Grid Computing In Smartphones

Vishalan Gharat, Aishwarya Chaudhari, Jaspreet Gill, Shweta Tripathi

***Abstract*—The overall speed of execution of a task, which is considerably reduced due to the typical limitations of mobile devices can be efficaciously increased by aggregating the capabilities of distributed handhelds. Through an Ad-Hoc Network, task and data parallelism can be achieved, thus providing potentially faster solutions and reduced battery drain. Grid computing is basically a type of parallel and distributed system that enables sharing, selection, and aggregation of geographically distributed autonomous resources. As the number of participating devices in the grid increase, the computation time is reduced significantly. This paper proposes a system that uses the most popular smartphone OS "Android", which is based on the Linux platform and prove the fact that the total computational time is indeed reduced under this implementation, as compared to local computation on a single device.**

***Index Terms*— Architecture, Data sharing, Collaboration, Performance evaluation**

I. INTRODUCTION

GRID computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in which grid computers have each node set to perform a different task/application. The proposed system focuses or the use of a computational grid as the application of resources of multiple computers in a network to a single complex problem at the same time, while crossing theoretical boundaries. Grid computing involves organizationally-owned resources: supercomputers, clusters, and PCs owned by universities, research labs, and companies. These resources are centrally managed by IT professionals, are powered on most of the time, and are connected by full-time, high-bandwidth network links. There is a symmetric relationship between organizations: each one can either provide or use resources. Malicious behavior such as intentional falsification of results would be handled outside the system, e.g. firing the perpetrator [4].

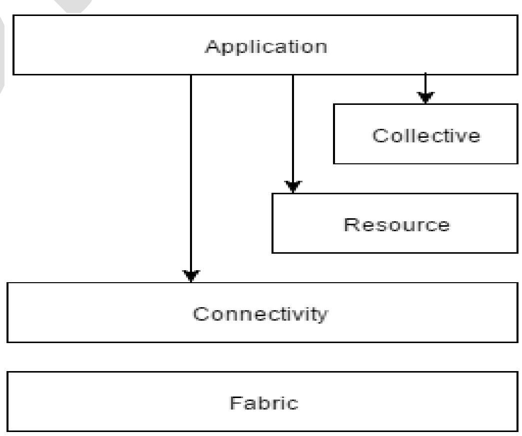
The Globus Project defines Grid as "an infrastructure that enables the integrated collaborative use of high end computers networks, databases and scientific instuments owned and managed by multiple organizations" [1]. The capability of modern computer systems and computer networks has increased exponentially as compared to traditional computer systems. This increase in their performance, most of the times lead to wastage of computational resources because most of the time the CPU sits idles. Grid utilizes this idle CPU cycles to perform the computation when requested by the grid users, which otherwise would have been wasted. This enables the users to perform complex computations that in traditional cases would have required large-scale computing resources e.g. image rendering, scientific researches, climatology, etc[2].

Android is a mobile operating system developed by the Open Handset Alliance for portable devices [3]. Wireless Ad-Hoc networks are typically dynamic and scalable, owing to the mobility of the devices and and decentralized management. The limitations of these devices such as limited power (battery) and small memory size can be overcomed by connecting multiple devices in a grid to achive a common objective.

1.1 Grid Architecture

In a grid computing environment, nodes might not be placed at common physical location but can be independently operated from different locations. Each computer on the grid is a distinct computer [9]**.** Collection of servers clustered together to work out a common problem forms a grid [10]**.** The computers joined to form a grid may even have different hardware and operating systems.

Fig.1 Grid Architecture



Grid consists of a layered architecture model providing protocols and service at five different layers represented by Fig.1.

Fabric layer: Fabric layer sits at the bottom of the grid layered architecture. It provides shareable resources such as network bandwidth, CPU time, memories, scientific instruments like sensors, telescope, etc. Data received by sensors at this layer can be transmitted directly to other computational nodes or can be stored in the database over grid. Standard grid protocols are responsible for resource control. Accomplishment of sophisticated sharing operation is the measure for quality of this layer. Operating system, queuing systems and processing kernels also form the part of this layer.

Connectivity layer: This layer specifies the protocols for secure and easy access. Protocols related to communication and authentication required for transactions are placed in this layer. These communication protocols permit the exchange of data between the resource layer and fabric layer.

Resource layer: This layer specifies the protocols for operating with shared resources. Resource layer build on the connectivity layer's communication and authentication protocols to define Application Program Interfaces (API) andsoftware development kit (SDK) for secure negotiation, accounting, initiation, control, monitoring and payment of sharing resources. E.g. GRIP (Grid resource Information Protocol; based on LDAP).

