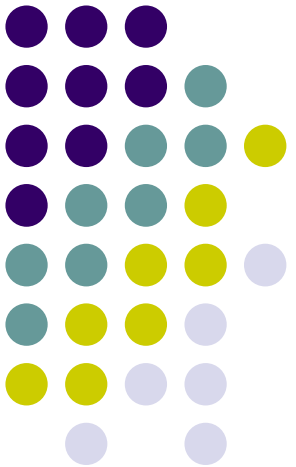
Ubiquitous and Mobile Computing CS 525M: Mobile MapReduce: Minimizing Response Time of Computing Intensive Mobile Applications

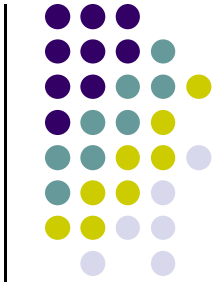
Vijay Sukhadeve

Computer Science Dept.

Worcester Polytechnic Institute (WPI)

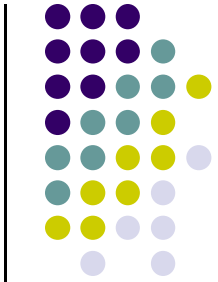


Introduction/motivation



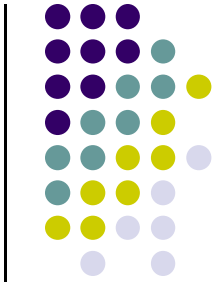
- overall users' response time due to network problems
- minimize users'response time
- outsourcing to nearby residential computers vs public clouds
- to build Mobile MapReduce (MMR)
- MMR to leverage the best computing resources to conduct computation.
- Apps : text searching, face detection and image processing

Smartphone Constraints



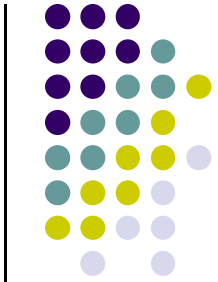
- Processing power Lagging behind due to size and weight
- Limited battery power, additional consumption due to sensors

To show



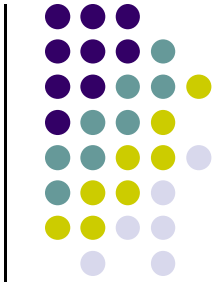
- Outsourcing to appropriate resources more advantageous
- Speed up computing – use parallel processing techniques

Idea



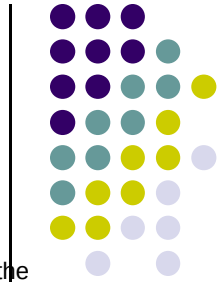
- Based on Original MR framework design and implement MMR
- Scheduling Model – Dynamically leverages best computing resources – residential computers vs clouds
- Results : outperforms on-device computing Response time: 15 times improvement , battery consumption: 20 times improvement

Path



- leveraging residential computers and MapReduce
- design of MMR
- mobile MapReduce implementation
- evaluation results
- Some related work

Nearby Computers vs. Public Clouds



Experiment: find a string in a text file: response time is longer than if the job is outsourced to nearby residential computers because of the impact of network latency

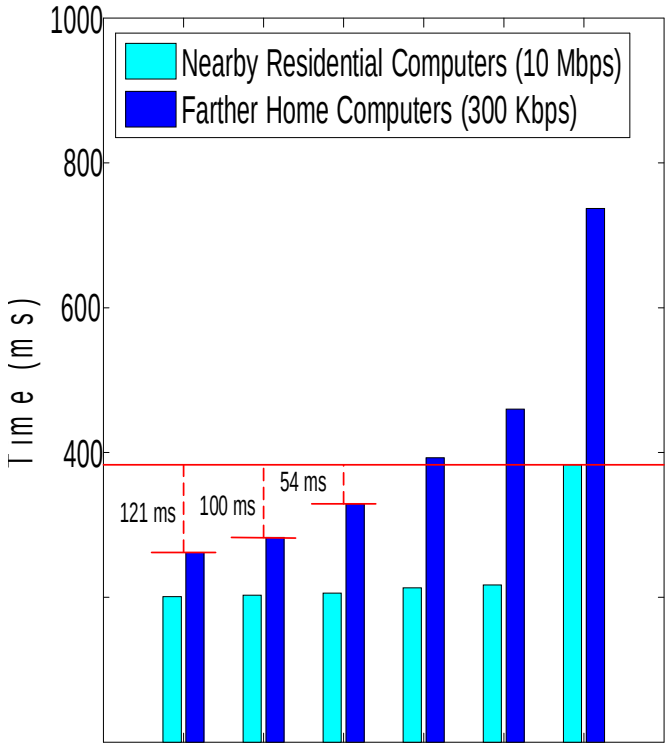
Table 1. Response Time (Sec)

