Customized Virtual Machines for Software Provisioning in Scientific Clouds

Wei Chen, Junwei Cao, and Ziyang Li
Research Institute of Information Technology
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University, Beijing 10084, P. R. China
e-mail: jcao@tsinghua.edu.cn

Abstract—**Scientific applications require special data computing and analysis software, which may not mature enough at a production level. Software provisioning, especially in a cloud computing environment, bring new challenges. In this work, software package management systems are summarized. A customized software provisioning service is developed using virtualization technology. According to users' requirements, virtual machines with required software deployed ready-for-use can be obtained in a very straightforward way. This work is being applied in the LIGO (Laser Interferometer Gravitational-wave Observatory) Scientific Collaboration for gravitational-wave data analysis. With software virtual machines available, scientific applications can benefit from cloud computing technology and resouces.**

Keywords-cloud computing; scientific applications; virtual machines; software provisioning

I. Introduction

Modern scientific applications, e.g. high energy physics and astrophysics, require specially designed software packages for massive data computing and analysis. These software are developed by research scientists instead of private companies, so can be only supported and maintained by the research community. The main problem is that other research scientists except the software developers may find it very difficult to get these software deployed.

For example, LIGO (Laser Interferometer Gravitational-wave Observatory) [1] is an astrophysical experiment aiming at direct detection of gravitational waves predicted by Einstein's General Theory of Relativity. There are two LIGO observatories, one at Hanford, Washington State and the other at Livingston, Louisiana State. LIGO Scientific Collaboration (LSC) [2] is a world-wide international academic organization, including over 700 research scientists from over 60 research institutes all over the world. LSC members share LIGO data and carry out data analysis in a close collaborative way. LIGO produces terabytes of experimental data per day and LIGO data analysis require large amount of CPU cycles and data storage [3].

The LIGO project is funded by the National Science Foundation, starting from 1990's. Before that many researchers from Caltech and MIT have been working on the design since 1960's. Over several decades' development, there are many software systems available for LIGO data analysis, including LDAS, LAL, LIGOtools, LSCsoft, LDG, DMT, etc. [4]. These software are developed in different languages, e.g. Python, C++ and Matlab, have dependency with external software packages, e.g. Condor [5], Globus [6], and ROOT [7], are maintained by different organizations. Astrophysicists have to rely on these software packages for LIGO data analysis [8]. Unfortunately it is very difficult for newbie to handle the complexity of software installation and deployment.

There are many existing software package management and distribution methods and tools. In this work, virtualization technology is utilized for software deployment and distribution. Virtual machines are self-contained environments that software can be pre-installed in before distribution. This provides a neat solution especially for users without many computer skills to handle complex software usage. Also the emerging cloud computing paradigm is also mainly enabled by virtualization technology. Virtual machines make it very straightforward for users to benefit from resources and services provided by a cloud environment. In this work, we use LSC software package management as a case study and we develop LIGO software virtual machines. A web-based user interface is also implemented and an online virtual machine generation system is available so that users can place customized order for software virtual machines for downloading.

The rest of the paper is organized as follows: LIGO software packages and existing distribution methods are summarized in Section 2; software virtual machines are introduced in Section 3 and web-based customization system is presented in Section 4; the paper concludes in Section 5.

II. Software Package Management

In this section, a brief summarization of LIGO software packages and existing distribution methods is given.

A. LIGO Software Packages

- LDAS [9]

LDAS (LIGO Data Analysis System) is a software package for online achieving system of real-time LIGO data. LIGO data are stored in binary files, called frame files, in a special Gravitational Wave Format (gwf). The package also includes software tools for follow-up data analysis, using message passing interfaces (MPI) and LIGO algorithm library (LAL). These are developed in C/C++ and there are also application interfaces available for extended implementation. The LDAS diagram is illustrated in Fig. 1.

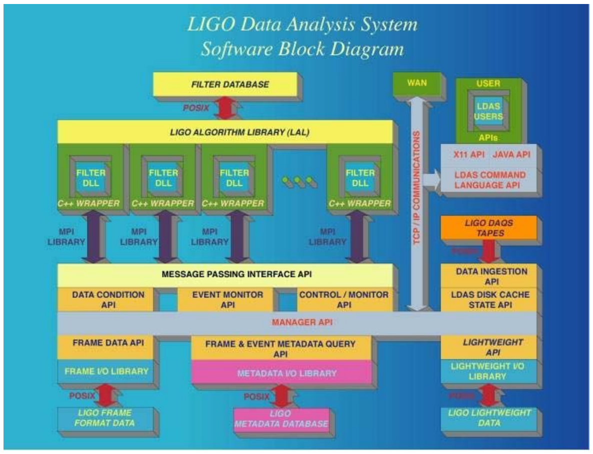


Figure 1. LIGO data analysis system diagram.

