

PROTOTYPE APPLICATION IN ANDROID PLATFORM FOR SYSTEM ADMINISTRATION OF HPC CLUSTERS

S.Chakraborty¹MirazNabi Azad²Sourav Sen³Pritomrit Bora⁴Aditya
Singh⁵Bipal Das⁶ and Mohd.Tabeesh Noori⁷

¹Department of Computer Engineering, NIT Silchar, Assam

ABSTRACT

*As the number of **HPC** systems across the globe is booming at a rapid pace. The task of system administration is turning out to be tedious. Using a computer for every petty work can be cumbersome. In recent times there has been a paradigm shift from PC to mobile on a rapid scale. So a need has risen for such paradigm shift in system administration too. To solve this problem, an android application has been developed which allows the admin to monitor the system statistics remotely using his cell phone*

KEYWORDS

Android, Eclipse, Java, JSCH, SSH, IPMI.

1. INTRODUCTION

Android platform is the most promising and widely used operating system in recent times for smart phones and other hand held devices. So it is a good platform to develop a prototype application so that the system administrator of an HPC cluster can remotely login to the system and retrieve data regarding system health which will help him to take appropriate measures if needed without being physically present in vicinity of the system simply by using his or her cell phone.

2. BACKGROUND

In the following sections, an explanatory view of the System Architecture of the android platform is given, followed by a detailed account of the Application Development Environment of Android. The third section consists of a comparison between this application and other Related Works followed by a section on Research involved in moving ahead with this application, and finally, the Technologies that have been employed in developing it.

2.1 System Architecture

Android OS is a stack of software components that consist of five sections and four main layers.

- **Linux Kernel:**

The bottom Layer is the Linux Kernel which is based on Linux 2.6. This layer is responsible for basic services like memory management, process management, device management like camera, keypad, display etc. Also the kernel handles all the things that Linux is really good at such as networking and vast array of device drivers which takes the pain out of interfacing to peripheral hardware.

- **Android Runtime:**

This section provides a key component called Dalvik Virtual Machine which is similar to Java Virtual Machine but specially designed and optimized for Android. The Dalvik Virtual Machine and a set of core libraries that handle different functionalities found in core java libraries are found in this layer. Each android application runs its own Dalvik Virtual Machine instance. The Dalvik Virtual Machine makes use of Linux features like memory management and multithreading.

- **Libraries:**

On top of Linux kernel there is a set of libraries such as SQL Lite for storing and sharing of application data, SSL libraries for Internet security, libraries to record and play audio and video, SGR 2D graphics Engine, Open GL etc.

- **Application Framework:**

These are a set of Managers that provide services for views, activities, content providers, notifications and more. Native libraries that are responsible for performance optimization are accessed through the application framework. Some of the components are listed below:

- Content Providers-which manages access to central repository of data.
- Resource manager-that provides access to resources like graphics, layouts and more.
- Notification Manager-which is used for displaying custom alerts.
- Activity Manager-that handles the lifecycle of application and more.

- **Application:**

All the applications can be found in this layer. We make application to run on this layer only. Examples of applications are browsers, games email client app etc.