File Size (KB)	Android	Amazon EC2	Residential Computers
10	0.0481	0.0146	0.0459
100	0.425	0.096	0.4245
200	0.424	0.971	1.300
400	0.465	1.600	1.300
750	0.480	3.400	3.300
1000	0.503	4.500	6.600

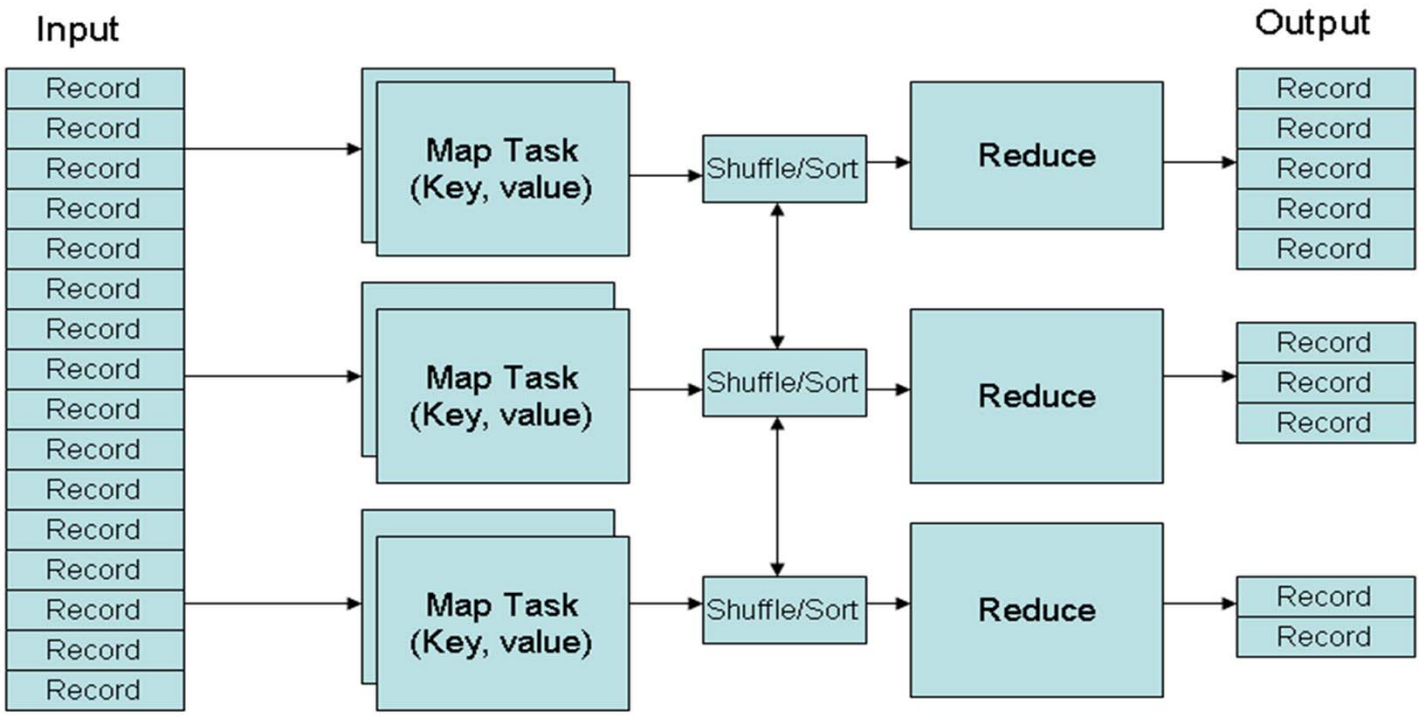
Table 2. Energy Consumption (J)
energy consumed on the mobile device