- LIGOtools [10]

LIGOtools is another toolkit with a self-contained lightweight distribution mechanism. LIGOtools include several C libraries, Matlab codes and Tcl scripts, which are organized in an uniform way. LIGOtools provides a lightweight runtime environment with bins and libs. It also provides a *ligotools_update* tool for getting the latest software package. This is similar to *yum* described later. Software tools included in LIGOtools are listed in Table I.

TABLE I. LIGOTOOLS SOFTWARE PACKAGES

Package	Description
Fr	C library and utilities in core Virgo distribution to read/write data in frame format
FrContrib	Additional utilities for working with frame files, exclusive of core Virgo distribution
dataflow	Raw data and metadata access utilities
detgeom	Matlab routines to define and manipulate detector geometry
guild	Graphical User Interface to LIGO Databases
httptools	Simple utilities to retrieve files via http
ilwdread	Matlab script to read an ilwd file
ldasjob	High-level interface for running LDAS jobs from Tcl scripts
medmguide	Graphical user interface to examine EPICS medm (*.adl) files
metaio	C library and utilities to read and manipulate LIGO_LW table files
runtools	Summarize status of interferometers during science/engineering runs
segments	Generate and manipulate lists of GPS time intervals

- LSCsoft [11]

LSCsoft is a software repository maintained by the Data Analysis Software Working Group (DASWG) of the LIGO Scientific Collaboration (LSC). Since LSC takes CentOS as the de facto OS, LSCsoft manages software packages in rpms and distributes software packages via yum. There methods are described later. Here a list of software packages included in the LSCsoft repository is provided in Table II.

TABLE II. LSCSOFT SOFTWARE PACKAGES

Package	Description
FrameL	for data_frame manipulation
MetaIO	for LIGO_LW files metadata manipulation
LAL	LIGO Algorithm Library
LALAPPS	LAL based Applications
GLUE	Grid LSC User Environment
FrameCPP	C++ interface to access frame structures
DOL	Data Monitoring Tool (DMT) Offline
GDS	LIGO Global Diagnostics System

For example, Data Monitoring Tool (DMT) [12] is a toolkit for online monitoring of LIGO data quality. It can be also used in a offline way as a Linux-based data analysis tools with support of remote user interfaces, as illustrated in Fig. 2.

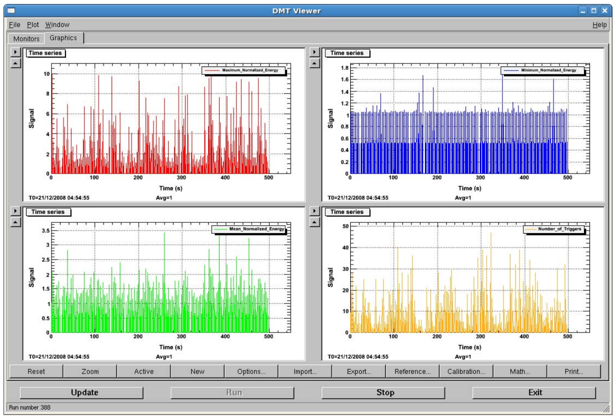


Figure 2. The DMT viewer

- LSC Data Grid [13]

LSC Data Grid (LDG) is a software package for LSC members to access grid resources, including over 10 computer clusters all cross the US and Europe. Major grid software packages, e.g. Condor and Globus, are included in LDG. Also there are utilities for user certificate application, renew and update. With certificates, users can access grid resources in a uniform way. LDG is also organized and maintained in the LSCsoft repository.

B. *Software Package Distribution*

- Tarballs

Tarballs are very common in Linux-based software packages, similar to zips in Windows and sits in Mac. Tarballs can be untarred and installed on Linux. But in general, software dependency cannot be handled in an automatic way. New users may become frustrated if there is any problem during the installation process. But for professional users, tarballs are still a straightforward way for software distribution, since more flexibility can be provided if required.

