

Implementation of Hadoop Pseudo-distributed Cluster on Android using 'chroot'

Prof. S. Y. Raut¹, Balwant B. Raut, Mangesh S. Jondhale, Nilesh R. Jaware⁴, Prachil S. Tambe⁵

¹Department of Computer Engineering
Pravara Rural Engineering College, Loni
Shrirampur, India
getdiya2008@gmail.com

²Department of Computer Engineering
Pravara Rural Engineering College, Loni
Kolhar, India
balwant.raut383838@gmail.com

³Department of Computer Engineering
Pravara Rural Engineering College, Loni
Nimgaonjali, India

⁴Department of Computer Engineering
Pravara Rural Engineering College, Loni
Mamdapur, India
nrjaware@gmail.com

⁵Department of Computer Engineering
Pravara Rural Engineering College, Loni
Dadh, India
prachil.tambe@hotmail.com

Abstract: *In this paper, we are discussing porting methods of Linux console based applications to android platform and primarily concentrating on migration of Hadoop from personal computer to android device. There are four different approaches towards porting methodology [1] viz. recompilation of Linux binaries using bionic c library, running android under Linux, running Linux in android environment, and Virtualization method of multiple virtualized Linux running parallel. We are focusing on implementation of Linux with help of 'chroot' method. This paper specifically discuss the 'Hadoop' migration to an Android device. Hadoop being important factor of big data processing, provides a flexible Map-Reduce Implementation. With this port, it would be easier to minimize cost of cluster formation. We are also discussing pseudo-distributed node formation on this ported Hadoop.*

Keywords: mobile Hadoop, Hadoop on android, chroot, linux on android
INTRODUCTION

Hadoop Is Combination Of Map-Reduce, Hdfs, Hadoop Common And Yarn. Hadoop Common Is A Set Of Utilities That Provides Support For Other Hadoop Modules. Hadoop Distributed File System Or Hdfs, Is A Distributed File System That Provides High-Throughput Access To Application Data. Yarn Is A Job Scheduler And Cluster Resource Manager. Mapreduce Is An Algorithm For Parallel Processing Of Large Data Sets.

Transfer same shared library files and recompiled binaries to Android Application.

- Build an XML framework to access Hadoop installation and perform various tasks.

- Build an application with above XML framework, recompiled binary files as well as recompiled Hadoop libraries.

PROBLEMS:

- Bionic libc have restricted POSIX compatibility.[1]
- Bionic libc do not have C++ exceptions and locales or wide char support.
- Recompiling Hadoop libraries will require Linux distribution on linux.

Due to these limitations, application porting with this method can be very difficult.

1.1 Android under Linux

This approach was first introduced by Pragmatux [2], by creating Pragmatux OS for embedded system. It utilizes tools, concepts from the Debian Linux, It is not complete Debian Linux, but look and feel are the same as Debian. Since it was the first embedded OS to do so, we will be discussing same.

Pragmatux manages the entire file system, package management, repository management, cross-toolchains etc. It minimizes the package size and keeps Linux applications compatible with ARM/Snapdragon processors. It utilizes android framework to visualize the contents.

Linux processes	Activities
Pragmatux OS	Android framework
Init	
Linux kernel	
Bootloader	

Fig 1: Pragmatux OS Framework

Both OS run side by side, but Linux keeps control and keeps full support for all packages. Android provides framework and other services like Wi-Fi, Bluetooth etc. Since native Linux is present, no program conversion is necessary. All the services like Wi-Fi, Bluetooth etc. can be used by both OS.

Problems:

- Need to use specific OS rather than AOSP android distribution.
- OS is suitable for embedded platform rather than mobile platform.
- Cannot be generalized as easy as android.
- This approach is only implemented for embedded platform, not for mobile devices.

With this approach, there is a possibility to create a Linux distribution for android device. One of such approach is Ubuntu OS for mobile devices being developed by canonical.

1.2 Virtualization

According to the strict computing definition of virtualization means “does not exist physically but rather simulated by software.” This is the technique which reduces hardware cost to run different operating systems with same hardware. It also provides security in terms of access limitations outside the defined folder. For android, there are multiple virtualization methods. Popular amongst them are Cells [3], LXC on Android, VMware Horizon Mobile [4], Samsung Knox [5], KVM/ARM [6] etc.

There are two approaches towards virtualization on embedded platform

1.2.1 Hypervisor

Hypervisor [7] is virtual machine manager on Android operating systems. Below fig shows types of Hypervisor based on location.