File Size (KB)	Android	Amazon EC2	Residential Computers
10	0.117	0.098	0.122
100	0.332	0.984	1.995
200	0.886	1.815	4.099
400	1.327	3.603	8.235
750	02.467	6.637	15.366
1000	3.092	8.823	20.583

band- width consumption has to be taken into consideration outsourcing to nearby residential computers is faster

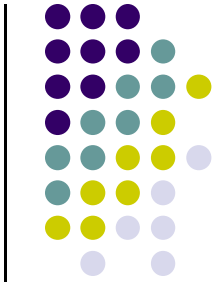


MapReduce Phases



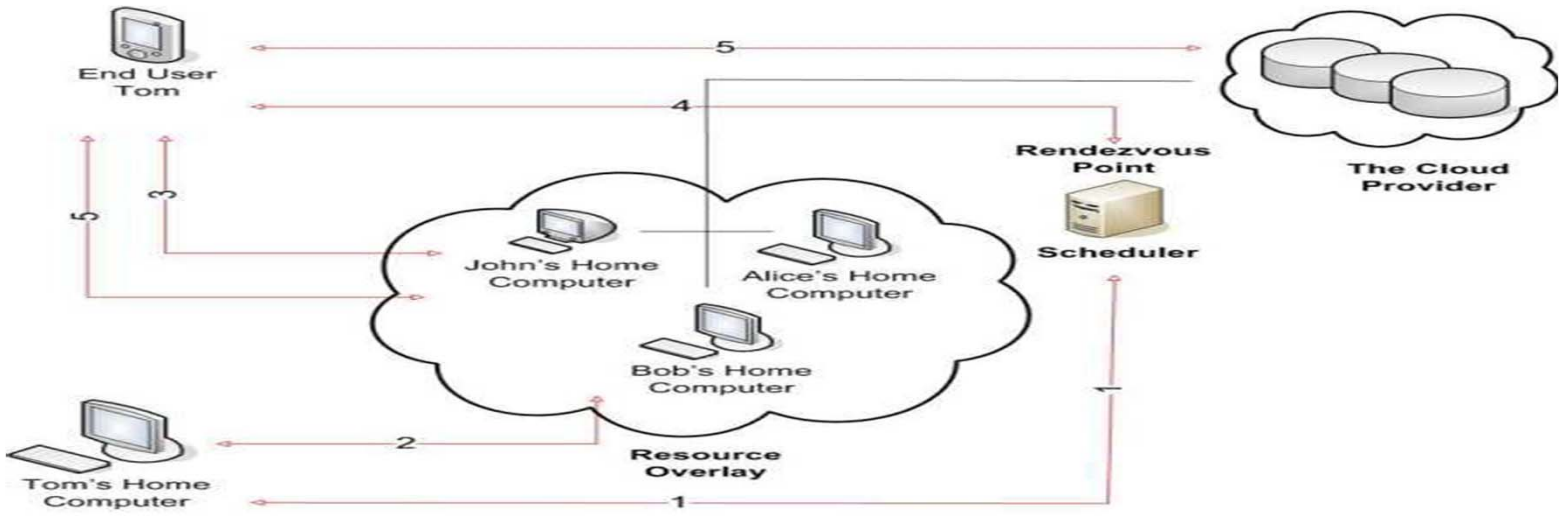
Deciding on what will be the key and what will be the value ➔ developer's responsibility

Reasons for Modification



- Map and Reduce nodes are connected to each other, which is not always possible in our mobile computing environment
- HDFS in the original MapReduce contains the data prior to the job submission and computation, which is less likely to be practical in our mobile computing environment
- data size in our mobile computing is relatively small

