:

1. Not necessary to handle better variety.
2. Can improve the velocity (through shared and large number of CPUs).
3. Can improve the volume (through shared memory, disk).

3. *Consider join processing using symmetric fragment and replicate with range partitioning. How can you optimize the evaluation if the join condition is of the form $| r.A - s.B | \leq k$, where k is a small constant? Here,$| x |$ denotes the absolute value of x. A join with such a join condition is called a band join.*

Answer:

We can partition both relation r and s using range partition. Since k is a small constant; it is practical to set up the partition length to be larger than k. Let n be the number of partitions. For each partition r_i and s_i, we allocate copies of r_{i-1}, r_i, r_{i+1} and s_{i-1}, s_i, s_{i+1} on one node. These partitions are joined locally. In total $3n$ processors are required instead of n^2 processors in join with equality ($r.A<s.B$). See the following figure:



4. *Describe a good way to parallelize each of the following:*

a. *The difference operation*

Partition is performed based on the same attributes. Difference operation is performed at each local node for the partitions. Then the results are combined.

b. *Aggregation by the count operation*

Let G be the attribute for **Group by** and A be the attribute for aggregation (**Count**). **Group by** and **Count** is performed locally for attribute A on each node (processor and disk). The partial counts are added up locally at each node to get the final result.

c. *Aggregation by the avg operation*

Each node returns **count** and **sum**. The partial counts and sums are added up together to get the final avg result.

d. Left outer join, if the join condition involves only equality.

Value based partition is performed on the attribute of equality. Left outer join operation is performed locally at each node. Then the results are combined.

e. Full outer join, if the join condition involves comparisons other than equality.

Let the left outer join operation be *l* outer-join *r*. Using fragment and replicate, the left outer joint operation is performed in parallel. Let the set of non-NULL join result be *A*. Let the intersection of NULL tuples from all nodes be B. The result is *A* union *B* because a full outer join returns all of the rows from *l* and *r*. We will have two sets of Null tuples because one from right and another from left. We have to remove Null tuples when report the final result.

5. *The attribute on which a relation is partitioned can have a significant impact on the cost of a query.*

a. Given workload of SQL queries on a single relation, what attributes would be candidates for partitioning?

The attribute that involves in one or more selection, join operation, and group by operation. If we have partitioned data on the group-by attribute or join attributes, we do not need to repartition it again for the query answering.

b. How would you choose between the alternative partitioning techniques, based on the workload?

We can make an estimation based on the partition cost of the data. Heuristic such as known statistics can be used to make the estimation.

c. Is it possible to partition a relation on more than one attribute?

Yes, use hash on combined attributes. If total order is defined on attribute set, range partition can be performed based on the defined total order.