

Emanuele Petrucci

<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

Mar 2011 - Present **System Automation Engineer** **Automat Snc/Freelance Terni, IT**

Employed in Automat Snc, a company which core business is in the industrial automation field. Work activities comprehend installation and maintenance of ABB & Ansaldo industrial drives, inverters and converters and software development, too.

I held a qualification course in Ansaldo Industrial Systems SpA about their Answer Drives photovoltaic systems. The pictures below show the inverter Solargate PV8M350 with closed and opened doors. This specific model provides 315 Kwatt operating in Medium Voltage (range 430-760V, Vmax 800V).



Automat also supply electric panels wholly designed, assembled and programmed internally. PLC (Programmable Logic Controller) usually are the core logic of these panels, used for a myriad of automation processes, eg. to control motor activation and speed, move robot arms, rotate pulleys, control conveyer belts, read digital/analog inputs to perform, when specific thresholds are met, actions like lighting lamps, opening safety valves or stop engines.

The most used PLCs are branded Siemens. I had the opportunity to use and learn the software Siemens Step 7 to program Siemens PLCs.

Other PLCs used are the Wago and



The PLC Siemens S7

Emanuele Petrucci

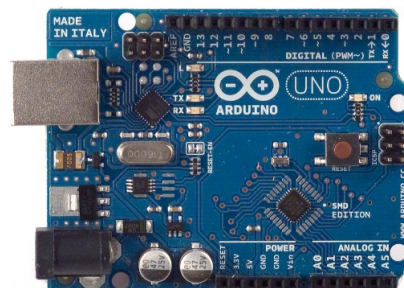
<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

These experiences helped me to increase the passion toward micro-controllers, especially Arduino, defined as an “*Open Source electronics prototyping platform*” (<http://arduino.cc>). Substantially, it is an electronic board with a 32KB flash memory, an EEPROM, a crystal oscillator, a low-power 8-bit Atmel CPU and several digital/analog input/output ports.

Arduino has a simple development environment and is programmed in a language C-like or Assembly.



Arduino UNO smd

Besides, recently I had the opportunity to work as Fortran/C programmer on OpenVMS for the migration and extension from Digital VAX to DEC Alpha and HP Integrity of the chemical laboratory program of ThyssenKrupp (<http://www.acciaiterni.com>, <http://www.thyssenkrupp.com>, <http://en.wikipedia.org/wiki/ThyssenKrupp>) a world-leader steel factory.

Even though OpenVMS is fallen into disuse, this experience gave me the occasion to understand the (or one of the) forefather of high technical solutions present in modern Operating Systems, implementing since its born in the last years of 1970ies concepts like virtual-memory, high level of security, process priority, clustering, hardware partitioning, distributed file system.

In the meanwhile still works as freelance.

Emanuele Petrucci

<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

Apr 2009 - **Computer Professor &**
Feb 2011 **Consultant**

Freelance
Terni, IT

Teach as professor in several computer courses for professional private schools and for the Region Industrial Associations (API/CONFAPI/APIFORM, (see www.apiterni.it). In the meantime works also as freelance for companies and privates, having also the opportunity to deepen web development knowledge, coding in HTML, CSS and Javascript.

Emanuele Petrucci

<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

Oct 2008 - Senior Linux & Windows
Feb 2009 SysAdmin

Movidius Ltd
Dublin, IE

Movidius is a fabless semiconductor company whose technology provides unique video editing and post production capabilities for the creation of User Generated Content for mobile social networking (see www.movidius.com). Responsibilities include the management and administration of Windows clients and Linux servers (RHEL, CentOS, Ubuntu). Contracted to implement from scratch:

- LDAP server in mirror mode & synch replication as unique common authentication system for company's users and services (Apache, Bugzilla, MySQL, SVN)
- BACULA backup server for both Windows clients and Linux servers
- ACP UPS client/daemon system, new Bugzilla r.3.2, VNC for server connections from Windows

May - Sep
2008

Linux IT Engineer

Silicon & Software
Systems (S3)-Dublin, IE

Hardware technology used:

- ✓ DELL PowerEdge 1850/1950/2850
- ✓ SUN Solaris servers
- ✓ NetApp filers
- ✓ StorageTek SL500 Modular Library System

Software technology used:

- ✓ RedHat Enterprise Linux 3, 4, 5 (32 & 64 bit)
- ✓ EMC Legato NetWorker 7.x
- ✓ LSF Compute Cluster Server
- ✓ Windows Active Directory - Services for Unix - Centrify

S3 (see www.s3group.com) is a 20+ years company which provide worldwide professional systems, embedded software and leading-edge silicon design for consumer, wireless and related applications.