Collective layer: This layer consists of general purpose utilities. Any collaborative operations in the shared resources areplaced in this layer and it coordinates sharing of resources like directory services, co-allocation, scheduling, brokering services, monitoring and diagnostic services, data replication services.

Application layer: At the top of the grid layered architecture sits the application layer. This layer consists of application which the user will implement, and provides interface to the users and administrators to interact with the grid.

II. LITERATURE SURVEY

- Porting BOINC to android (SETI@home) based on AVRF – Android Volunteered Resource Framework [4].
- Mobile OGSI.NET extends this implementation of grid computing to mobile devices [6], [7], and [5].
- Monte Carlo Technique of π estimation process distributed over several smartphones.

2.1 Porting BOINC to android (SETI@Home) based on AVRF – Android Volunteered Resource Framework

2.1.1 Boinc FrameWork

BOINC (Berkeley Open Infrastructure for Network Computing) is a platform for public-resource distributed computing. BOINC is being developed at U.C. Berkeley Spaces Sciences Laboratory by the group that developed and continues to operate SETI@home. BOINC is open source and is available at http://boinc.berkeley.edu. Public-resource computing and Grid computing share the goal of better utilizing existing computing resources. However, there are profound differences between the two paradigms, and it is unlikely that current Grid middleware will be suitable for public-resource computing. [4], [8].

2.1.2 Primary Goal

BOINC's general goal is to advance the public resource computing paradigm: to encourage the creation of many projects, and to encourage a large fraction of the world's computer owners to participate in one or more projects [4]. Specific goals include:

1. Reduce the barriers of entry to public-resource computing.
2. 2.Share resources among autonomous projects.
3. 3.Support diverse applications
4. 4.Reward participants.

2.2 Mobile OGSI.NET

Mobile OGSI.NET extends an implementation of grid computing, OGSI.NET, to mobile devices. Mobile OGSI.NET addresses the mobile devices' resource limitations and intermittent network connectivity, factors which differentiate them from traditional computers[5].

Mobile OGSI.NET aims to construct a platform that provides a better potential for collaboration among mobile devices, support this style of collaboration among mobile devices with traditional, non-mobile workstation computers and server computers, the collaboration architecture should operate on many device platforms, and finally, the collaboration architecture must address the particular characteristics of mobile devices.

The Mobile OGSI.NET architecture consists of three main layers: Monash University Mobile Web Server, the Grid Services Module, and the Grid Services. Each layer handles a separate concern. The Mobile Web Server handles endpoint to endpoint message reception and transmission. The Grid Services Module handles Grid Services message parsing and multiplexes messages to the appropriate Grid Service. The Grid Service handles application logic and processing. [11]

Mobile OGSI.NET bridges two disparate fields, high performance super comuting designs and personal mobile devices. This benefical investigation gives way to hosting applications that can interoperate with a spectrum of computing resources.

2.3 Monte Carlo Technique of π estimation process distributed over several smartphones connected using Bluetooth network.

The Monte Carlo method for ¼ estimation served as a experimental application for distributed computing with Android devices. Leveraging Bluetooth wireless technology to establish a low power ad hoc network, multiple mobile systems can collaborate in performing a collective computation. The method used to estimate ¼ followed the implementation of the popular random darts method. This method allows for an approximation of ¼ to be calculated by throwing darts randomly at a hypothetical dart board. Imagine a unit circle circumscribed by a square. By randomly throwing darts, hits inside the circle and square will be proportional to the respective area of each part, which can be writtenas

$$\frac{ndc}{nds} = \frac{ac}{as} = \frac{\pi r^2}{4r^2}$$

where ndc is the number of darts in the circle, nds is the number of darts in the square, ac is the area of the circle, as is the area of the square, and r is the radius of the circle. After substituting the number of darts in the square with the total number of throws, solving for ¼ leads to the following equation:

$$\pi = 4 * \frac{ndc}{nt}$$

Where nt is the number of dart throws. For this Monte Carlo method, the value of ¼ becomes more precise as the iteration count increases.

III. Proposed System

- This paper will prove the fact that the total computational time is actually reduced under this implementation compared to local computation on a single device, and as the devices increase the computation time is reduced significantly. The proposed system uses the most popular smartphone OS "Android", which is based on the Linux platform.
- The application developed aims to:-
  Prove the efficiency and reliability of Mobile Grid Computing which includes:

I. Selecting an Algorithm or Program which can be successfully distributed, processed in parts and prove the concept.
II. Opting a reliable and power efficient mode of communication.
III. Including any number of devices in the Grid.
IV. Designing a Grammar or algorithm to divide the selected program in subtasks and integrate the result.
V. Maintaining connectivity throughout the process.

