A New Approach to Affordable and Reliable Cardiology PACS Architecture Using Open-Source Technology

P Marcheschi[1], A Ciregia[1], A Mazzarisi[2], G Augiero[1], A Gori[3]

[1]Fondazione Toscana Gabriele Monasterio CNR-Regione Toscana, Pisa, Italy
[2]CNR, Pisa, Italy
[3]ASL1, Massa, Italy

Abstract

In the past ten years we followed the expansion of cardiology imaging technology and how it relates to integration, tele-consulting and standards. The new frontier of Picture archiving and communication systems (PACS) is expanding and require more technology and resources. Cardiology imaging is evolving and needs even more resources than before. The present work proposes a new approach to manage a large cardiology PACS installation using new open-source facilities on affordable hardware creating a reliable system. For this kind of application we moved our core system from Linux platform to the new OpenSolaris unix and to novelty Functions like Solaris Containers, ZFS file-system and Fair Share Scheduler, in order to achieve server consolidation, ease of use and reliability.

1. Introduction

PACS, Picture archiving and communication system are the most needed and expensive instruments in a modern hospital, this because the need for iconography representation of human body is growing in every field of health-care. PACS needs performance, stability and security in order to treat thousand of images a day, for every modality in an hospital: it needs storage space, ability to compress and decompress images in parallel and very high band throughput to satisfy health-care professional needs. One of the most used open-source PACS software is dcm4chee [1] [2] this because is written in the platform independent java language, and is a feature complete implementation of an IHE PACS system. We evaluated different lossless and lossy compression schema in order to achieve fast client/server response and to overcome low bandwidth connections limits. Moreover actual local hi-bandwidth environments required logical and physical network separation by imaging modality by the use of VLAN support and spanning tree technology. In order to assure the optimal service availability, the system is constantly monitored to quickly rise alerts and to graph system performance and status. Particular attention is posed to IT security, achieved with a cluster of two Linux firewall with network filtering support. At the moment we are sharing with the proposed system over 23 millions images, coming from MRI, CT, PET, ECHO, CR/DR and XA imaging laboratories.

2. Methods

The classic approach to server consolidation and deploying is to use windows or Linux based architecture, with Vmware virtualization solution [3]. Among different flavors of unix we can choose one of the myriad of linux distributions but when we tried Solaris/OpenSolaris [4] we realized that an alternative to these popular operating systems exists and we found in solaris unix what we were looking for. A very reliable and full of features right for our need. This Operative System is not new and has been used for years as a commercial UNIX in contrast to open source solutions based on linux. First of all ZFS (Zettabyte File System) [5], is a 128 bit file system, with end to end checksum integrity, built in compression, volume manager and unlimited snapshot features. It offers unprecedented ease of use and reliability. Solaris is the ideal solution for consolidate multiple image managers in a single server due to its native ability to support server virtualization without the need for different products and technologies. It uses the concept of zone, or container, that is the OS environment [6]. It is a light-weight OS virtualization with only one kernel per hardware, virtual networking and with a complete isolation between zones or virtual machines. OpenSolaris is a bit ahead of Solaris, it has a better support for peripherals, and has newer ZFS and kernel features, also the standard version is mature for production use. The choice of a new OS for production use is difficult because we had to develop new skills. In order to do so, SUN Microsystems [7] helped us with a free training workshop, to be able to achieve, base necessary skills, to start the project. The main goal of the project was to develop a cheap, fast and reliable server configuration in order to store and redistribute images coming from our multiple modalities (MRI, CT, PET, CT-PET, Ultrasound, CR/DR and XA imaging laboratories).