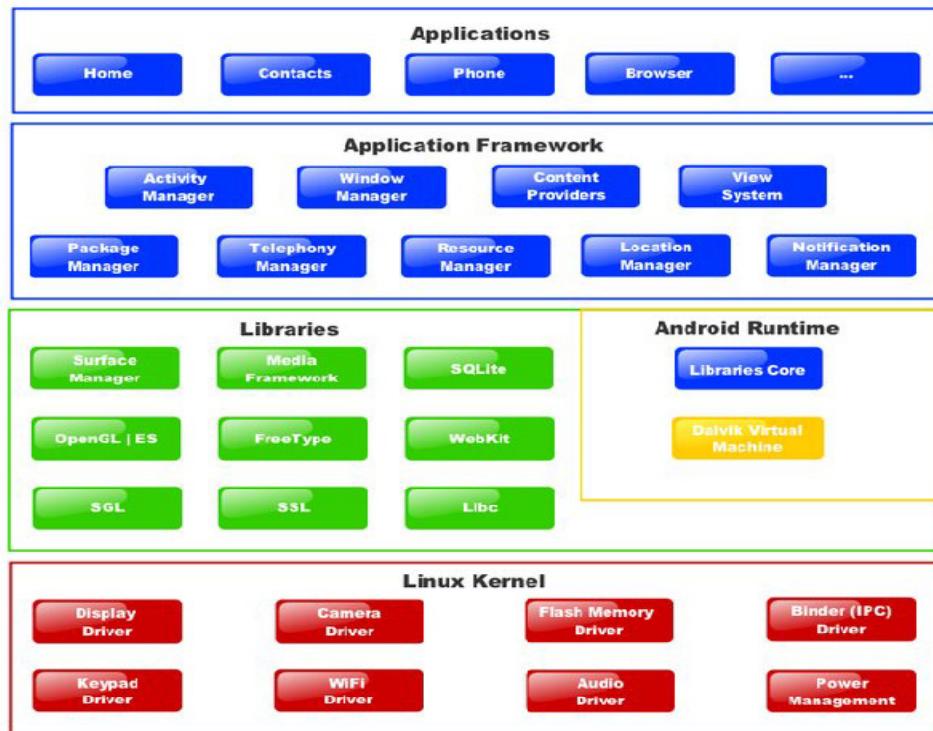


Fig 1. System architecture of Android

2.2 Android Application Development Environment

The Android Application Development Environment[3] is simple to comprehend. Android applications are written in Java and compiled into byte codes which will be converted to .dex file (Dalvik Executable File) using dx converter. This will further be compiled into Android package file (apk file) that can be installed to an Android device. Android application consists of one or more of these four components. The main building blocks are:

- **Activity:** It is the core component of Android Application. It is a user interface that dictates how the user interacts with the smart phone screen.
- **Services:** It is designed to keep running in the background to perform tasks that perform long running operations.
- **Content Providers:** The content providers are used as an interface to data. Content provider helps maintain shared data between applications.
- **Broadcast Receivers:** These are broadcast announcements made by the system.

2.3 Related Work

Application Name	Description	Features	Comparison
Connectbot	ConnectBot[9] is an very famous and well known Android App that is specially developed for Linux System Admins. It is a powerful client for the open-source Secure Shell (SSH) protocol that helps in managing simultaneous SSH sessions, create secure tunnels, and copy/paste between other applications.	<ul style="list-style-type: none"> • It supports login with a username and password to any arbitrary server on the local network or internet • Supports connections based on a public/private keypair instead of username/password for increased security • Allows frequently accessed hosts to be saved in a menu, so that they can quickly be re-connected to 	<ul style="list-style-type: none"> • ConnectBot supports both password based and key based authentication but this application supports password authentication only because that's more user friendly and people are more used to it. • ConnectBot provides list of frequently accessed hosts to allow quick connection. This application provides this through "remember me" checkbox. • This application supports problem reporting feature which allows developers to get reports on bugs and new features that are

		<ul style="list-style-type: none"> • Other programs on the android device can use ConnectBot as a ssh-agent so that the other program can pass data and commands securely to the server 	<p>required as user feedback which is not present in ConnectBot</p> <ul style="list-style-type: none"> • Additional command line access and graphical representation of state of the system is also provided in this app which is not present in ConnectBot.
Server Monitor	<p>Server Monitor[9] is an system Admin App that helps you to check the status of your network and web servers. And the best part about the App is that, Whenever the Server is Down, you will be first to get Notified. It is easily configurable and can check the Status a lot of times to</p>	<ul style="list-style-type: none"> • Provides status of network and web server. • User notification whenever the server is down. • Supports plain tcp connection as well as SSH tunnels. 	<ul style="list-style-type: none"> • Server Monitor supports both ssh and tcp but this application makes use of ssh tunnels only. • User notification facility is not available in this app since for that the user has to be logged in continuously which has some security issues and may compromise the security of the system.

	make sure that everything is going as expected,supports plain TCP connections as well as SSH tunnels.		
CuraSysAdmin	Cura[9] is the bundle of remote servers-administration tools that facilitate the easy maintenance, configuration and all-around administration of UNIX/Linux servers.	<ul style="list-style-type: none"> • A personalized Terminal emulator for direct interaction with said server. • Syslog module that allows for reading logs directly from the server. • SysMonitor module that visually graphs CPU and RAM usage percentages. 	<ul style="list-style-type: none"> • Personalized Terminal Emulator is not provided in this application but similar command line access is provided. • Graphical representation of the state of the system is provided just as in Cura Sys Admin.