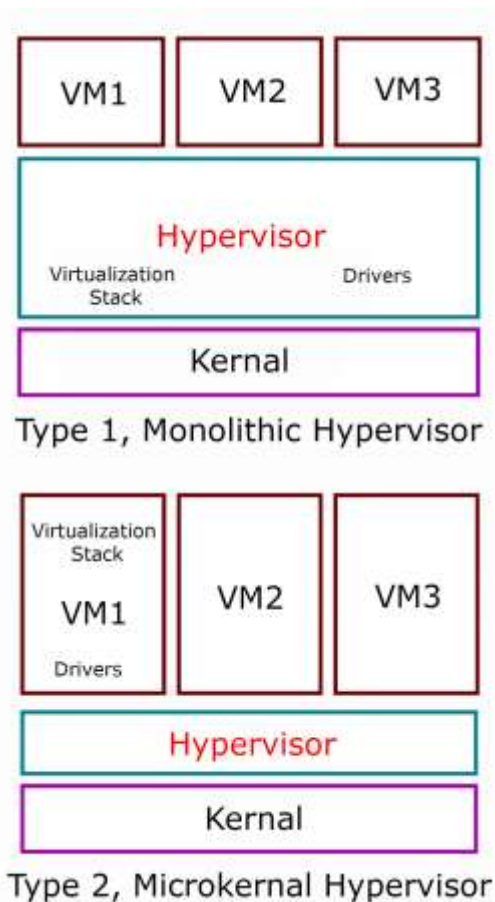


Figure 1: Types of Hypervisors

KVM/ARM is kernel based virtual machine manager, provides full virtualization solution for Linux based operating systems, can also be called as Linux ARM Hypervisor. KVM/ARM is open source, acts as hypervisor. It utilizes existing Linux kernel functionality thus comparative to other hypervisors, provides small lightweight code-base. It has been demonstrated under Arndale board with Exynos 5250 SoC.

BYOD [8] policy implementation introduced a new concept in virtualization where employees were allowed to use their own technology or devices for work purpose, led in development of separate virtual working environment. Here type 2 hypervisor is used, where two different OS runs parallel to each other.

VMware Horizon Mobile Secure Workspace is same virtualization environment which uses hypervisor with lightweight paravirtualization technique for ARMv7 processor. VMware has created their own workspace removing android interface. It also have facility to control your desktop and connect with your work device with the help of VMware Horizon 6[9].

KNOX EMM is mobile client app provided by Samsung. It uses Samsung KNOX Mobile workspace – a web based enterprise solution. Samsung KNOX hypervisor uses TIMA - TrustZone-based Integrity Monitoring as hardware security provided with ARM. For software, SE for Android.

VMware and Samsung uses Type 2 hypervisor with their products. Other[9] Hypervisor includes OKL4 Microvisor, is a microkernel-based embedded hypervisor (Type 1) used in Motorola Evoke AQ4, Red Bend's VLX Virtualisor (Hypervisor Type 1), Xen Hypervisor on ARM (Type 1) etc.

1.2.2 Container

'Cells' is container based virtual environment which enables user to run an android operating system side by side with multiple virtual android operating systems. Here android operating systems are may not be of same version.

LXC is a Linux container, which can be ported to Android. It has a library of container tools which can be ported to android.

1.3 Linux in Android - 'chroot' method

1.3.1 What is 'chroot'?

'Chroot' changes root environment. It mounts a specific directory as root and create a sub-process tree for particular system. Processes running under this environment, typically can't access the directories outside the 'chroot'ed directory. This behavior of the newly created root directory is known as 'chroot jail' [10]. We can see such implementation with DebianRunner [11] also.

Following are the requirements needs to be fulfilled, in order to implement this method.

Requirements

- Root Privileges.
- ARM supported Linux distribution
- Kernel modules loaded.
- Swap enabled and Looping enabled device
- Internet connection forwarding.
- High Speed Internet connection to download and install additional components.

Generally chroot is used as testing and development environment, to test dependency control, to test the capability of software for particular device, recover system etc.

1.3.2 Steps to implement chroot

There are two ways to implement chroot on android device

- Manual – using terminal emulator application and busybox from Google play store, and performing normal chroot on android. One can perform it on Linux by mounting the new Linux system directory or image.
- Automatic – using an app called Linux deploy[12] from Google play store, which has configuration setting for different variants of Linux and custom boot script support.

After installing compatible Linux for your device(x86/armv5/armv71), install VNCviewer app for remote login interface to same Linux. VNC viewer typically uses 5900 port number.

2. Implementation

All current implementation of Linux are available for various platforms. However Hadoops performance on various platform varies with processor architecture [13], [14]. Comparing with different architectures, and Hadoop's basic need of commodity hardware, low power consuming arm devices should be ideal for Hadoop implementation.

2.1 Setting up environment

To install Hadoop in such 'chroot'-ed environment we need to perform the following steps.