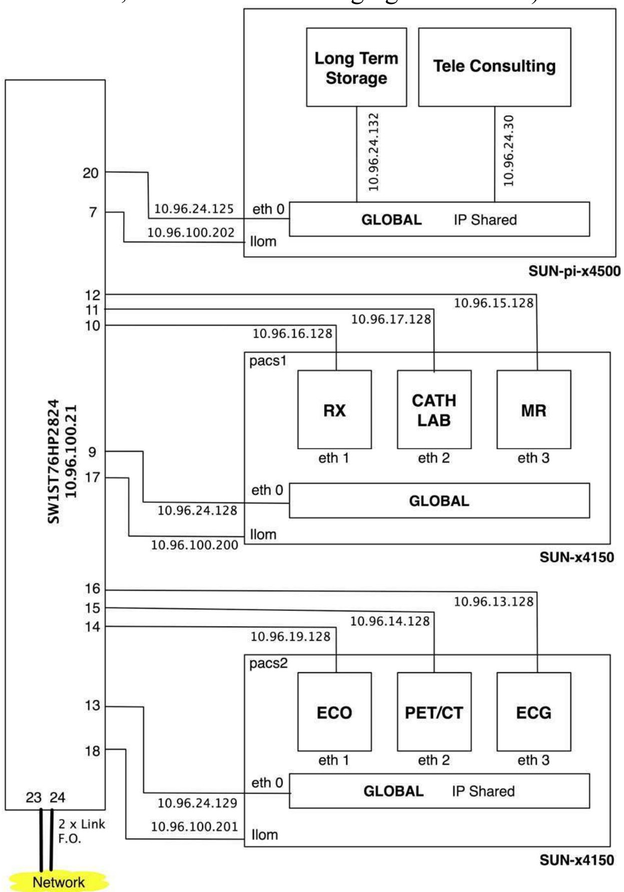


Figure 1. Server Farm PACS Architecture

The architecture of the system is made by three solaris servers two for the front-end, directly connected to the consoles of the modalities, and one for the historic PACS archive, that has a big storage pool (24 TB). We decided to redesign also our network using vlans. In order to develop this kind of isolation between the first level of the PACS, the front-end, we needed a way to virtualize and consolidate servers, instead of buying one machine for every modality we choose to use the Solaris containers virtualization solution. This solution gives compartmentalization and proportional use of resources trough FSS (Fair share schedule) scheduling software, which controls the allocation of available CPU resources among workloads. A zone is a virtualized environment with it's own root, but with a lack of a private dedicated kernel, it uses instead the host kernel, this is a big performance gain. There is also the possibility to create a linux branded zone with the same level of abstraction, with an emulated kernel that map primitive linux system calls to native solaris calls. This in order to migrate effortless a complete linux environment. Another possibility offered by this OS is the file-system snapshot feature, available only in more expensive NAS (Networked Attached Systems) or storage solutions, this makes possible making backups of the overall system or a particular subdirectory, in different point in time and while the system is running in production. With this facility one can recover from accidentally removed files , or is possible to backup the DB without service interruption. We used this feature to upgrade the image manager software with only a software restart. Also the purpose of this kind of architecture is to split the image compression pipeline in two different steps. Actually, to be able to store more images on storage, we adopted the JPLL (Jpeg Lossless) [8] compression schema to achieve 3:1-2:1 compression ratio. This heavy duty task is executed by the front-end servers that have low storage capacity, but high processor power. Once compressed, the images are forwarded to the historic archive in night batches. The IHE [9] [10] image manager is built upon the well known open-source project dcm4chee, a few modification were added to the standard distribution, but only in logging and control scripts because Solaris uses a different service facility called SMF (Solaris Management Facility). The dcm4chee software is rock solid, and is full of features comparable to commercial one.

In the network context, each zone uses a different physical network interface that exists on the server. We allocated a different network subnet for each network interface. On each network segment there is a modality that sends the medical study to the relative Solaris zone. This choice, implemented during the network design, assures the best performance and security in network communications. Specifically, at the end of each study, each modality sends a huge amount of data to its dedicated zone. Network traffic is composed of a number of large clinical pictures transiting on the network. Generally the implementation of routing techniques requires single frame delivery to the gateway of the subnet to which they belong, through a Layer 2 protocol, the gateway has to unpack the frame and analyze information to Layer 3 then forward the package to the proper interface where the recipient is present. The activity of unpacking and packing requires a short delay for the router computation in order to create each package, lowering the performance and adding a slight delay in communication. In addition, the router become a new point of failure in communication between the modality and the Image Manager. The above problems are avoided by positioning modality and Image Manager on the same subnet.

