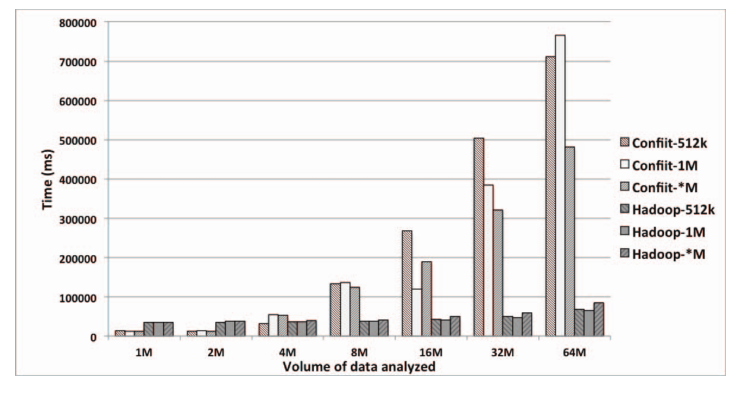
**Performance Evaluation**

1. **Pre-Mare Adaptive Deployment of MR over Pervasive Grids**

* **Wordcount** with Confiit (Predecessor of Cloudfit)
  + Varying Data (512MB, 1GB, 2GB)
  + **3 different input splits for each data size** (512MB, 1GB, 2GB): single file, 1MB, 512KB splits (analyse impact of input files on map step)



1. **CloudFIT, a PaaS platform for IoT applications over pervasive grids:**

* **Wordcount compared to Wordcount in Hadoop on a dedicated (fast) cluster:**
  + 8 Machines, 2.6GHz, 32GB ram, Infiniband QDR Network 40Gbps, Fairness parameters: #parallel tasks per node, Max JavaVM memory
  + Overall execution time (map & reduce phases) of CloudFIT and Hadoop with **Varying**  Data (512MB, 1GB, 2GB)
    - Data: Textbooks from the Gutenberg Project
    - 64MB blocks to reproduce the size of an HDFS data block
    - Results = Average of 10 Executions
  + TomP2P faster than Past faster than Hadoop
* **Wordcount on top of common desktop equipment** (no tuning, anti vir, word etc)
  + 3 Nodes
  + Influences: processor type and speed, but also heavily: NETWORK Speed
    - Writing 64MB of data on DHT with Infiniband interconnection 🡪 average 2seconds, but with wifi (here) it takes 15 secs
  + **Influences:** concurrency between faster and slower nodes: both node types have similar chances to draw tasks in the beginning, but faster node will complete thei tasks earlier and start reexecuting he tasks from slower nodes 🡪 wasting computing resources

1. **P2P MapReduce: Parallel data processing in dynamic cloud environments**

* Use a custom-made, discrete-event simulator (reproduces behavior of the P2P MR prototype)
* **Tests:**
  + **Fault Tolerance: % failed jobs and corresponding % of lost computing time**
    - % of failed jobs increases with higher leaving rate.
  + **Network Traffic**: **# messages and corresponding amount of data in Mbytes** (sum of sizes of messages exchanged through the network)
  + **Scalability**: Varying network size

1. **Towards MR for Desktop Grid Computing**
   1. **Wordcount**
      * Injected faults by killing worker processes, simulate heterogeneity by launching concurrent processes
      * Evaluate
        1. **scalability** when # nodes increases
        2. impact of **varying** **number** of **mappers** & **reducers** (most jobs are map only/map mostly applications (93%, yahoo survey reference [16])
        3. **Fault tolerance** (workers crash at different times) 🡪 again master nodes…
        4. **Laggers effect** (nodes which spend an unusual long time to process the data and slow down the whole computation): more replica, lagger is less problematic
2. **Moon**
   * **Sort and Wordcount** (Inputsize: 24GB and 20GB respectively, inputdata randomly generated with tools from Hadoop)
   * Evaluate:
     1. Execution time
     2. # of duplicated tasks