3.1 The target Program: Prime Number Count Identifier

- The target program must be iterative or repetitive and it must have an exponential computation task, as such algorithms provided accurate estimations in grid computing applications.
- This paper presents Prime Number Count Identifier as the target algorithm, which functions to the number of prime numbers in a given specified range of integral numbers.
- The results of the program are integrable at the end. The program has inner and outer bounds which can be varied to check the performance under different task size or execution times.

3.2 Mode of Communication:AD-HOC network

- The mode of communication scheme is an AD-HOC network. Since, there is no need of extra hardware, like a router.
- WIFI is more optimal than Bluetooth, considering the speed of communication.
- The system will use direct WiFi to connect several devices to a single master, which will be selected according to the hardware specifications,viz: the device with betterspecifications (more powerful specifications or more cores) will be assigned a computationally tougher task.

3.3Designing a Grammar for Division and Distribution

- This stands as the most crucial part of the overall proposed project.
- It involves retrieving the hardware specifications of all the devices connected in the network.
- Ordering them according their performance considering the following parameters:-
  - Frequency of Processor
  - Main Memory available(RAM)
  - Number of Cores of processor
- The next includes dividing the Task into the subtasks according to the complexity, considering the number of devices connected in the network.
- Further, assigning the subtask to the devices according to the order of their hardware specifications.
- And finally, collecting the results from each device and integrating it to derive the final output.

3.4 Major Challenges

- Taking into account all the devices available in the android eco-system.
- The formation of Cluster Network.
- Assigining tasks according to the hardware specification of each device on the network.
- Integration of results on the host device, and computational analysis.

IV. Design

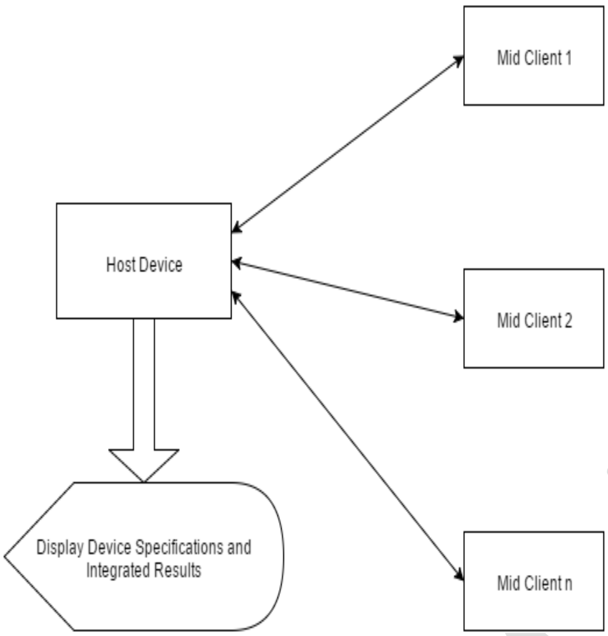
4.1 Prime number count identifier

- Checking a number is prime or not involves checking divisibility of that number by all the numbers between the number two and sqrt(number) viz- Square root of the number.
- Considering this, given a range to find all the numbers in it, we need to check all the numbers in the range.
- As the value of the number goes on increasing, even its square root value increases hence we need to di-

vide and check it by all the numbers till its square root.

- Thus, prime number generation is the best example of exponential algorithm, and such algorithms are most suited for demonstrating grid computing on a device.

Prime number count identifier

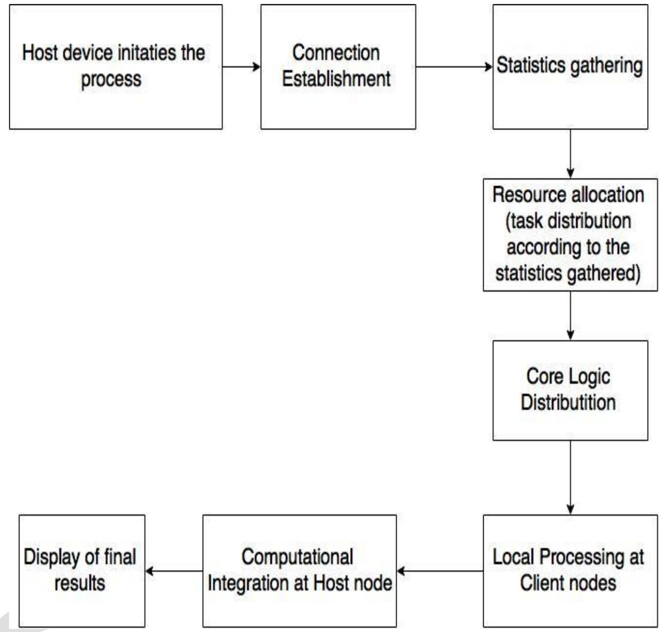


4.1.1 Component Modules

- Host Device-
  - a) Responsible for initiating the ad-hoc connection to the client devices.
  - b) Determines the specifications of the participating devices in the grid.
  - c) Distributes the task amongst the devices with respect to their specifications.