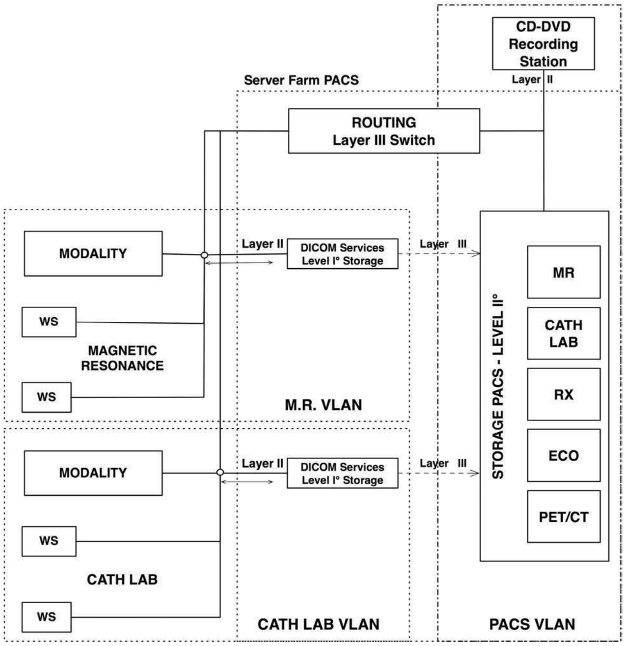
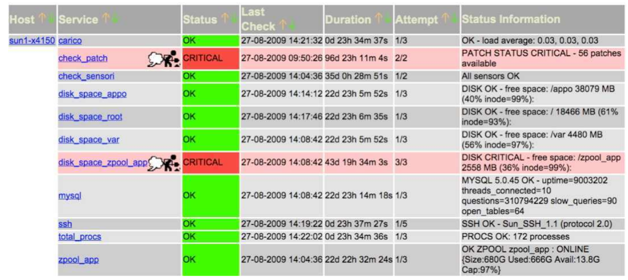


Figure 2. Organizational Model

Surgeons, doctors and Technicians rely on PACS 24x7 availability, hence we must provide quick response in case of system malfunctioning. In addition we have considered worthwhile, having an historical view of servers performance and a day by day usage charts. To achieve such goals we used two Open Source software: Nagios [11] and Cacti [12]. In addition, for logs management produced by Dcm4chee archive manager itself, we used another Open Source software: SEC (Simple Event Correlator) [13]. Nagios is a powerful and widely used network monitoring system. The strength of this useful application is the large base of community supported add-ons, in particular the plug-ins. They are standalone piece of softwares that plug into the main application in order to do specific task. One of this plugin is the DICOM service checker that allow to monitor the availability of a DICOM service class provider (SCP).

Thanks to this kind of plug-in we can easily verify every aspect of our PACS network and in particular the status of our services. The following are the parameters checked every 5 minutes: Server Load, Hardware sensors, Disk Space, Number of processes, Raid Status, Database Status, DICOM services. We can for example actively monitor the storage status, or if a service is running and if the server load is too high. We can have a look to the hardware sensors, and we can check if the DICOM SCP service is answering through the network. If there is a problem, such as the free disk space is becoming to small (we can define various thresholds) or the DICOM SCP is no more answering, Nagios will