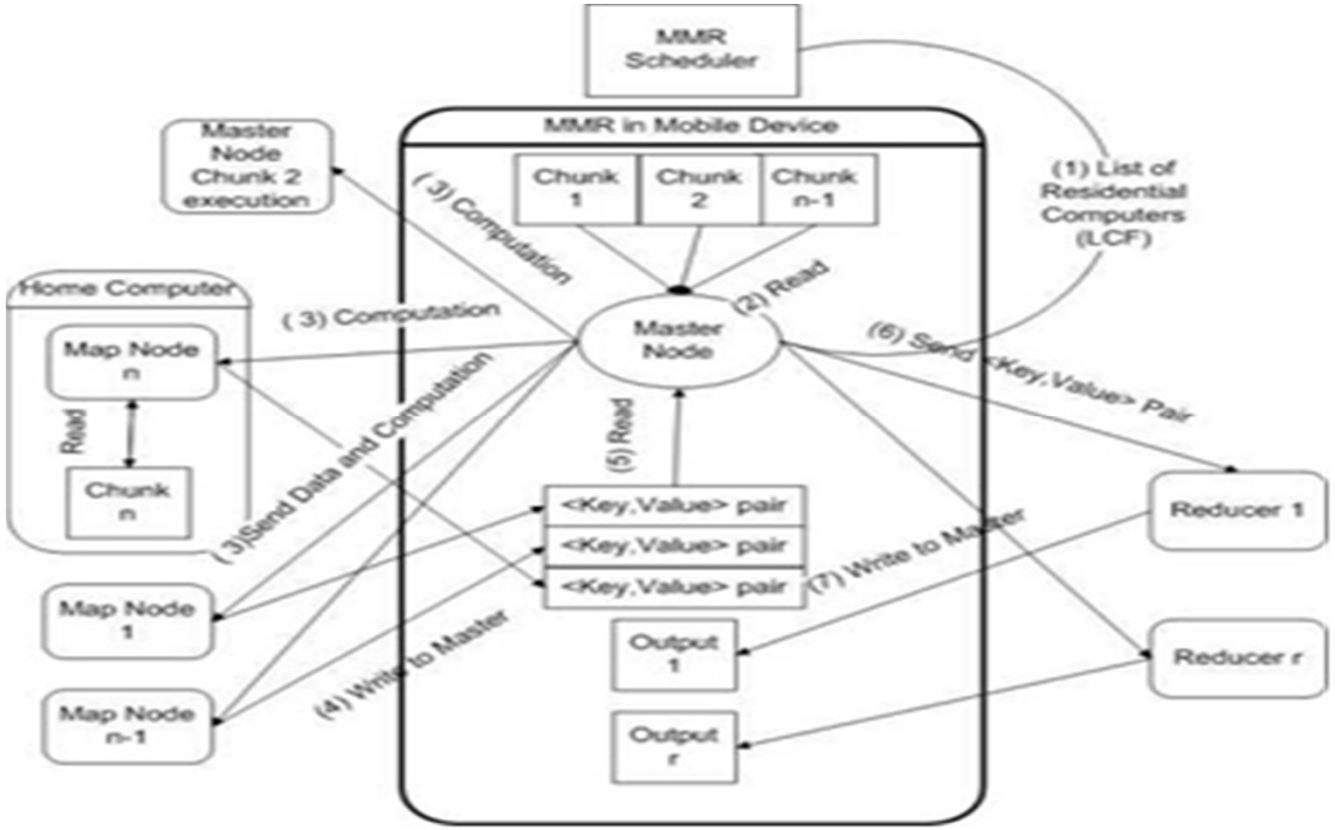
Architecture of MMR



Resource Overlay – Users residential computers plus public cloud

MMR then submits the job to an appropriate set of computers

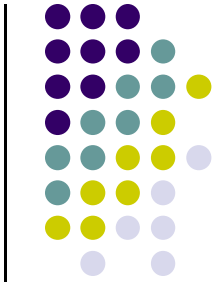
MMR Workflow



Mobile Device – Master Node

Residential computer – worker node and may work as Map and reduce

Mobile MapReduce Implementation

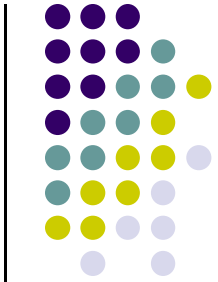


- Dynamic Mobility Property of MMR

mobile users without persistent connections and the master node does not have any knowledge about the neighboring worker nodes.

