

CS 443 Parallel DB

Winter 2013 Adapted from Suciu & Balazinska

#### Parallel DBMS

- Inter-query parallelism
  - Each query runs on one processor
  - Only for OLTP queries
- Inter-operator parallelism
  - A query runs on multiple processors
  - □ An operator runs on one processor
  - For both OLTP and Decision Support
- □ Intra-operator parallelism
  - An operator runs on multiple processors
  - For both OLTP and Decision Support
  - □ Main parallelism used in parallal DBMS sihce 1980's tibe Good

## Horizontal Data Partitioning

3

- □ Have a large table R(K, A, B, C)
  - Need to partition on a shared-nothing architecture into P chunks R<sub>1</sub>, ..., R<sub>P</sub>, stored at the P nodes
- □ Block Partition: size( $R_1$ ) ≈ ... ≈ size( $R_P$ )
- □ Hash partitioned on attribute A:
  - □ Tuple t goes to chunk i, where  $i = h(t.A) \mod P + I$
- □ Range partitioned on attribute A:
  - □ Partition the range of A into  $-\infty = v_0 < v_1 < ... < v_P = \infty$
  - Equiwidth or equidepth
  - □ Tuple t goes to chunk i, if  $v_{i-1} < t.A < v_i$

UofT:DB Group

## Parallel GroupBy

- □ R(K,A,B,C), how could we compute these GroupBy's, for each of the partitions
- γA,sum(C)(R)
- If R is partitioned on A, then each node computes the group-by locally
- Otherwise, hash-partition R(K,A,B,C) on A,
   then compute group-by locally



# Performance Metric: Parallel DBMS

- P = the number of nodes (processors, computers)
- Speedup:
- More nodes, same data leads to higher speed
- Scaleup:
- More nodes, more data leads to same speed
- OLTP: "Speed" = transactions per second (TPS)
- Decision Support: "Speed" = query time



### Speedup and Scaleup

- □ The runtime is dominated by the time to read the chunks from disk, i.e. size(R<sub>i</sub>)
- If we double the number of nodes P, what is the new running time of γA,sum(C)(R)?
- □ If we double both P and the size of the relation R, what is the new running time?



#### Uniform Data v.s. Skewed Data

- Uniform partition:
- size(R<sub>1</sub>) ≈ ... ≈ size(R<sub>P</sub>) ≈ size(R) / P
- Linear speedup, constant scaleup
- Skewed partition:
- For some i, size(Ri) >> size(R) / P
- Speedup and scaleup will suffer



#### Uniform Data v.s. Skewed Data

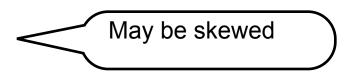
8

- Let R(K,A,B,C); which of the following partition methods may result in skewed partitions?
- Block partition
- Hash-partition
- On the key K
- On the attribute A
- Uniform Assuming perfect uniform hash

  May be skewed

Uniform

- Range-partition
- On the key K
- On the attribute A



Difficult to maintain perfect range-partitioning

11/16/11



## Parallel Join?

9

 $\Box$  R(A,B) join on B with S(B,C)