Service Status Details For Host
'sun1-x4150'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
sun1-x4150	carico	OK	27-08-2009 14:21:32	0d 23h 34m 37s	1/3	OK - load average: 0.03, 0.03, 0.03
	check_patch	CRITICAL	27-08-2009 09:50:26	96d 23h 11m 4s	2/2	PATCH STATUS CRITICAL - 56 patches available
	check_sensori	OK	27-08-2009 14:04:36	35d 0h 28m 51s	1/2	All sensors OK
	disk_space_appo	OK	27-08-2009 14:14:12	22d 23h 5m 52s	1/3	DISK OK - free space: /appo 38079 MB (40% inode=99%):
	disk_space_root	OK	27-08-2009 14:17:46	22d 23h 6m 35s	1/3	DISK OK - free space: / 18466 MB (61% inode=93%):
	disk_space_var	OK	27-08-2009 14:08:42	22d 23h 5m 52s	1/3	DISK OK - free space: /var 4480 MB (56% inode=97%):
	disk_space_zpool_app	CRITICAL	27-08-2009 14:08:42	43d 19h 34m 3s	3/3	DISK CRITICAL - free space: /zpool_app 2558 MB (36% inode=99%):
	mysql	OK	27-08-2009 14:08:42	22d 23h 14m 18s	1/3	MYSQL 5.0.45 OK - uptime=9003202 threads_connected=10 questions=310794229 slow_queries=90 open_tables=64
	ssh	OK	27-08-2009 14:19:22	0d 23h 37m 27s	1/5	SSH OK - Sun_SSH_1.1 (protocol 2.0)
	total_procs	OK	27-08-2009 14:22:02	0d 23h 34m 36s	1/3	PROCS OK: 172 processes
	zpool_app	OK	27-08-2009 14:04:36	22d 22h 32m 24s	1/3	OK ZPOOL zpool_app : ONLINE {Size:680G Used:666G Avail:13.8G Cap:97%}

Figure 3. Nagios: Screen Service Status Detail For Host

rise an alert on his dashboard, and it will send a notification via e-mail to the configured group of recipients (Nagios can be configured to also send notifications via SMS or instant chat systems). An extremely useful program is the Nagios Firefox plugin, that allows a constant view over PACS alerts on our desktop.

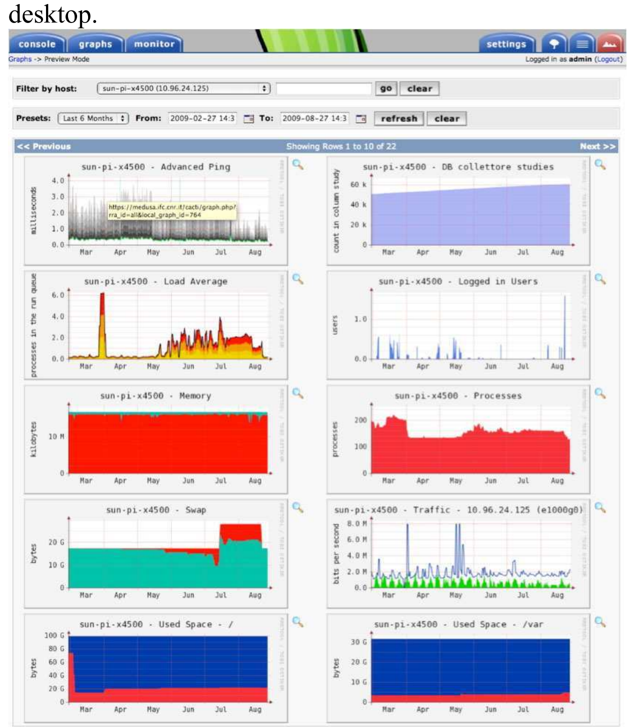


Figure 4. Cacti screen

Cacti is an RRDtool (Round Robin Database Tool) based graphing solution. RRDtool is a industry standard created for data logging and graphing system of time series data. The Cacti application is configured in order to collect data from the PACS network and it renders it on its dashboard. The control panel looks nice and intuitive, and like Nagios has many community contributed scripts useful to gather any kind of information. For example we graph server load, network usage, disk space utilization, swap space, various MySQL DB aspects. Data is mainly collected using the SNMP (Simple Network Management

Protocol) protocol.

Having a look to data plotted in graphs, we can have a global view of the overall system activity of the PACS, and we can estimate also when to do small maintenance painlessly for example; looking at the network graphs we can understand if the LAN is overloaded; with home made scripts we can also monitor the growth over time of the PACS studies stored in the database.