- Update Linux distribution
- Install java for your device architecture/abi
- Download latest Hadoop from apache repository
- Navigate to home folder of user, then use 'vi .bashrc' to add following lines and add environmental variables. Use your installation path of java and Hadoop in the following code

```
export JAVA_HOME=/usr/lib/jvm/java
```

```
export HADOOP_HOME=/usr/local/hadoop
```

```
export PATH=$PATH:$JAVA_HOME/bin
```

```
export PATH=$PATH:$HADOOP_HOME/bin
```

```
export PATH=$PATH:$HADOOP_HOME/sbin
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_HOME=$HADOOP_HOME
```

```
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

```
export YARN_HOME=$HADOOP_HOME
```

```
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

```
export HADOOP_OPTS="-
java.library.path=$HADOOP_HOME/lib"
```

- Edit `hdfs-site.xml`, to configure replication, Namenode and datanode. Edit `yarn-site.xml`, to configure nodemanager and nodemanager_handler. Edit `mapred-site.xml`, to configure `mapreduce_framework`
- Run `java -version` to check if java is properly configured
- Run `hadoop version` to check if Hadoop is properly configured
- Install `ssh, rsync`
- Run following commands to set up password-less ssh authentication for current user.

```
$ ssh-keygen -t dsa -P "" -f ~/.ssh/id_dsa
```

```
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

- Run `ssh localhost` to check password-less login

After this implementation Hadoop will runs smoother. We can easily implement and run any mapreduce program on this Linux and Hadoop installation. It also supports execution of any Linux platform program.

3. Conclusion

We have successfully implemented Hadoop on android using 'chroot' method. However we are expanding our research towards expanding this method to greater extent as the number of mobility devices are growing. Since ARM is power saving processor family and Android provides portability and flexibility, and with Hadoop's power we can bring great productivity platform on android. We are also researching towards bringing Hadoop outside the 'chroot jail'.

4. Applications

BYOD [8] policy implementation introduces mobile devices in our daily industrial life. While we busy with our work, these devices are not being used at all. Utilizing that computing power towards productivity is more affordable for small type industries. Small scale businesses wishing to run Hadoop jobs could benefit from this. Educational institutions could also implement this methodology to study and understand the industrial implementation of Hadoop in economical and energy efficient way.

One such application is also stated as to develop, debug and test MapReduce algorithms. Any small or large industry individual/programmer can develop, debug and test his or her MapReduce program or job on android device.

5. Acknowledgments

Our thanks to Mr. Mike Anderson of The PTR Group, Inc. for providing every kind of help during this research. We also would like to thank Mr. AntonSkshidlevsky, for allowing us to use his application.

References

- [1] Anderson M., (2014). Using Chroot to Bring Linux Applications to Android, Embedded Linux Conference + Android Builders Summit April 29th – May 1st 2014, The Linux Foundation.
- [2] Kuester R., Developing on DragonBoard: Getting Started with APQ8060A and Pragmatux+Android, LinuxCon Europe 2012.
- [3] Andrus, J., Dall, C., Hof, A. V. T., Laadan, O., & Nieh, J. (2011, October). Cells: a virtual mobile smartphone architecture. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (pp. 173-187). ACM.
- [4] Bowker M. (2013). The VMware Mobile Secure Workplace, Whitepaper, ESG Inc.
- [5] Samsung, (2014). Meet evolving enterprise mobility challenges with Samsung KNOX, White Paper, Samsung Inc.
- [6] Dall, C., & Nieh, J. (2014, February). Kvm/arm: The design and implementation of the linux arm hypervisor. In Proceedings of the 19th international conference on Architectural support for programming languages and operating systems (pp. 333-348). ACM.
- [7] Chawla, S., Nigam, A., Doke, P., & Kimbahune, S. (2011). A survey of virtualization on mobiles. In Advances in Computing and Communications (pp. 430-441). Springer Berlin Heidelberg.
- [8] Jindal, A. K., & Naik, V. (2013). Protecting android devices following BYOD policy against data security and privacy attacks.
- [9] Jaramillo, D., Furht, B., & Agarwal, A. (2014). Virtualization techniques for mobile systems. Springer International Pu.
- [10] McFearin, L. D. (2011). Chroot Jail. In Encyclopedia of Cryptography and Security (pp. 206-207). Springer US.
- [11] Gupta, A., Preston, K., & Rodriguez, A. (2010). DebianRunner: Running Desktop Applications on Android Smartphones.
- [12] Skshidlevsky, A. A. (2013). The Software to Automate Processes of Installation and Launch of GNU/Linux Applications on Android Devices. Information & Control Systems/Informazionno-Upravlyaushie Sistemy, 66(5).
- [13] Ferreira, T., Espinosa, A., Moure, J. C., & Hernández, P. (2013). An Optimization for MapReduce Frameworks in Multi-core Architectures. Procedia Computer Science, 18, 2587-2590.
- [14] Yan, F., Cherkasova, L., Zhang, Z., & Smirni, E. (2014, May). Heterogeneous cores for mapreduce processing: Opportunity or challenge?. In Network Operations and Management Symposium (NOMS), 2014 IEEE (pp. 1-4). IEEE.