- Rpms and Debs

Redhad Package Manager (Rpm) is widely used in Linux and maintained by open source community. Other Linux providers, e.g. SuSE, also utilize rpms. Similarly, Debian uses Debs. There are several types of rpms, e.g. binary, source and delta. For each rpm, there is a XML descriptor, including all the package related information. Especially, software dependency and version management can be handled in an automatic way when applied together with yum or apt. These provide a neat solution for software management and distribution.

- Yum and Apt

In order for users to find available software package quickly, software repository is becoming very popular nowadays. Different versions of different software packages are organized in a systematic way. Yum works with rpms and Apt-get works with debs. Software repositories work online that can provide software downloading services according to users' requirements. These are currently mainstream software management tools for Linux-based systems, very close to the idea of Software as a Service (SaaS).

In this work, we are proposing an additional solution for software management, software virtual machines. As described above, LSC has many software packages maintained and distributed in different ways. Virtual machines can provide a self-contained way for handling all these complexity and deliver to end users with an all-in-one environment, which is quite essential in future cloud computing environments.

III. Software Virtual Machine

A. Scientific Clouds

Cloud computing is considered as the latest wave of computing infrastructuralization [14], after cluster computing, utility computing, grid computing and services computing. Clouds provide remote resources as services at different levels, IaaS, PaaS and SaaS. Cloud computing will have a bright future since supported strongly by industry [15].

Scientific applications can benefit a lot from cloud computing. Most scientific applications require large-scale CPU and storage resources on-demandingly. In a cloud environment, many applications can share one large resource pool so that elastic computing can be achieved, where Virtualization plays a key role.

B. Virtualization Technology

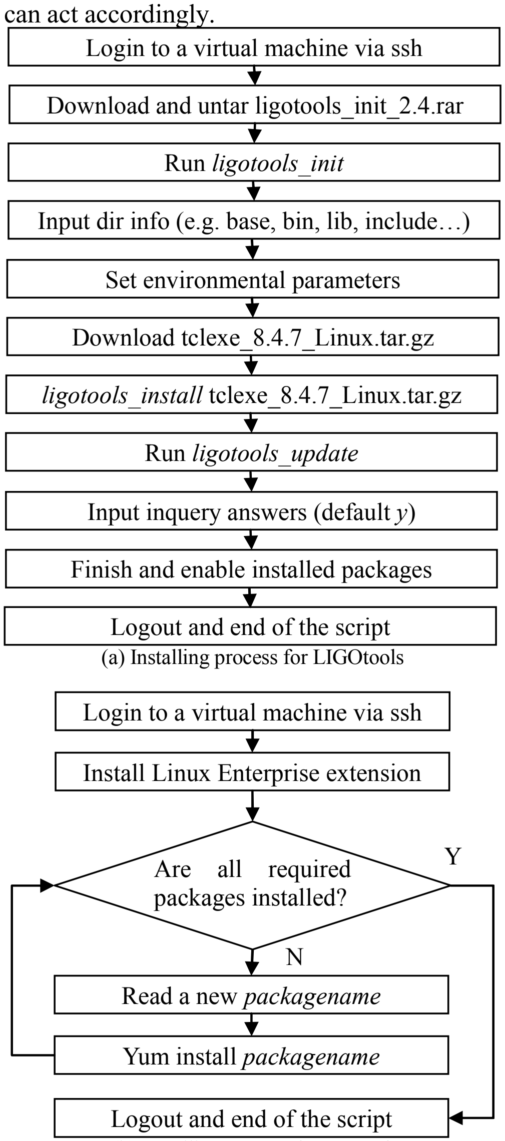
Virtualization at the hardware layer can enable find-grained resource sharing of a computer. Virtual machines are isolated environments that can be customized to meet specific application requirements. Different virtual machines can use different OS on one physical machine. Since virtual machines utilize hardware resources indirectly, there will be unavoidable overhead when virtualization technology is applied.

Virtualbox [16] is a free implementation of virtual machines provided by Oracle. Vitualbox has an internal modular structure with C/S supports. Configurations of virtual machines are written in XML and can be shared among multiple machines. Other features include supporting share folders between hosts and virtual machines. Also Virtualbox provides SDK for application development.

