Virtual Appliances for geospatial data processing

Christian Schwartze
Sven Kralisch
Wolfgang-Albert Flügel

Follow this and additional works at: https://scholarsarchive.byu.edu/iemssconference

Schwartze, Christian; Kralisch, Sven; and Flügel, Wolfgang-Albert, "Virtual Appliances for geospatial data processing" (2012). International Congress on Environmental Modelling and Software. 237.
https://scholarsarchive.byu.edu/iemssconference/2012/Stream-B/237

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Virtual Appliances for geospatial data processing

Christian Schwartze, Sven Kralisch, Wolfgang-Albert Flügel
Friedrich Schiller University, Department of Geoinformatics, Hydrology and Modelling, Jena, Germany
christian.schwartze@uni-jena.de

Abstract: Virtualization is increasingly taking on a key role in various system architectures which follow new platform concepts like Software as a Service (SaaS). This trend addresses more instant and short-term environments and comes with new methods and strategies for the distribution of mainly complex application stacks not only in large IT infrastructures. The paper presents how a so called Virtual Appliance can be set up in order to operate in virtual server environments using hypervisor software like Oracle VirtualBox. It will be shown that the use of standardized tools and interfaces for invoking geospatial data processing on servers brings a high level of flexibility regarding the development of clients requesting the services. As an example, we present the almost automatic delineation of spatially distributed model entities that serves as a component in the Integrated Land Management System (ILMS) [Králisch et al., 2012].

Keywords: Virtual Appliance; GRASS-GIS; Web Processing Service; Model pre-processing; Hydrological Response Units.

1 INTRODUCTION AND BACKGROUND

Virtualization in all its variations creates simulated computer environments to meet the requirements of streamlined IT infrastructures and modern platform concepts like the Cloud or SaaS. Contrary to the hype on outsourcing, the ideas and technical aspects behind “software on-demand” are not completely new: Terms as for example Virtual Machine and Virtual Machine Monitor were already outlined in different research studies by Popek and Goldberg [1974]. Similarly, providing software solutions via internet has been known for quite some time. In the late 1990s, Application Service Provider (ASP) have begun to turn their backs on traditional licensing by offering web-based infrastructures for external software usage. Even so, virtualization techniques (section 1.1) and the new ways in software packaging and distributing (section 1.2) only recently come together, not least due to the widespread availability of broadband internet and the increasing awareness of Green-IT.

1.1 Virtualization and hypervisors

The process where a physically existing system (hardware) is reproduced in order to run architecturally identical guests is called virtualization. In practice, virtualization can appear in at least two forms: Full or bare metal virtualization uses the hardware directly without the need for an operating system, which is not taken into account in this paper. In the case presented, host-based virtualization has to be used. This approach may be realized by a Virtual Machine Monitor (VMM), also referred to as hypervisor, operating either at hardware or operating system level of physical systems. Using the Oracle VirtualBox software as a VMM hosted by the operating system, the term type-2-hypervisor is very common. This approach is
very popular and compared to other virtualization mechanisms more flexible concerning the operating system, since the virtual machines (VM) may run on their own and native kernels and the VMM takes care of hardware emulation for all virtual guest machines.

Each virtual machine can be packaged and distributed as a Virtual Appliance. This feature was added in version 2.2 of VirtualBox and enables rapid deployment of virtual environments due to the standardized Open Virtualization Format (OVF), which basically encodes configuration settings of the VM and compress required hard disk images.

1.2 Virtual Appliance concept

Extending the definition from the Distributed Management Task Force [DMTF, 2010], a Virtual Appliance can be considered as a service that consists of a pre-installed, pre-configured operating system and an use-oriented application stack on top of it. The high level of interest in such ready-to-use, quite often customized and OVF-encapsulated software environments is reflected in the wide variety of available Appliance libraries, as for instance Turnkey Linux¹ and BitNami Stacks².

As a result, Virtual Appliances minimize installation and maintenance effort at customer site as long as the hypervisor comes with support for the OVF exchange format.

To build a Virtual Appliance from scratch can be more or less difficult depending on the applications, drivers or frameworks to be packaged, but current development tools like e.g. VMWare Studio lead and support the Appliance author during this process. Otherwise, there are comprehensive and universal sets of recommendations available in order to provide production-ready Appliances before shipping them [Ubuntu, 2011].

1.3 Building a geospatial Virtual Appliance

While already a few business units such as customer relationship management (CRM) or marketing (see e.g. www.salesforce.com) accept the centralization of software execution, geospatial applications featuring specialized services or functions on demand are still rare. One particularly prominent example is ArcGIS Explorer Online³ that expands the ESRI product range to include web-based geodata exploration and visualization.

Applying the VMM approach mentioned above, a GRASS-GIS based solution has been implemented that provides services for subdividing river catchments into Hydrological Response Units (HRU) [Flügel, 1995] for the purpose of hydrological process simulation. They represent areas with homogeneous process characteristics and are generally delineated by means of GIS overlay analysis. Various extensions on the HRU concept, such as for instance flow routing based topologies, have been taken into account to meet the requirements of spatially distributed model entities used in JAMS/J2000 model suite [Kralisch and Krause, 2006].

The software core needed for this purpose will be introduced in the following.