2.4 Research in the Paper

The key research element in this paper is the introduction of a System Administration tool in mobile platform. For decades SSH has been used for the task of system monitoring through a PC or work station, but this paper shows us the possibility of doing same by using hand held devices or smart phones with the help of this application, since mobilization is the future. This paper also introduces tool for convenient System monitoring by providing visual aid

such as plotting graph from the extracted information in real time which makes it more perspicuous and helps in quick decision making.

2.5 Technologies Used

- **SSH AND Jsch:** SSH is a unix based interface and protocol for securely getting access to a remote computer. It is widely used by network administrator to control web and other kind of servers remotely. It provides strong authentication and secure communication over in secure channel by encrypting transmitted data during SSH session.
- **Jsch(Java Secure Channel):**Jsch is a java implementation of SSH. It is a Java library which provides the implementation of SSH functionality.
- **IPMI tool:** IPMI(Intelligent Platform Management Interface) is a an open standard hardware management subsystem to communicate .IPMI is a single command line interface useful for managing IPMI enabled devices .It enables user to manage system hardware ,monitor system health, monitor and manage the system environment independent of the operating system.

3. OBJECTIVE

- The application will act as a client and the administrator can connect to the SSH server by providing valid credentials.
- The administrator can use this application to analyze the statistics of the cluster by extracting information such as Memory usage, Hardware information, CPU and processing information, Disk space statistics and Network Packet statistics.
- This application also enables user to run **IPMI command** so that user can analyze information such as Power Status, Fan Sensors, Temperature Sensor and System Event Log.
- The application can also be used by the administrator to manage the processes running on the cluster in real time.
- The application can also be used for visual representation of the extracted cluster information.

4. SCOPE OF THE PROJECT

Many android applications with similar prototypes have been researched and developed but the existence of one single concrete application such as our application **HPC Health** is dubious. This application overcomes the problem of constant monitoring of the HPC cluster by enabling admin to access the system remotely and remain updated about the state of the system. As a new solution to develop such a system, this paper introduces the use of secure shell protocol using java.

5. EXPERIMENTAL DESIGN

- Firstly to connect the android device to the server[4], ssh connection was planned to be used to accomplish which jsch library was incorporated in the android environment.
- For communication between the client application and the server, input and output streams were created.
- The commands to be fired on the server were sent through the output stream and the results were channeled back to the device through the input stream.
- Graphs were plotted to check the performance of the HPC system using the data received. This was accomplished using the GraphViewapi for android. To give a real time view of the performance the graph data was refreshed every few milliseconds using threads.