C. Automated Software Installation

In order to install software packages in existing virtual machines, a software tool, Expect [17], is utilized to automate the process. Expect is a tool for automating interactive applications such as telnet, ftp, passwd, etc. Expect is also useful for testing these same applications. And by adding Tk, users can wrap interactive applications in X11 GUIs.

Previously installation of software in virtual machines is an interactive process. With Expect, this process can be scripted and enabled automatically. For example, Fig, 3 shows the automated process for installing LIGOtools and LSCsoft to virtual machines using Expect. For LSCsoft, users can specify which packages are required. These information are stored in files so that corresponding Expect scripts can act accordingly.


Login to a virtual machine via ssh
Download and untar ligotools_init_2.4.rar
Run *ligotools_init*
Input dir info (e.g. base, bin, lib, include…)
Set environmental parameters
Download tclexe_8.4.7_Linux.tar.gz
ligotools_install tclexe_8.4.7_Linux.tar.gz
Run *ligotools_update*
Input inquery answers (default *y*)
Finish and enable installed packages
Logout and end of the script
Login to a virtual machine via ssh
Install Linux Enterprise extension
Are all required packages installed?
Y
N
Read a new *packagename*
Yum install *packagename*
Logout and end of the script

(a) Installing process for LIGOtools

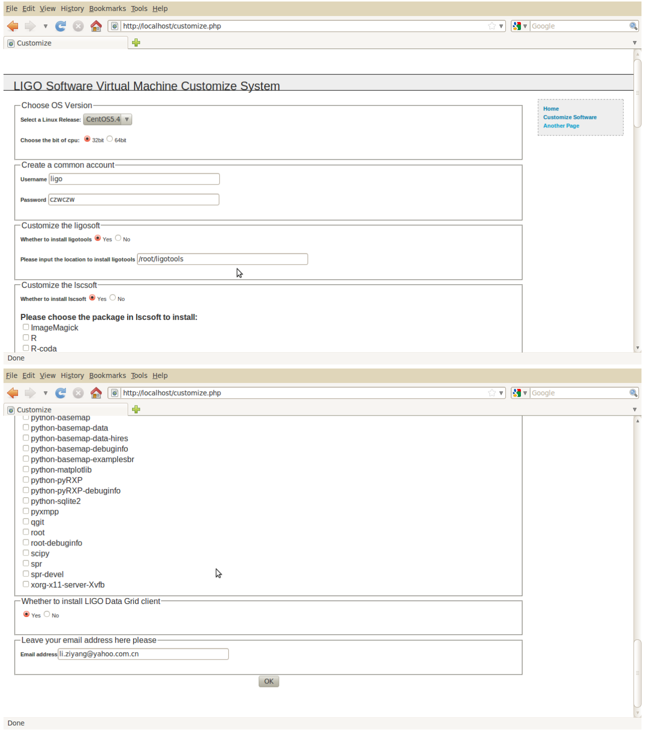
(b) Installing process for LSCsoft

Figure 3. Automated software installation to virutal machines

IV. CUSTIMIZED SYSTEM IMPLEMENTATION

The system implementation includes several components: web-based user interfaces, task management, a background daemon for task execution.

As illustrated in Fig. 4, users can specify their requirements via web-based user interfaces, where they input the Linux version, required software packages, and contact information. These information is maintained in a task queue. Task management is responsible to handle different requests from different users. A background daemon is also running on the server. It scans the queue and put requests into an execution mode. The daemon will call Expect scripts to start a virtual machine, install required software packages, retrieve corresponding Virtualbox image files, and email the downloading link to users. Users can download and install Virtualbox locally and run the virtual machine using the downloaded image files. Users then have a customized LIGO data analysis environment locally available without bothering about details on software installation.


File Edit View History Bookmarks Tools Help
http://localhost/customize.php
Google
Customize
Home
Customize Software
Another Page
OK
Done

LIGO Software Virtual Machine Customize System

Choose OS Version

Select a Linux Release: CentOS5.4

Choose the bit of cpu: ● 32bit ○ 64bit

Create a common account

Username: ligo

Password: CZWCZW

Customize the ligosoft

Whether to install ligotools ● Yes ○ No

Please input the location to install ligotools: /root/ligotools

Customize the lscsoft

Whether to install lscsoft ● Yes ○ No

Please choose the package in lscsoft to install:

□ ImageMagick
□ R
□ R-coda

□ python-basemap
□ python-basemap-data
□ python-basemap-data-hires
□ python-basemap-debuginfo
□ python-basemap-examplesbr
□ python-matplotlib
□ python-pyRXP
□ python-pyRXP-debuginfo
□ python-sqlite2
□ pyxmpp
□ qgit
□ root
□ root-debuginfo
□ scipy
□ spr
□ spr-devel
□ xorg-x11-server-Xvfb

Whether to install LIGO Data Grid client

● Yes ○ No

Leave your email address here please

Email address: li.ziyang@yahoo.com.cn

Figure 4. Web-base user interfaces.