- Integrates the individual results obtained from the client devices to produce a unified result. Client Devices-
  - a) Participate in the grid and by coordinating with the host device
  - b) Assist the host device in establishing a secure and reliable connection using Wi-Fi Direct.
  - c) Successfully perform the allocated task and submit the result to the host device.

4.2 Flow diagram



4.2.1 Steps Involved

a) The host device initiates the connection process using Ad-hoc network.
b) Once the connection is established, the host device collects the statistical specifications of the component devices (participating devices in the grid.)
c) Resources are allocated to the devices (tasks to be computed) in accordance with statistics gathered previously.
d) Core Logic Distribution
e) Local computation of the task is done at client nodes.
f) Results of Computations at individual client nodes are integrated at the host node.
g) Displaying the final results.

4.2.2 Hardware and Software Requirements

Hardware Requirements:

- Sufficient number of smartphones Of varying specifications
- Wi-Fi

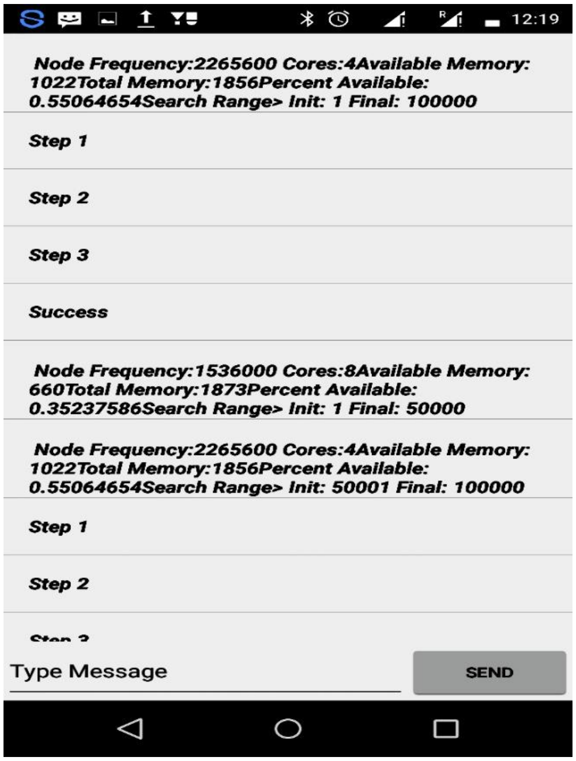
Software Requirements:

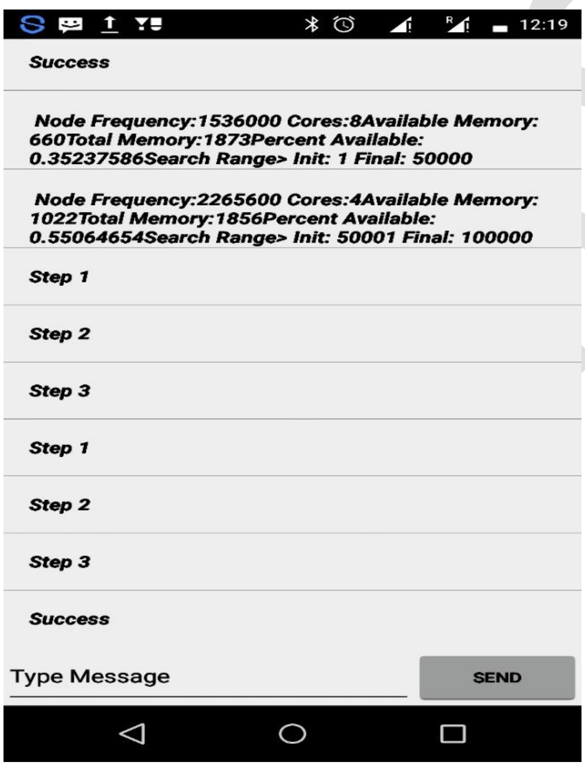
- Android OS

VI. IMPLEMENTATION RESULTS

5.1 Screenshots

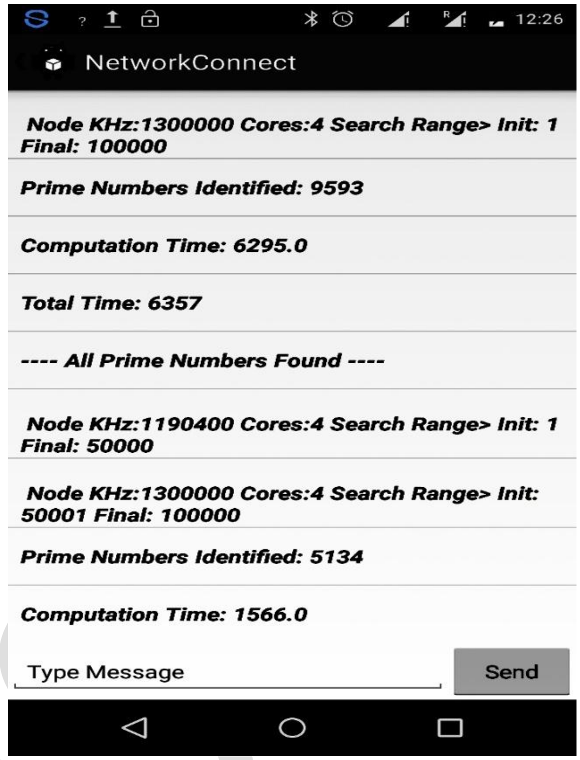
Phase 1: Retrieval of client device specifications





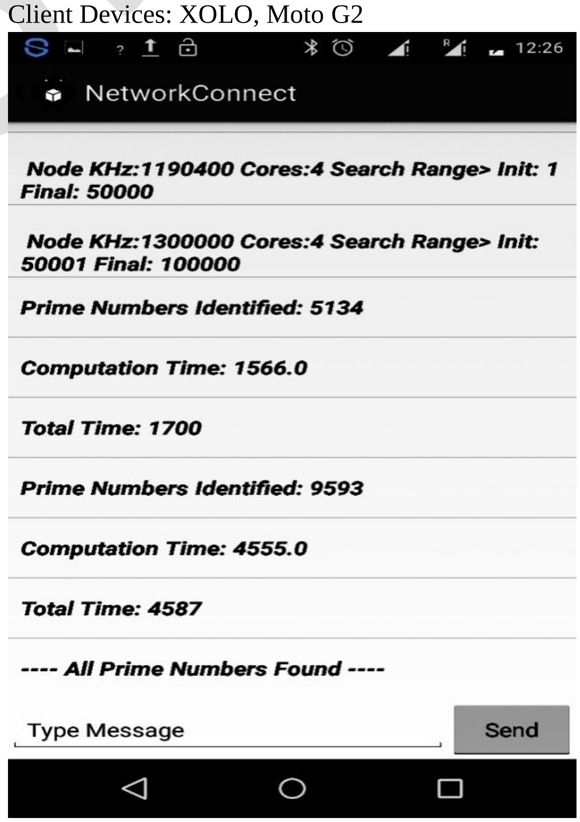
Phase 2: Grid Computation Statistics

a. Number of clients in the grid: 1
Host device: Moto G1



Client Device: XOLO

b. Number of clients in the grid: 2
Host device: Moto G1
Client Devices: XOLO, Moto G2



Specifications of devices used in the grid:-

i) XOLO q1010i:-

OS: Android OS, v4.2 (Jelly Bean)
CPU: Quad-core 1.3 GHz Cortex-A7
Memory:-Internal: 8GB, 1GB RAM.

ii) Moto G2:-
OS: Android OS, v4.4.4 (KitKat)
CPU: Quad-core 1.2 GHz Cortex-A7
Memory:-Internal: 8GB, 1GB RAM.

iii) Moto G1 (Host):-
OS: Android OS, v4.3 (Jelly Bean)
CPU: Quad-core 1.2 GHz Cortex-A7
Memory:-Internal: 8GB/16GB, 1GB RAM.

iv) YU Yureka:-
OS: Android OS, v4.4.4 (KitKat)
CPU: Quad-core 1.7 GHz Cortex-A53
Memory:-Internal: 16GB, 2GB RAM

v) Xperia Z:-
OS: Android OS, v4.1.2 (Jelly Bean)
CPU: Quad-core 1.5 GHz Krait
Memory:-Internal: 16GB, 2GB RAM

vi) Nexus 5:-
OS: Android OS, v5.0 (Lollipop)
CPU: Quad-core 2.3 GHz Krait 400
Memory:-Internal: 16/32GB, 2GB RAM.