2 SOFTWARE COMPONENTS

The HRU-delineation Virtual Appliance is built on top of JeOS⁴ – a very much minimized Ubuntu Linux derivative that provides less packages and a tuned kernel optimized for the usage in hypervisors. As the operating system is shipped with the

¹ http://www.turnkeylinux.org
² http://www.bitnami.org/stacks
³ http://www.arcgis.com/explorer
⁴ http://help.ubuntu.com/community/JeOS
final and customized Virtual Appliance, it is worth choosing one with a small footprint and tailored to program packages actually required.

2.1 GRASS-GIS

The modularly structured and hybrid Open Source GIS has a large repertoire of built-in modules and tools including hydrological analysis programs [Neteler et al., 2012]. More than 400 single commands cover a wide variety of vector and raster operations partially equipped with options for watershed and waterflow related parameter processing. Not only the powerful hydrological features but also the scripting capabilities and programming interfaces led to the decision to perform all geospatial data processing in GRASS-GIS. Currently at least two methods are provided that enable straightforward scripting of own add-ons: either by using C programming language directly or via any scripting language that is able to manage and run external commands. In view of the component described in section 2.2, Python language was chosen to implement necessary routines for the HRU delineation. Furthermore, this procedure combines the rich features of Python with the comprehensive set of geo-processing function in GRASS-GIS. The latter includes, for example, the \texttt{r.mapcalc} module, which can be considered as a lightweight scripting tool for map algebra and proves to be very useful in realizing HRU operations by means of neighborhood-based filters.

2.2 OGC WPS

In addition to a standard Apache web server, the Virtual Appliance is configured for remote processing as addressed by the OGC \textit{Web Processing Service} [OGC, 2007]. The PyWPS software [Cepicky, 2007] implements the standard in Python language and is easy to setup and work in many Linux-based environments and thus also in JeOS. Accessing geospatial functions or to put it more precisely, HRU processing steps (figure 1, right), is performed by HTTP requests, that send required input parameters or data to the WPS server and immediately initiate the process. In the meantime, the client waits for a corresponding XML response document so that included results (literal data or complex outputs, e.g. images) can be further processed.

\begin{center}
\includegraphics[width=\textwidth]{figure1.png}
\end{center}

\textit{Figure 1.} The HRU delineation workflow as part of ILMS consists of WPS compliant Python modules (right)
PyWPS comes with native GRASS-GIS support but is not limited to the usage of geospatial functions. It is conceivable to use other software like statistical computing or geospatial libraries (section 2.3).

2.3 GDAL/OGR

The GDAL/OGR\(^5\) library can be understood as an abstract data model that offers unified access to a large number of raster and vector data formats. Even though it is used by GRASS-GIS for reading and writing geo-referenced features, another option would be use supplied command-line utilities. They are well suited for fast data processing and translation in shell scripts or as in this case WPS process implementations.

2.4 OpenLayers

Current JavaScript libraries often provide easy access to technologies for building modern and rich web applications. Developed and prepared with the intention to take complexity out of client-side Web 2.0 features, they are usually stored in the web root directory of the HTTP server. This is what was done with OpenLayers\(^6\) and ExtJS\(^7\), while the first focuses on dynamic mapping including cartographic components and the second makes available control elements for graphical user interfaces.

3 CLIENT DEVELOPMENT

The optimal client environment to access the HRU processing methods depends on the way in which the Virtual Appliance is provided. Various applications are possible but in the context of the Integrated Land Management System (ILMS) [Kralisch et al., 2012], only two should be considered and briefly outlined in the following. ILMS has been developed at the University of Jena to provide an integrated and modular software platform covering different steps of environmental system analysis and planning (figure 1). Besides the delineation of model entities, also referred to as ILMSgis, components for model development/application, automated object-based image classification and storage including analysis of watershed related data and timeseries contribute to the flexible and user-friendly ILMS workflow. For its easy distribution and direct use, all modules are bundled as an installation package which also contains QGIS\(^8\). The latter acts a client because of great capabilities to write own extensions to this free available GIS, for example in Python or C++. Hence, an intuitive and wizard-based plugin was developed that guides the user through the HRU delineation process and manages the communication between QGIS and the Virtual Appliance shipped with ILMS (figure 2, top). Usability is achieved as each of the GRASS/WPS processes is mapped to a tabbed module within the plugin and requests the required inputs. The modular concept of the delineation process guarantees that workflow can be extended as required. This applies to the second client implementation as well, where instead of Python a JavaScript based approach is used for the graphical interface (figure 2, below). Consequently, only a web browser is required and the client profits from libraries mentioned in section 2.4 above. Compared to the QGIS solution, the web client variant comes close to being a real SaaS application, no matter where the presented Virtual Appliance and its software stack is hosted.

---

\(^5\) [http://www.gdal.org](http://www.gdal.org)

\(^6\) [http://www.openlayers.org](http://www.openlayers.org)

\(^7\) [http://www.sencha.com](http://www.sencha.com)

\(^8\) [http://www.qgis.org](http://www.qgis.org)
4 CONCLUSIONS

The delineation of model entities is a typical use case for developing and setting up a Virtual Appliance to be run in hypervisors like Oracle VirtualBox. Some technical details were explained, but the number and kind of software that is included in the application stack and shipped with the Virtual Appliance depends on the problem which is to be addressed. Since ILMS modules are mainly based on Open Source software, state-of-the-arts methods and standards, it was appropriate to implement the HRU delineation workflow by means of GRASS-GIS and WPS. Other, standardized protocols such as Web Map Service (WMS) or Sensor Observation Service (SOS) may come into consideration, in particular for creating a geospatial Virtual Appliance.

REFERENCES


