**Broadcasting results of Map phase**

* **Broadcast of data** = **overhead** 🡪 how about using a parameter (e.g. estimate probability of failures according to available nodes, set it manually (e.g. 0 – 1, where 0 is keep it locally on one node and 1 means broadcast it to every node, in between broadcast randomly to a number of nodes 🡪 may be refined by estimating the number of nodes needed by taking into account node properties like cpu, age of computer, etc. (**likelihood of failure?** 🡪 **could estimate a likelihood of failure for a node.**)
* Possibility of **directly compressing data??**

**Results**

* If a node has completed a task, it should notify all others about this that other nodes may continue with another job
  + To ensure this behavior, all data should first be transferred (to nodes that are currently not using the bandwidth but are executing tasks) 🡪 needs to know which nodes are executing tasks (e.g. broadcast a “running ThisAndThat task” message)

**Node Bootstrapping**

* On startup, a node receives the current task list (if there are jobs running)
* Else, it just stays there idle for a job to be submitted
* If a second node is connected, it receives the current jobs to transmit
* Simultaneously, a node is an observer and an observable. It observes all the other nodes by storing the information about their context. If a node disconnects and another node detects this, it informs all the other nodes and updates their list