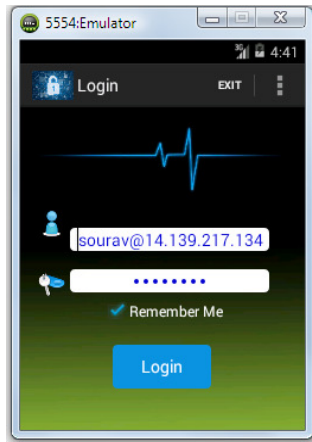


Fig 2. Login screen

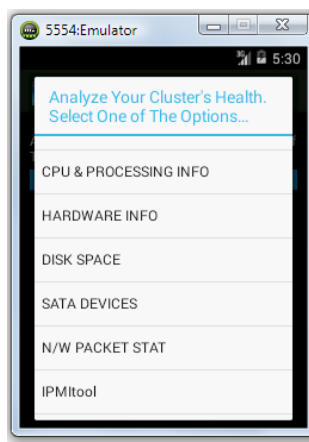


Fig 3. Menu screen

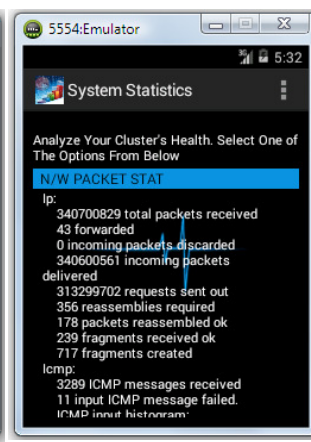


Fig 4. Output displayed

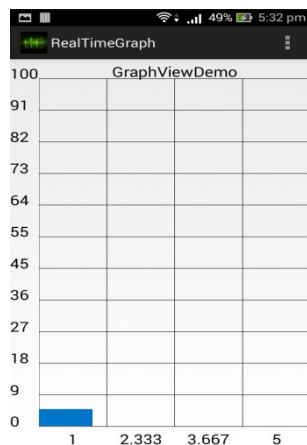


Fig5.percentageCPU usage graph

PID	USER	COMMAND
19686	root	lrm
16940	sourav	top
452	root	kipmi0
6180	root	java
1	root	init
2	root	kthreadd
3	root	migration/0
4	root	ksoftirqd/0
5	root	migration/0
6	root	watchdog/0
7	root	migration/1
8	root	migration/1
9	root	ksoftirqd/1
10	root	watchdog/1
11	root	migration/2
12	root	migration/2
13	root	ksoftirqd/2
14	root	watchdog/2
15	root	migration/3
16	root	migration/3
17	root	ksoftirqd/3
18	root	watchdog/3
19	root	migration/4
20	root	migration/4
21	root	ksoftirqd/4
22	root	watchdog/4

Fig 6. Process List

5.1 Use Case Diagram

HPC Health will be a project consisting of Client and Server. The interaction between user and the system is depicted in the use case diagram given below.

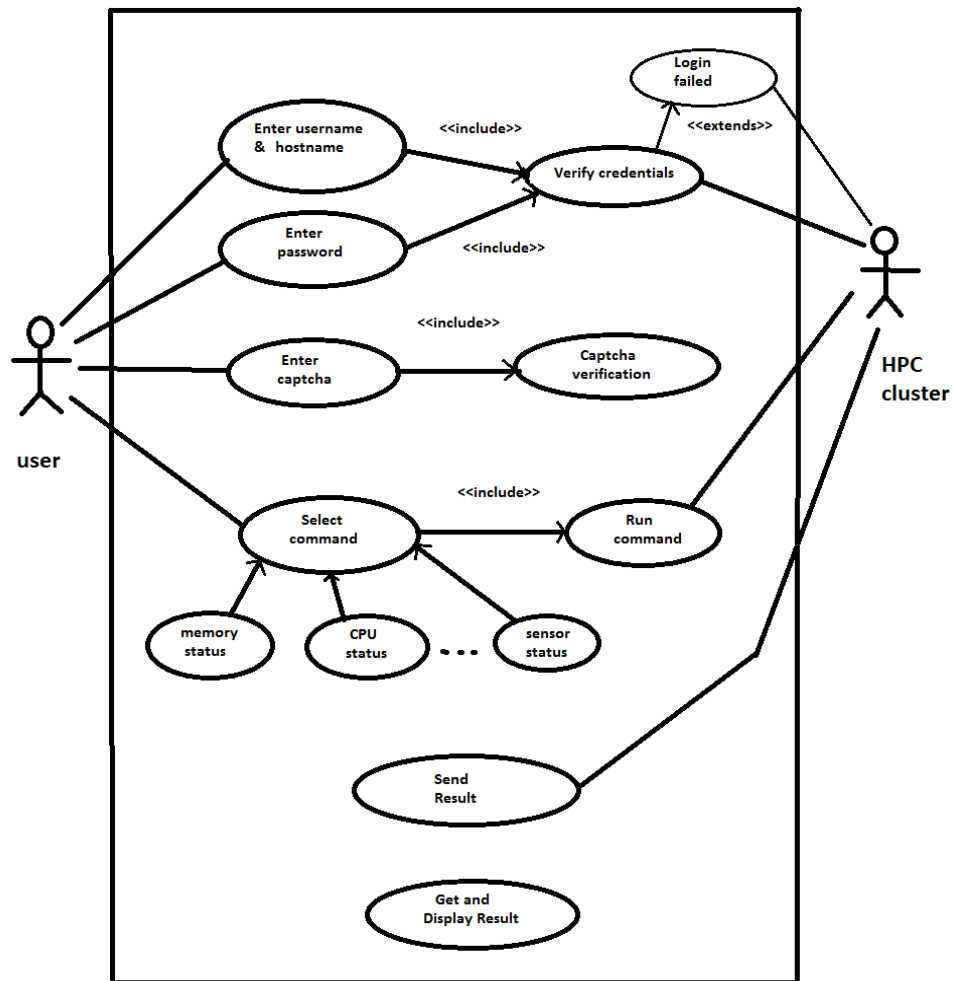


Fig 7. Use Case Diagram

5.2 Activity Diagram

The order of events that take place while manoeuvring the application by the user is depicted below with a activity diagram.

