

Virtual Appliances for geospatial data management and proLand Management System (ILMS)

Schwartz, Christian – Zander, Franziska –
Kralisch, Sven – Flügel, Wolfgang-Albert

Friedrich Schiller University, Department of Geoinformatics,
Hydrology and Modelling, Jena, Germany
christian.schwartz@uni-jena.de

SUMMARY

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 Virtual-Box. Using the example of two server-side components within the Integrated Land Management System (ILMS), it will be shown that the use of state-of-the-art methods, standardized tools and interfaces on servers enables different aspects of environmental system management, analysis and planning.

Keywords: watersupply of soil, value of waterdeficit, precipitation, irrigation, maize, systems of crop-rotation

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 and the new ways in software packaging and distributing (see following sections) only recently come together, not least due to the widespread availability of broadband internet and the increasing awareness of Green-IT.

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 (Net1) 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 Oracle 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.

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 also OVF-encapsulated software environments is reflected in the wide variety of available Appliance libraries, as for instance Turnkey Linux (Net2) and BitNami Stacks (Net3). 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 (Net4) lead and support the Virtual Appliance author during this process. Otherwise, there are comprehensive and universal sets of recommendations available in order to provide production-ready Virtual Appliances before shipping them (Ubuntu, 2011).

Building a geospatial Virtual Appliance for ILMS

Integrated tal system assessment and planning requires the application of different methods and thematic environment software tools in a sys and consistent workflow, often involving spatial data management, remote sensing and environmental modelling. The Integrated Land Management System (ILMS) has been developed at the University of Jena to provide a modular software platform covering different steps of environmental system analysis and planning that supports the integrated and sustainable

management of land and water resources on spatial scales ranging from upper local- to upper meso-scale. Four main modules exist in order to provide a flexible and user-friendly workflow.

1. ILMInfo is used for saving and analysis watershed related data and timeseries.
2. ILMGis implements the approach of Hydrological Response Units (HRU) by means of GIS methods and geospatial analysis to enable distributed environmental modelling that relies on the subdivision of river catchments into polygens.
3. ILMImage is designed for automated object-based image segmentation and classification of remotely sensed data as it is a required input in the hydrological system simulation process.
4. ILMmodel is based on the Jena Adaptable Modelling System (JAMS) – a framework for developing, running and analysing environmental simulation models.

The ILM modules mentioned above mainly use open source software and can be divided into two basic categories: On the one hand, pure desktop applications like ILMImage which comes as a plugin for the well-known QuantumGIS (Net5) software, or the JAVA-based framework ILMmodel that is shipped with different graphical user interfaces for model generation (JAMS Builder), parameterization/application (JAMS Launcher) and data analysis/visualization (JAMS Data Explorer). On the other hand, some ILM components work according to the client-server principle and use server-side functions and applications which are either accessible via standard web browsers (ILMInfo) or a customized QuantumGIS extension (ILMGis). This web- and service-oriented approach requires a new kind of software packaging and distributing in order to guarantee an error robust installation and deployment of all ILM components for the cooperating small and medium enterprises (SMEs) and other potential users. Subsequently, all server-side modules have been packaged as a Virtual Appliance and a central and common installation routine prepares the local system for ILM and ensures that necessary basic components are present (e.g. Python, QuantumGIS, JAVA). In this way, all ILM components can be executed locally where the so-called ILM Launcher allows a convenient and individual start of them.

The software core on which the server-side ILM modules are based is shown in *figure 1* and will be introduced in the following.

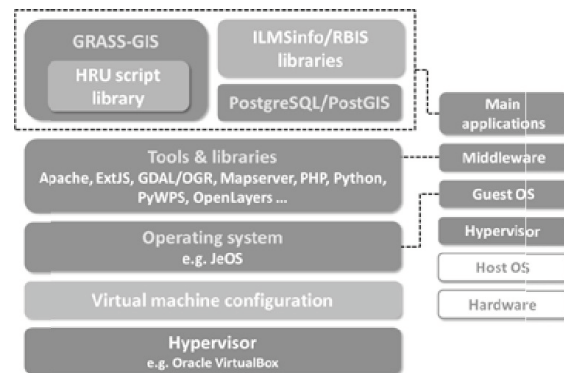
SOFTWARE COMPONENTS OF SERVER-BASED ILM MODULES

The Virtual Appliance is built on top of JeOS (Net6) – 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 final and customized Virtual appliance, it is worth choosing one with a small footprint and tailored to program packages actually required.

In addition to the operating system and a standard Apache web server including scripting language PHP, other special software on top of JeOS is needed for both server-side ILM modules. In general, the number

and kind of software that is included in the Virtual Appliance depends on the field of use, or to put it more precisely and service-oriented, on the problem which is to be addressed. For example, in order to implement a geospatial Virtual Appliance not only a GIS and different spatial analysis tools are needed, but also standardized protocols for serving georeferenced data over the network are of high importance. Web Map Service (WMS), Web Feature Service (WFS) or other OGC compliant standards may come into consideration according to the requirements and specific application context.