The working environment is a mixed Windows/Linux based on Active Directory, Windows Services for UNIX and Centrify.

Daily work includes the support to all silicon & software engineers and the ticket administration, the management, administration and support for all the Linux RHEL environments in Dublin.

One of the most sensible system to administrate is the LSF Compute Cluster Server, massively used by silicon engineer for their integrated circuits and chipset design & development. LSF is composed by a server farm of DELL PowerEdge 1950 and IBM System-x with RHEL 4.

I was also delegated of the vital tasks of the management of all the company's backups, made using Legato NetWorker on LTO Ultrium 2 Tapes with a StorageTek SL500 Modular Library System.

Since when I was employed I tried to understand in which way I can contribute to improve S3's IT systems and for now my efforts yields the following:

- dissh, a Distributed Secure Shell, a program to facilitate the system administration allowing to execute one or more commands on any customized list of unix servers and having the results eventually logged and sent via e-mail.
- Complete renewal of the backup scripts which resulted in improved reliability, stability and clearness of the weekly backup of all S3 data.
- A nourished list of CGI scripts used in the intranet for: company project management, project size report (using SNMP with filers), S3

Emanuele Petrucci

<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

computer list to maintain updated, and others with diverse purposes.

Emanuele Petrucci

<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

2007

**IBM AIX & Linux System
Administrator**

**Telecom Italia Spa
Roma, IT**

Hardware technology used:

- ✓ IBM System p5 595 and eServer pSeries 690 in LPAR environments
- ✓ SUN servers intel-based

Software technology used:

- ✓ IBM AIX 5.3
- ✓ RedHat Enterprise Linux 3 and 4 (32 & 64 bit)
- ✓ SUSE Linux Enterprise Server 10

Telecom is Italy's largest provider of telecommunications products and services. Contracted to manage and administrate Telecom's IBM systems in LPAR environments when failures occurs and/or remote management result to be impossible.

Because the wide experience also with Linux, I was asked also to administrate and manage Linux RHEL and SLES servers in the several sites of Telecom Italia in Rome.

I was been involved also in the support for SUN systems, helping my Solaris colleagues, and having so the great opportunity learn some more about those systems.

2006

**Linux Specialist &
Platform Architect**

**Anritsu Group
Rome, IT**

Hardware technology used:

- ✓ HP Proliant DL 360/380/385/585
- ✓ DELL PowerEdge 1850/2850

Software technology used:

- ✓ RedHat Enterprise Linux 2.1 and 4 (32 & 64 bit)
- ✓ Oracle DB 10g
- ✓ Oracle Application Server 9i/10g
- ✓ Data Ware House applications

Anritsu (see www.anritsu.com) is a conglomerate of technology-based companies throughout the world headquartered in Japan. NetTest (now Anritsu Solutions SpA) serves as a research and development arm for large scale telecommunications systems monitoring and control (currently 3rd in overall market share). Contracted during the transition of NetTest's systems to the new owners.

● **Linux Specialist & Platform Architect**

After performing a needs analysis, determined that by upgrading system software and hardware, Anritsu could eliminate the then current storage limitations and at the same time greatly improve system throughput. Once approved, the idea I had was to create and implement the new generation 3 of this "platforms": the DWH Platforms either 64 and 32 bit (DWH64 and DWH32 Platforms).

My basic idea was to remove all complex and nested operations to perform at the time actual DWH Platforms version 2.x.y, based on Red Hat Linux Enterprise 2.1, so that field offices could independently install the new OS Red Hat Linux Enterprise 4.0 and application software on new hardware and be up-and-running immediately, and at the same time improve security, stability and avoiding any user interaction that could result jeopardizing for a successful and clean installation.

Of course this doubled my work and my efforts in order to prevent anything could be diseasing or interfere with the right functioning, thus the job required a big and a meticulous planning.