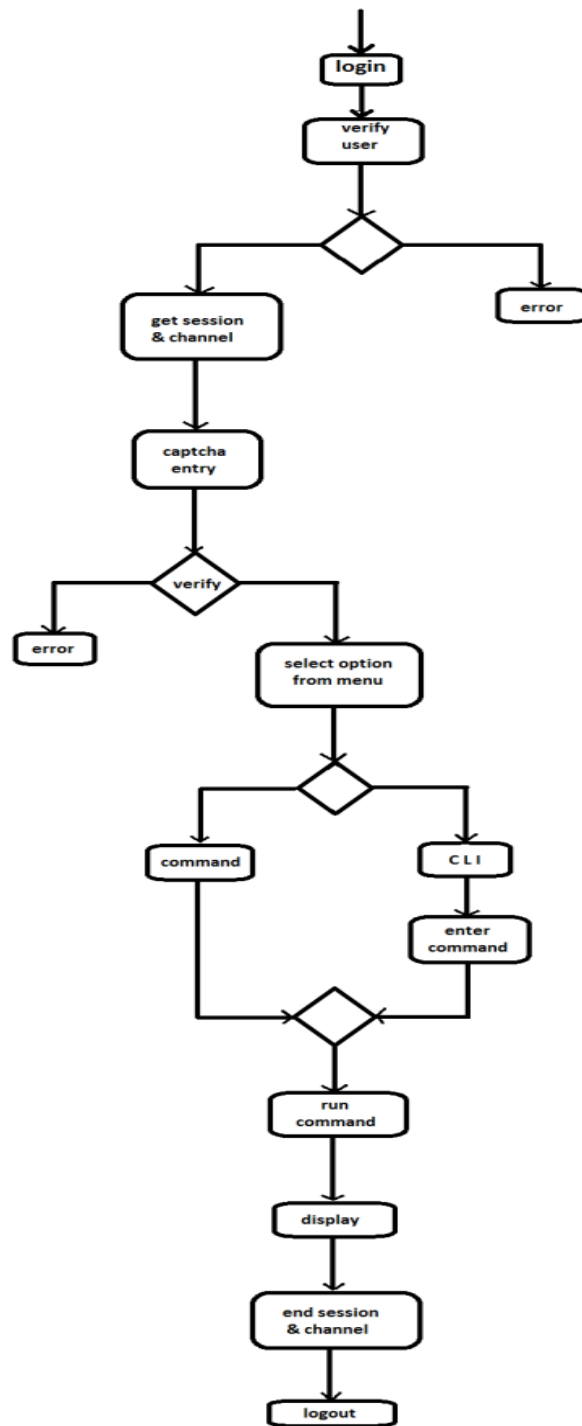


Fig 8.Activity Diagram.

6. IMPLEMENTATION

The implementation has been explained in the following sections. The first section is the establishment of SSH session and channel from the device to the server. Next section contains the Running of Commands through the connection established. The third section explains the implementation of graphs from the retrieved data. And the final section is about the generation of the Process list for the System Administrator.

6.1 Establishing the SSH Connection

The data extraction would require a session and a secure channel. The Java Secure Channel (JSch)[1] library was incorporated into the application for that. A JSch object was created and the `getSession()` function called with parameters username, hostname and port number (22 for SSH). The `setpassword()` method was called with the password as parameter. The connection was then established using `session.connect()` after verification of the user credentials. The channel was created of "shell" type rather than "exec" so that multiple commands could be sent and output could be retrieved one after the other. In "exec" the channel breaks after a single command is fired and output retrieved.

Session created before was used to create two channels:

- Outputstream channel is used to send the command to the remote server.
- Inputstream channel is used to receive the result from the remoteserver.

6.2 Firing Commands on the Server

Spinner element was used to provide the user a set of options in a drop down menu to choose from such as cpu and processing info, mem info, sensor info etc. Now, the next requirement was to implement the on click listeners for each of the menu items. On clicking each menu item, the item is stored in a string and using an if-else condition, the item clicked was checked and the appropriate command was sent through the Outputstream associated with the channel and then the result was read from the Inputstream.

A small amount of delay was necessary between writing the command and starting to read the input stream so as to allow the command to run in the server first before starting to read its result.