5.2 Grid computation Results

a) Individual node Computation time results for N devices in grid (N=1 to 5)

Number of Clients or Device Name	1 Client	2 Clients	3 Clients	4 Clients	5 Clients
XOLO Q1010i	6295	4555	3127	1234	805
Moto G2		1566	705	420	290
Yureka			1598	942	634
Xperia				2167	1338
Nexus 5					1417
Moto G1	Host	Host	Host	Host	Host

b) Individual node Total time results for N devices in grid (N=1 to 5)

Number of Clients or Device Name	1 Client	2 Clients	3 Clients	4 Clients	5 Clients
XOLO Q1010i	6357	4587	3148	1253	823
Moto G2		1700	756	500	357
Yureka			1641	974	751
Xperia				2706	1665
Nexus 5					1888
Moto G1	Host	Host	Host	Host	Host

c) Overall grid total time results for N devices in grid (N=1 to 5)

Number of Clients or Total time	1 Client	2 Clients	3 Clients	4 Clients	5 Clients
Total time	6357	4587	3148	2706	1888

d) Individual Latency results for N devices in grid (N=1 to 5)

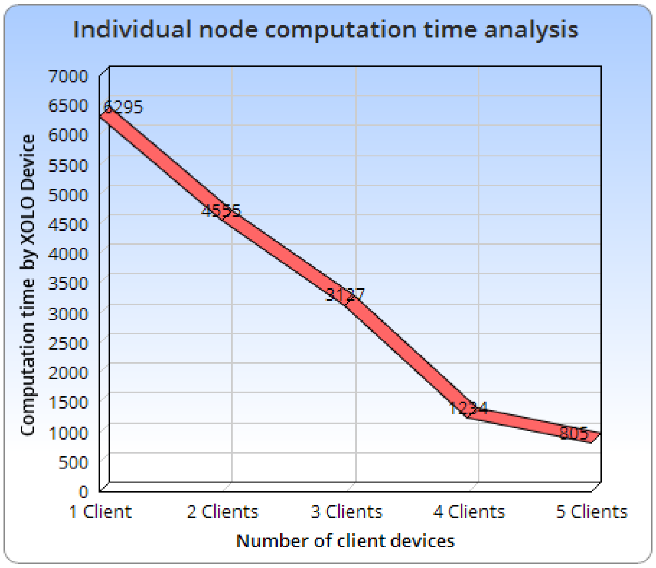
Number of Clients or Latency	1 Client	2 Clients	3 Clients	4 Clients	5 Clients
Latency	62	32	21	19	18

*Note: - The numbers indicate the time in miliseconds.

The tables above represent the actual statistical results obtained on ‘android’ Operating System based smartphones, on connecting the devices of varying specifications in a grid. The android device Moto G1 is the host device in the grid and the rest XOLO, Moto G2, Yureka, Xperia, Nexus 5 are the participating client devices.

4.3 Graphical Analysis

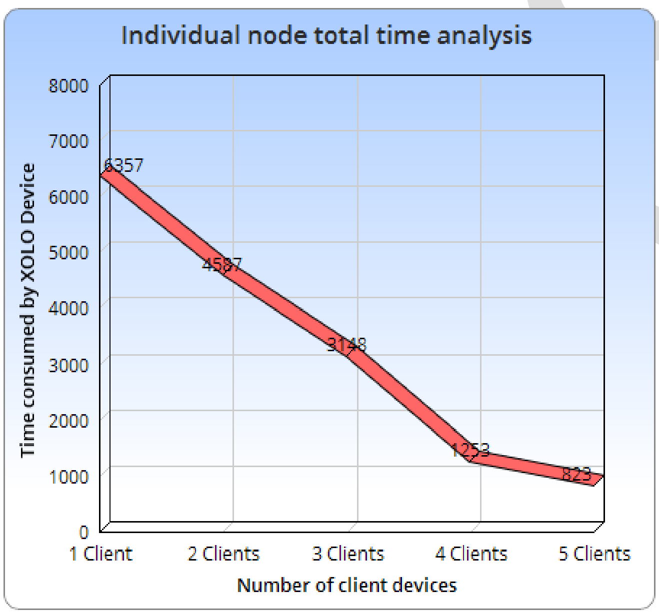
4.3.1 Individual node computation time analysis.



Line Graph

The individual node computation time analysis shows that the computation time of the XOLO smartphone is significantly reduced as the number of client devices in the grid increases.