V. CONCLUSIONS

In this work, LIGO data analysis software is taken as a case study for demonstration of software virtual machines. A web-based customization system is also developed for end users. Scientific applications that require specific software support for data computing and analysis will in particular benefit from techniques proposed in this work.

Taking advantage of software virtual machines and cloud computing technology, future work will focus on hosted virtual machines. Instead of downloading image files and enabling virtual machines locally, cloud servers could actually provide hardware resources and host these virtual machines, and users could access resources via virtual desktop technology. Scientific clouds aims to enable scientific applications by providing both software virtual machines and hardware computational resources.

ACKNOWLEDGMENT

This work was supported by National Science Foundation of China (grant No. 60803017) and Ministry of Science and Technology of China under National 973 Basic Research Program (grants No. 2011CB302505 and No. 2011CB302805).

REFERENCES

[1] A. Abramovici, W. E. Althouse, et. al., "LIGO: The Laser Interferometer Gravitational-Wave Observatory", Science, Vol. 256, No. 5055, pp. 325 – 333, 1992.

[2] LIGO Scientific Collaboration. http://www.ligo.org.

[3] E. Deelman, C. Kesselman, G. Mehta, L. Meshkat, L. Pearlman, K. Blackburn, P. Ehrens, A. Lazzarini, R. Williams, and S. Koranda, "GriPhyN and LIGO, Building a Virtual Data Grid for Gravitational Wave Scientists", Proc. 11th IEEE Int. Symp. on High Performance Distributed Computing, pp. 225-234, 2002.

[4] DASWG, LIGO Data Analysis Software Working Group. https://www.lsc-group.phys.uwm.edu/daswg/.

[5] M. Litzkow, M. Livny, and Matt Mutka, "Condor – A Hunter of Idle Workstations", in Proc. of 8th Int. Conf. on Distributed Computing Systems, pp. 104-111, 1988.

[6] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Int. J. Supercomputer Applications, Vol. 11, No. 2, pp. 115-128, 1997.

[7] ROOT, A Data Analysis Framework. http://root.cern.ch.

[8] D. A. Brown, P. R. Brady, A. Dietz, J. Cao, B. Johnson, and J. McNabb, "A Case Study on the Use of Workflow Technologies for Scientific Analysis: Gravitational Wave Data Analysis", in I. J. Taylor, D. Gannon, E. Deelman, and M. S. Shields (Eds.), Workflows for eScience: Scientific Workflows for Grids, Springer Verlag, pp. 39-59, 2007.

[9] LDAS, LIGO Data Analysis System. http://www.ldas-sw.ligo.caltech.edu/doc_index/.

[10] LIGOtools. http://www.ldas-sw.ligo.caltech.edu/ligotools/.

[11] LSCsoft repository. https://www.lsc-group.phys.uwm.edu/daswg/download/repositories.html.

[12] J. Cao, E. Katsavounidis, and J. Zweizig, "Grid Enabled LIGO Data Monitoring", Proc. IEEE/ACM Supercomputing Conf., Seattle, WA, USA, 2005.

[13] LDG, LSC Data Grid. https://www.lsc-group.phys.uwm.edu/lscdatagrid/.

[14] D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. GarciaMolina, M. L. Klein, D. G. Messerschmitt, P. Messina, et. al., Revolutionizing Science and Engineering through Cyberinfrastructure, National Science Foundation Blue - Ribbon Advisory Panel on Cyberinfrastructure, 2003.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, et, al. "Above the Clouds: A Berkeley View of Cloud Computing", Univ. of California, Berkerley, Berkerley, CA, Technical Report No. UCB/EECS-2009-28, 2009.

[16] Virtualbox. http://www.virtualbox.org/.

[17] Expect. http://expect.nist.gov/.