Figure 1: Main application stack of the Virtual Appliance used by ILMInfo and ILMGis



The individual software needed for ILMInfo and ILMGis will be introduced in the following subsections, but shared tools and libraries also exist. These include, for example, GDAL/OR (Net7) – an open source library that can be understood as an abstract data model offering unified access to a large number of raster and vector data formats. Another basic software used by both ILM modules is OpenLayers (Net8). The JavaScript library provides easy access to technologies for building modern and dynamic web mapping applications including cartographic features. As particular attention is paid to usability, OpenLayers based web clients will be used for ILMInfo as well as for ILMGis.

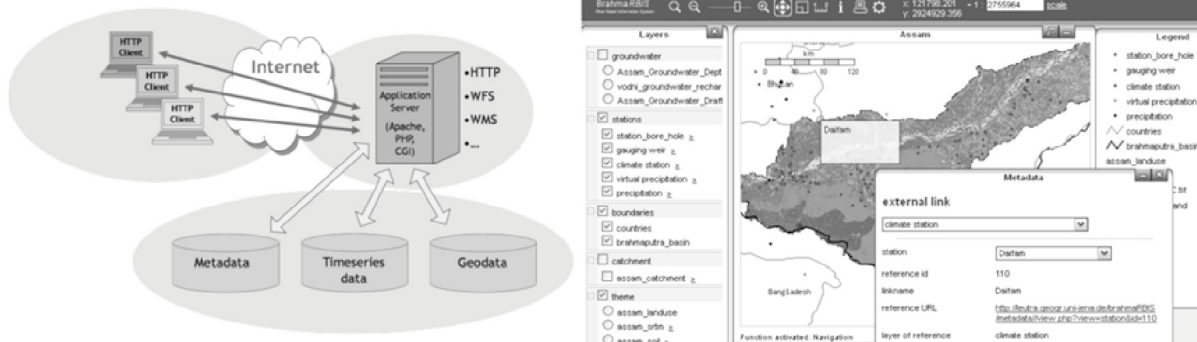
Data management with ILMInfo

Providing sophisticated data storage, management and interface functionality, the River Basin Information System (RBIS) is one of the core components of ILM. RBIS is a modular structured web-based information system that offers user-friendly interfaces for the management, analysis, visualization and presentation of different types of data in the context of multidisciplinary environmental assessment and planning. This includes besides metadata (RBISmeta) and datasets of certain content types [e.g. soil (RBISsoil) and vegetation (RBISbio)] also the support for measured or simulated time series data (RBISs), documents (RBISdoc), remote sensing data (RBISrs) and spatial data in raster or vector format (RBISmap). The system was designed to fit the requirements arising from integrated research projects as well as to offer a platform to preserve knowledge beyond their end. Since RBIS is accessible via the web it features a fine grained permission management based on data types (e.g. time series data, soil, geodata etc.)

and types of access e.g. view, edit, delete etc.) to ensure data security and quality control. In addition to the permission management, transparency is ensured by an access logging mechanism which is responsible for recording data manipulations.

The common layout of RBIS follows a standard 3-tier architecture design using the database management system PostgreSQL (Net9) as storage engine and the extension PostGIS (Net10) to add support for spatial data to PostgreSQL (*see figure 2, left*) (Kralisch et al., 2009).

Figure 2: Technical layout of RBIS (left) and geodata visualization component RBISmap (right)



Delineation of distributed model entities with ILMGis

For a second application of the Virtual Appliance approach, a GRASS-GIS (Net12) 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).

In order to address this challenge, GRASS-GIS had to be extended by scripts and additional software-components to cover the HRU processing steps outlined in *figure 3*. For example, the Virtual Appliance is configured for remote processing as addressed by the OGC Web Processing Service (WPS). 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. As PyWPS comes with native GRASS-GIS support, consequently almost all tasks are implemented as WPS compliant Python modules which form the HRU script library (*see software stack in figure 1*).

Besides the OpenLayers library, geodata management and visualization with RBISmap is also supported by the Mapserver (Net11) software package. Using Mapserver, RBISmap offers standardized geodata interfaces (WS, WMS) for external applications. Linking its geospatial data management, analysis and visualization features with other RBIS modules results in powerful functions for map-based search and selection of information about measuring stations, time series, observation points or hydrological catchments.

This procedure combines the rich features of Python with the comprehensive set of geo-processing functions in GRASS-GIS. More than 400 single commands and built-in modules cover a wide variety of vector and raster operations partially equipped with options for watershed and waterflow related parameter processing (Neteler et al., 2012).