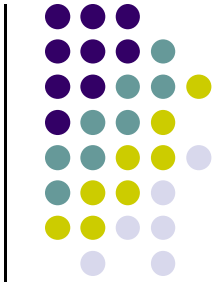
- Non-Distributive File System of MMR

selected nearby residential computers do not have a copy of the input file until the file is transferred there



- Handling Isolated Worker Nodes
- In our framework, the Map node sends the list of<Key,Value> pairs to the master node who eventually forwards to the Reducers.
- Node Failure
- As the input data is partitioned into small independent chunks, the failure of any worker causes only re-execution of that portion of data.

Mobile MapReduce Implementation



- Handling Isolated Worker Nodes
- In our framework, the Map node sends the list of<Key,Value> pairs to the master node who eventually forwards to the Reducers.
- Node Failure
- As the input data is partitioned into small independent chunks, the failure of any worker causes only re-execution of that portion of data.

Preliminary Evaluation
3 user model and they have identical residential computers and mobile devices

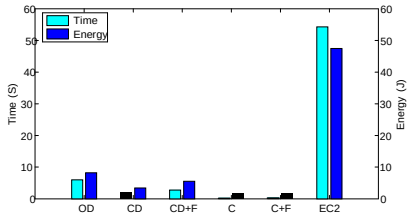
- Text Search
- Face Detection
- Image Sub Pattern Search

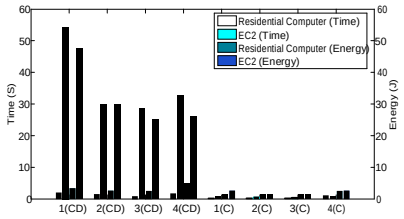
Experimental Results
repeat each experiment five times and present the average of the results

- Text Search

a)outsourcing to EC2 results in the worst performance in terms of both the response time to the user and the amount of energy consumed

b) computation is parallelized among multiple machines shows that the response time and energy consumption first decrease with the increase of parallelization level





(a) Response Time and Energy Consump- (b) Response Time and Energy Consump-

References

- 1. AWS SDK for Android. http://aws.amazon.com/sdkforandroid/.
- 2. BlastReduce: High Performance Short Read Mapping with MapReduce. www.cbcb.umd.edu/software/blastreduce/.
- 3. Cross Correlation . http://en.wikipedia.org/wiki/Cross-correlation.
- 4. Diamedic. Diabetes Glucose Monitoring Logbook. http://ziyang.eecs.umich.edu/projects/powertutor/index.html.
- 5. International Data Corporation : Press Release 28 Jan and 4 Feb, 2010. http://www.idc.com/.
- 6. International Telecommunication Union : Press Release 10 June, 2009. www.itu.int.
- 7. iPhone Heart Monitor Tracks Your Heartbeat Unless You Are Dead. giz- modo.com/5056167/.
- 8. Mint. http://www.mint.com/.
- 9. Power Tutor. http://ziyang.eecs.umich.edu/projects/powertutor/index.html.
- 10. Rajesh Balan, Jason Flinn, M. Satyanarayanan, Shafeeq Sinnamohideen, and Hen- I Yang. The case of cyber foraging. In Proceedings of the 10th ACM SIGOPS European Workshop, Saint-Emilion, France, July 2002.
- 11. Rajesh Krishna Balan, Darren Gergle, Mahadev Satyanarayanan, and James Herb- sleb. Simplifying cyber foraging for mobile devices. In Proceedings of The 5th In- ternational Conference on Mobile Systems, Applications, and Services (MobiSys), San Juan, Puerto Rico, June 2007.
- 12. Byung Gon Chun and Petros Maniatis. Augmented smartphone applications through clone cloud execution. In Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS), Monte Verit, Switzerland, May 2009.
- 13. Pierluigi Crescenzi and Viggo Kann. A compendium of NP optimization problems.
- 1998.
- 14. Eduardo Cuervo, Aruna Balasubramanian, Dae ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. MAUI: Making smartphones last longer with code offload. In Proceedings of The 8th International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, USA, June 2010.
- 15. Jeffrey Dean and Sanjay Ghemaawat. Mapreduce a flexible data processing tool.
- In Communication of the ACM, Jan 2010.
- 16. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI), San Francisco, CA, Dec 2004.
- 17. Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In Proceedings of The 6th International Conference on Mobile Systems, Applications, and Services (MobiSys), Breckenridge, Colorado, June 2008.