CLOUD COMPUTING
CONCEPTS

with **Indranil Gupta (Indy)**

MAPREDUCE

Lecture D

MAPREDUCE FAULT-TOLERANCE

Fault Tolerance

- Server failure
  - NM heartbeats to RM
    - If server fails, RM lets all affected AMs know, and AMs take action
  - NM keeps track of each task running at its server
    - If task fails while in-progress, mark the task as idle and restart it
  - AM heartbeats to RM
    - On failure, RM restarts AM, which then syncs up with its running tasks
- RM failure
  - Use old checkpoints and bring up secondary RM
- Heartbeats also used to piggyback container requests
  - Avoids extra messages

Slow Servers

Stragglers (slow nodes)

- The slowest machine slows the entire job down (why?)
- Due to bad disk, network bandwidth, CPU, or memory
- Keep track of "progress" of each task (% done)
- Perform backup (replicated) execution of straggler task: task considered done when first replica complete. Called speculative execution.

LOCALITY

- Locality
  - Since cloud has hierarchical topology (e.g., racks)
  - GFS/HDFS stores 3 replicas of each of chunks (e.g., 64 MB in size)
    - Maybe on different racks, e.g., 2 on a rack, 1 on a different rack
  - MapReduce attempts to schedule a map task on
    - A machine that contains a replica of corresponding input data, or failing that,
    - On the same rack as a machine containing the input, or failing that,
    - Anywhere

MapReduce: Summary

- MapReduce uses parallelization + aggregation to schedule applications across clusters

- Need to deal with failure

- Plenty of ongoing research work in scheduling and fault-tolerance for MapReduce and Hadoop