If necessary, the modular concept of the delineation process guarantees that the workflow can be extended both at server and client side. To supplement the HRU workflow with special functions, additional Python scripts must be implemented and integrated into the existing processing library by specifying input and output parameters according to the WPS/PyWPS specification. The use of standardized interfaces like WPS also facilitates straight forward client development. Currently, QuantumGIS is used as a platform for delineating HRUs because of great capabilities to write own extensions, for example in C++ or Python. Hence, an intuitive and wizard-like plugin was developed that guides the user through the delineation process and manages the communication between Quantum GIS and the Virtual Appliance, both shipped with ILMGis. A second client implementation is still under progress which is based on JavaScript [OpenLayers, ExtJS (Net13)] and should only require a standard web browser (*figure 4*). Compared to the QuantumGIS solution, this web client variant comes close to being a real SaaS application, no matter where the presented Virtual Appliance and its software stack is hosted.

Figure 3: The HRU delineation workflow consists of WPS compliant Python modules

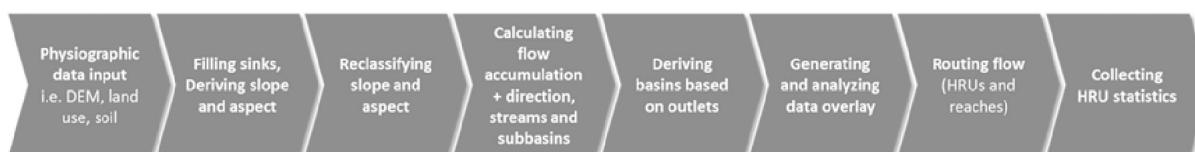
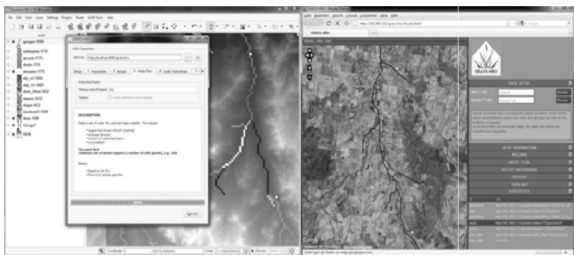


Figure 4: HRU delineation clients – QuantumGIS plugin (left) and OpenLayers-based web user interface (right)



CONCLUSIONS

Using the two ILMS components ILMSinfo and ILMSgis as an example, this paper illustrated the technical background and implementation of a Virtual Appliance in geospatial application areas with focus on the deployment

under Oracle VirtualBox. Due to the OVF standard designed for virtualized software environments under control of hypervisors, applications can be easily packaged and distributed to users. It is not just that the encapsulated solution stacks are installable without large installation costs and knowledge of included software, but also the additional security they provide since application errors have no impact on other hosted virtual machines. The implementation of both server-side components was carried out by open source products, ranging from the underlying and streamlined operating system to specially tailored or rather extended main software packages like PostgreSQL and GRASS-GIS as well as smaller but nonetheless important interfaces and tools.

The latter include, in particular, OGC standard protocols, geospatial and mapping libraries and also programming languages depending on the problem to be addressed.

REFERENCES

- Popek, G. J.–Goldberg, R. P. (1974): Formal requirements for virtualizable third generation architectures. *Communications of the ACM*. 17: 412–414.
- Cepicky, J. (2007): PyWPS 2.0.0 – The presence and the future. *Proceedings of the Workshop Geoinformatics FCE CTU*. 2: 61–64.
- Distributed Management Task Force – DMTF (2010): DMTF Standard DSP0243. Open Virtualization Format Specification 1.1.0. Ubuntu Community Documentation. <https://help.ubuntu.com/community/JeOS>.
- Flügel, W. A. (1995): Delineating Hydrological Response Units (HRU's) by GIS analysis for regional hydrological modelling using PRMS/MMS in the drainage basin of the River Bröl. Germany. *Hydrological Processes*. 9: 423–436.
- Kralisch, S.–Krause, P. (2006): JAMS - A Framework for Natural Resource Model Development and Application. *Proceedings of the iEMSs Third Biannual Meeting*.
- Kralisch, S.–Zander, F.–Krause, P. (2009): Coupling the RBIS Environmental Information System and the JAMS Modelling Framework. [In: Anderssen, R.–Braddock, R.–Newham, L. (eds.) *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*.]
- Neteler, M.–Bowman, M. H.–Landa, M.–Metz, M. (2012): GRASS GIS – A multi-purpose open source GIS, *Environmental Modelling & Software*. 31: 124–130.
- Net1: <http://www.virtualbox.org>
- Net2: <http://www.turnkeylinux.org>
- Net3: <http://www.bitnami.org/stacks>
- Net4: <http://vmware.com/products/studio/overview.html>
- Net5: <http://www.qgis.org/>
- Net6: <http://help.ubuntu.com/community/JeOS>
- Net7: <http://www.gdal.org>
- Net8: <http://www.openlayers.org>
- Net9: <http://www.postgresql.org>
- Net10: <http://postgis.refrains.net/>
- Net11: <http://www.mapserver.org>
- Net12: <http://www.grass.fbk.eu>
- Net13: <http://www.sencha.com/products/extjs/>