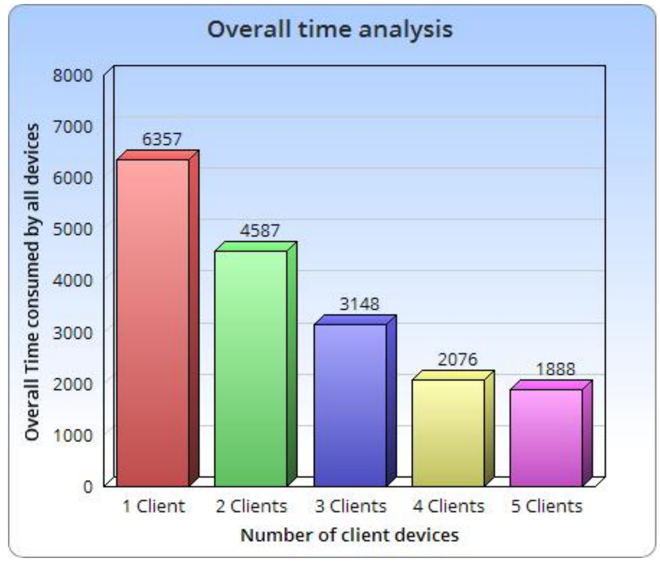
4.3.2 Individual node total time analysis



Line Graph

The individual node total time analysis distinctively points to the fact that the total time of the XOLO smartphone is significantly reduced as the number of client devices in the grid increases.

4.3.3 Overall time analysis

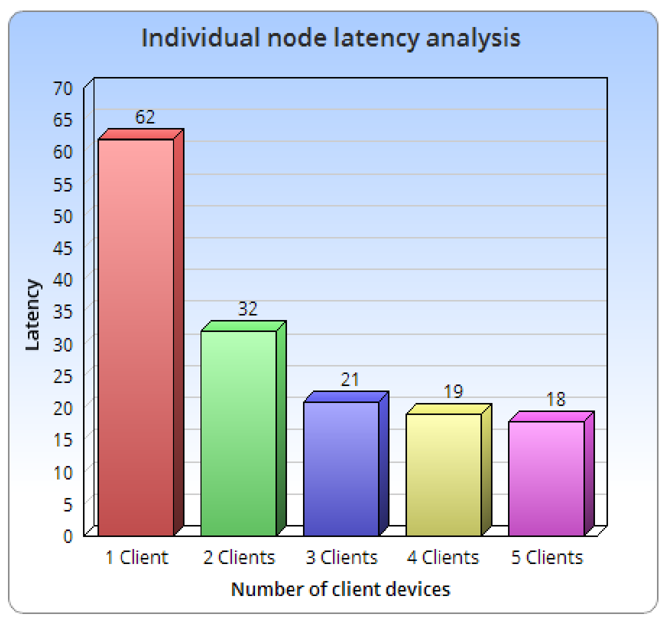


Bar Graph

The Overall time analysis shows that the overall time of computation of the entire grid is reduced as the number of client devices in the grid increases.

Note:- The overall time is not the sum of the individual nodes total time taken by each device, but rather the highest value of all the values of the individual total times of all the client devices in the grid, as any task consuming lesser time with be completed by the time the task taking the maximum time is completed.

4.3.4 Latency analysis



Bar Graph

The Latency analysis depicts the exponential decrease in the latency factor as number of client devices in the grid is increased, thus indicating a reduction in the overall task processing time.[11].

V. Conclusions

5.1 Grid computation ResultsMultiple devices on the go can be connected to an ad-hoc network to complete a task in lesser time

1. Compression and Decompression of files
2. Rendering Multimedia files
3. Improve a particular application performance on Targeted phone.

5.2 Graphical and Visual Enhancements

1. The multimedia content available now a day is getting more and more graphic intensive, i.e. 4k video, Games etc.
2. Using grid computing, we can distribute the processing over multiple devices to improve performance on a single target device.

5.3. More advance applications

- The scenario where in, soldiers in a patrol may have smartphones that are connected via a mesh network to perform processing, i.e. breaching enemy network to retrieve data.
- Routing multiple soldiers, to a single destination by finding shortest path and intermediate checkpoints.

There are Grid projects in areas such as:

- Biomedical research
- Industrial research
- Engineering research
- Studies in high energy Physics
- Studies in Chemistry

Originating in the early 1990's as a metaphor for making computer power as easily accessible as electric power, the term 'Grid Computing', though in its sprouting stage as of today, stands to possess immense potential in the upcoming era of technology. Certain hitches that stand in the way include security issues, development and worldwide adoption of standards for representing and executing applications and workflows, dynamic services composition and aggregation, and realization of the monumental efficacy of computation using grid technology. Along with these measures, a conducive, potent and economical business model has to be developed, so that all involved parties gain the benefits of adopting a Grid technology. With tremendous research work going on and the standards of grid technology continuously evolving, it can be rightly assumed that in the near future, grids will be based on Service Oriented Architecture and would be available as a utility with demand and supplydriven by an economic market, just the very essentials of water and electricity!

Acknowledgment

This work was guided by Mrs.Shweta Tripathi (assistant professor), Mr.Harish Kaura (HOD), and Mr.Khot (Principal). The work was supported by the Department of Computer Engineering, Father Conciecao Rodrigues Insitute of Technology, Navi Mumbai, India.