The Dcm4chee software is written in Java, and it uses log4j for logging informational or error events. In case an error exceed a threshold, log4j can be configured to send an email message with the log attached to it. In the past we have had some problem due to the large amount of log produced, very often useless, uninteresting or duplicated. Configuring log4j for a smarter log notification can be a pain for non programmers, so we settled to use another tool called SEC that is a very smart and powerful daemon written in Perl. This tool attend to parse log files using Regular Expressions or simple string comparison; if a match is encountered, we can associate an action to this event: send a mail, ignore further match for an amount of time, group together the event to notify, drop the event, and so on. SEC is therefore a very valuable addition to the monitoring tools that we made up to follow the PACS overall health.

3. Results

One of the biggest advantage of the overall system is the reliability and the price lower than proprietary systems. With this system we are storing over 23 millions images with 60 thousand studies, and this figures are growing every day. One of the disadvantage of this system is the need for prepared technician to understand what is going on with the system itself, the personnel must be familiar with Solaris unix system administration, slightly different from linux one, with java virtual machine and web work-flow since the software is based on java award winning technology and is web based.

Another disadvantage, but this is common to other system is the crucial phase of system upgrade. Actually this kind of system has to be upgraded at least every 6 months for system operative restart and image software upgrade, this phase is delicate in order to reduce system off-line time. We decided to face this problem making a replica of the virtual containers in order to have the possibility to recover to the old state in the case of problems. The system is very stable and compatible with all of our modalities and backup system (RIMAGE from GE) and is comparable with most blazoned systems in terms of network response and features.

4. Discussion and conclusions

At the moment we have replicated the architecture in another site in Massa that is our reference cardiology surgical hospital. This has the same structure but is serving over 4 millions of images in over 35 thousand studies. This architecture seems to be reliable and stable, the choice of using Solaris Operative system and mysql database along with dcm4chee server software, was the right choice to sustain very critical tasks with one tenth of the budget needed for a commercial PACS.

References

[1] Zeilinger G. Open Source Clinical Image and Object Management. http://www.dcm4che.org/ *accessed sept 2009*.

[2] Warnock MJ, Toland C, Evans D, Wallace B, and Nagy P. Benefits of Using the DCM4CHE DICOM Archive. *Journal of Digital Imaging*, vol. 20, 2007:125–129.

[3] VMware Cloud Computing with Virtualization, Green IT, Virtual Machine & Servers. http://www.vmware.com/go/vmware-go/ *accessed sept 2009*.

[4] Solaris Operating System. http://www.sun.com/software/solaris/ *accessed sept 2009*.

[5] Strobl R. ZFS: Revolution in File Systems. *Sun Tech Days* 2008-2009.

[6] Price D and Tucker A. Solaris zones: Operating system support for consolidating commercial workloads.

[7] Sun Microsystems. http://www.sun.com/ *accessed sept 2009*.

[8] Wallace GK and others. The JPEG still picture compression standard. *Communications of the ACM*. vol. 34, 1991:30–44.

[9] Grimes S. The challenge of integrating the healthcare enterprise. *Engineering in Medicine and Biology Magazine. IEEE*, vol. 24, 2005: 122-4.

[10] Elion J, Becker T, Keller A, Simon R, Sippel T, Solomon H, and Weintraub W. Integrating the healthcare enterprise (IHE) interoperability for cardiology: year 1 demonstration. *Computers in Cardiology 2005*, 2005:691-694.

[11] Nagios - The Industry Standard in IT Infrastructure Monitoring. http://www.nagios.org/ *accessed sept 2009*.

[12] Cacti: The Complete RRDTool-based Graphing Solution. http://www.cacti.net/ *accessed sept 2009*.

[13] Vaarandi R. SEC - a lightweight event correlation tool. *IP Operations and Management. 2002 IEEE Workshop on*, 2002:111-115.

Address for correspondence

Paolo Marcheschi
Fondazione Gabriele Monasterio CNR
Via G. Moruzzi 1 56126, Pisa, Italy.
paolo.marcheschi@ftgm.it