6.3 Graphical representation of cluster stats

Graphs were used to provide the user a graphical view of the system stats and as a visual aid in understanding the current state of the cluster. With this vision `GraphViewapi`[2] was used which is useful in providing well-structured graphs in android applications just as a task manager does. Graph View is a library for android to programmatically create flexible and nice looking graphs. It is easy to understand, to integrate and to customize it. The stats were extracted by firing commands and then were used to plot the graph. An abstract data type was created with two double type attributes to hold the x and y co-ordinates to be plotted. Stepwise approach to developing this feature is as follows:

- Run the appropriate command and retrieve the data to be plotted.

- The data is stored in series of data of abstract type set the type of the graph.
- Add the series to the graph for display set the specifications of the graph such as maximum y bound, maximum x bound, viewport etc.
- Add the graph to the layout.
- Keep updating the graph periodically using background threads and timers.

6.3 Managing processes running in the cluster

The system administrator invariably has to control and manage the processes running on the system from time to time. For the task of managing the processes running, it is essential for the administrator to be able to get a complete list of the processes running on the system and then modify them to his/her needs such as killing process, spawning new process, changing their permissions and priorities, etc. For this, the top command was used but output of top being dynamically changing it was difficult to manipulate its output. So we used a two-step mechanism whereby we first run and write the output of top onto a text file and then read and modify the content of this text file.

The steps used are as follows:

1. Whenever the user requests the list of processes internally the top command is run and the output is written onto a text file. These files content are then manipulated using stringmanipulation techniques to extract the name, process id and the user to which the process belong.
2. A table is then constructed in the current activity and the stings extracted in previous steps are used to fill the cells of the table. This is done using the Table Layout in the Android Development Tools.
3. A listener is then used for detecting clicks on the rows of the table. On clicking any row of the table a new activity is created and the process id and name of the corresponding process are passed via Intent to the new Activity where it is used for managing the process.
4. The new Activity possesses multiple options for the user to choose from such as changing process priority, killing the process etc. This provides a handy tool for viewing all the processes in the system and to choose to control any of them. Doing this is essential to administer the cluster.

7. CONCLUSION

This thesis project has covered mobile application development in Android platform. The basic components of android have been described along with the overview of the Android architecture. The outcome of this project has accomplished most of the goals we set at the start. This project was an opportunity to discover and explore new skills in the field of mobile application development. The final application can be useful to system administrators for keeping track of their systems (Cluster) remotely without disrupting the normal performance of the system.

ACKNOWLEDGEMENTS

We would like to extend our gratitude to Dr P.K. Sinha(Senior Director, HPC, CDAC), Mr Abhishek Das and the C-DAC members at NIT Silchar for their cordial support, valuable information and guidance provided by them, which helped us in moving ahead with this task through various stages. We are also very grateful for their cooperation during the period of our assignment.

REFERENCES

- [1] "Jcraft". Web.<http://www.jcraft.com>. January 10, 2015.
- [2] "Graph View". Web. <http://www.android-graphview.org>. January 16, 2015
- [3] RamziN.Sansour, NidalKafri, MuathN.Sabha "A survey on mobile multimedia application development frameworks". Multimedia Computing and Systems (ICMCS), 2014 International Conference on 14-16 April 2014, pp 967 – 972.
- [4] Ms Loveena Lionel "Android As A Server Platform". Proceedings of National Conference on New Horizons in IT - NCNHIT 2013, pp 135-137
- [5] DeepaliJavale "Home Automation and Security System Using Android ADK", International Journal of Electronics Communication and Computer Technology (IJECCCT) Volume 3 Issue 2 (March 2013).
- [6] Andreas Sommer, Stephen Krusche "Evaluation of cross platform frameworks for mobileapplications". 1st European Workshop on Mobile Engineering (ME13). Aachen - Germany, March 2013
- [7] Henning Heitkotter "Evaluating Cross Platform Development Approaches for Mobile Applications". Web Information Systems and Technologies Lecture Notes in Business Information Processing Volume 140, 2013, pp 120-138.
- [8] Ifeanyi P. Egwutuoha, David Levy,BranSelicShipping Chen "A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems". The Journal of Supercomputing. September 2013, Volume 65, Issue 3, pp 1302-1326.
- [9] "Linuxaria". Web.<http://www.linuxaria.com>. November 1, 2014.