References

[1] R.Esposito, P.Mastroserio et al., (March 2003) "Standard FTP and Grid FTP Protocols for International Data Transfer in Pamela Satellite Space Experiment". [Online]. Available: http://arxiv.org/ftp/hepex/papers/0305/0305084.pd

[2] Grid Computing: An Introduction,Dabas et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(3),
March - 2013, pp. 466-470

[3] Open Handset Alliance http://www.openhandsetalliance.com/

[4] BOINC: A System for Public-Resource Computing and Storage David P. Anderson Space Sciences Laboratory, University of California at Berkeley davea@ssl.berkeley.edu

[5] Mobile OGSI.NET: Grid Computing on Mobile Devices* Appears in the Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (associated with Supercomputing 2005), November 8, 2004. Pittsburgh, PA.

[6] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm., D. Snelling, and P. Vanderbilt. "Open Grid Services Infrastructure (OGSI) Version 1.0." Global Grid Forum Drafts. http://www.ggf.org/ogsi-wg. April 2003

[7] G. Wasson, N. Beekwilder, M. Morgan, and M. Humphrey. OGSI.NET: OGSI-compliance on the .NET Framework. In Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid. April 19-22, 2004. Chicago, Illinois

[8] I. Foster and C, Kesselman. Globus: A metacomputing infrastructure toolkit. Int'l Supercomputer Applications, 11(2), p. 115-128, 1997.

[9] http://cloudcomputingapps.org/what-is-the-difference-betweencluster-computing-and-grid-computing.html

[10] http://www.linkedin.com/answers /technology/enterprisesoftware/TCH_ENT/552522-14941035

[11] https://loadfocus.com/blog/2013/08/01/what-is-latency In-load-and-performance-testing/

[12] D. Buszko, W. Lee, and A. Helal. "Decentralized Ad-Hoc Groupware API and Framework for Mobile Collaboration." Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work. 2001.

[13] J. Roth and C. Unger. "Using Handheld Devices in Synchronous Collaborative Scenarios." Personal and Ubiquitous Computing. Volume 5, Issue 4, December 2001

[14] M. Baker, R. Buyya, and D. Laforenza. "Grids and Grid technologies for wide-area distributed computing" Software - Practice and Experience. 2002.

[15] I. Foster, C. Kesselman, and S. Tuecke. "The Anatomy of the Grid - Enabling Scalable Virtual Organizations." International Supercomputer Applications. 2001.

[16] T. Sandholm, S. Tuecke, J Gawor, R. Seed, T. Maguire, J. Rofrano, S. Sylvester, and M. Williams. "Java OGSI Hosting Environment Design - A Portable Grid Service Container Framework." Global Grid Forum Drafts http://www.gridforum.org/Meetings/ggf7/drafts/OGSI%20Java%20Hosting%20Environment12.pdf. March 2003.

[17] T. Phan, L. Huang, and C. Dulan. "Integrating Mobile Wireless Devices into the Computational Grid." Mobicom 2002.

[18] B. Clarke, M. Humphrey. "Beyond the 'Device as Portal': Meeting the Requirements of Wireless and Mobile Devices in the Legion Grid Computing System." 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (associated with IPDPS 2002), Ft. Lauderdale, April 19, 2002.

Vishalan Gharat Will be receiving the B.E degree in Computer Engineering in June 2016, currently pursuing the same.His intrests revolve around Applications Development on Android Operating Systems and Robotics, and has successfully lead his team to be ranked in top five teams in National Robotics Competitions organized by IIT Bombay.He is a member of the Computer Society of India (CSI).

Aishwarya Chaudhari Will be receiving the B.E degree in Computer Engineering in June 2016, currently pursuing the same.Her intrests include Grid architectures, Ad-Hoc networks, and Distributed systems.She is a member of the Computer Society of India (CSI).

Jaspreet Gill Will be receiving the B.E degree in Computer Engineering in June 2016, currently pursuing the same.His intrests include Data sharing mechanisms, Ad-Hoc networks, Cluster computing and Computattional Analysis. He isa member of the Computer Society of India (CSI).

Shweta Tripathi has received the B.E degree from Jawaharlal Nehru Engineering College, Aurangabad, India, in the year 1999 and an Mtec in Computer Engineering with specialization in 'Network Infrastructure Management Studies' from Veermata Jijabai Institute of Technology (VJTI), Mumbai, in the year 2010. She is currently working as an assistant professor in Father Conciecao Rodrigues Insitute of Technology, Navi Mumbai, India. Her intrests are computer networks, computer security, Computer forensics and Project Management. She is a member of the Computer Society of India (CSI).