These new platforms so were automated that the resulting installation guide was trimmed to 1/3 the previous size and ALL installations were completed on time & without failure.

I studied an installation procedures for all Platforms extremely simple for the field engineers, that consist, essentially, in starting the server with the Platform DVD media loaded in the DVD reader unit, after having

assured that the first boot unit is the DVD reader, and when the boot screen appears type in the name of the Platform wanted to install, followed by the Enter key (see Screenshot 1 and Screenshot 2).



Screenshot 1: Bootstrapping the system with the DWH64 Platforms 3.0.0 media the operator can choose to install Oracle DB, Oracle Client or XDR Platform.



Screenshot 2: The DWH32 Platform media provide the Oracle AS installation.

As visible in Screenshots 1 and 2, the system engineer can choose from different platforms to install: in the DWH64 Platforms media the

Oracle DB 10.2.0.2 (**DB**), Oracle Client (**OC**) and the Generator (**XDR**) Platforms 64 bit; in the DWH32 Platform media the Oracle Application Server (**AS**) Platform 32 bit.

Another Platform, intentionally **hidden** from the list, is even present in both DVD media, but those are the installations I used to generate my personal developing and general purpose platforms.

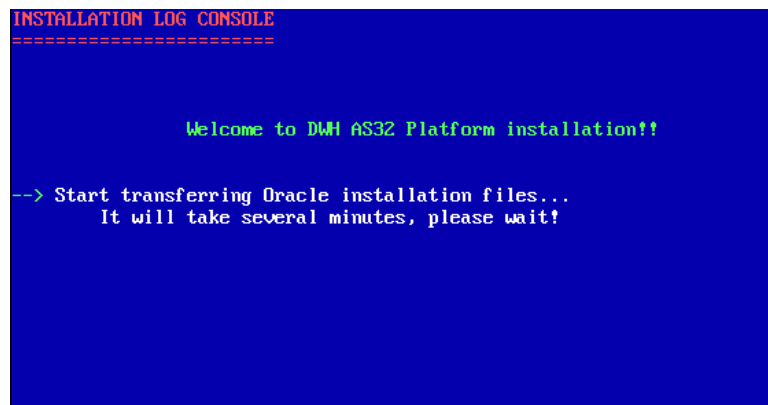
Pressing “F2” key the user will see a brief description about DWH Platforms (see Screenshot 3).



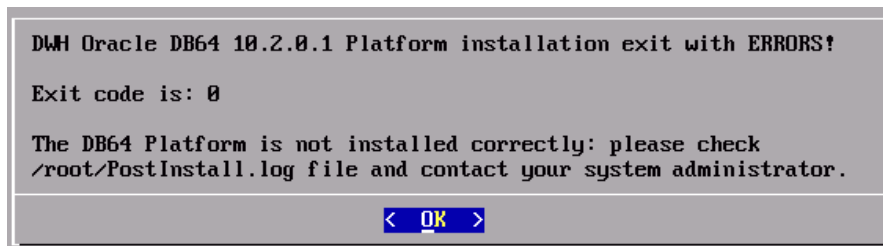
Screenshot 3: Pressing the key F2 the user is showed a brief description about the DWH32 AS Platform.

During all phases of all Platforms installation procedure the user is informed about the steps the system is performing. After the RHEL packages installation, pressing together keys **Alt+F12** the user can switch on a detailed log of all operations performed by installation procedures in real-time (see Screenshot 4) and can see if everything is going good or not. To simplify this task I added a colorization system to the logs, so that the user can see at glance if all operations ends successfully, marked in green, or ends with an error, marked in red.

In any case the installation procedure is smart enough to verify if something goes wrong, so the user is warned about on the main view too, accessible pressing **Alt+F1** keys (see Screenshot 5).



Screenshot 4: The installation log console is used to provide the user a high detailed status of all the operations performed during the installation process.



Screenshot 5: If something goes wrong during the installation procedure, the system warn the user and provide exiting error codes. This message is even reported into log file which shows in detail the operation that failed and is visible in the installation log console.

Depending on which Platform user chooses, the system automatically reboot one or more times before end the installation procedure ends.

At the end of the Platform installations the operator is asked to choose the time zone before the system release the control to the user: that is because the Platforms are spread to Anritsu customers all around the world (see Screenshot 6).



Screenshot 6: At the end of the installation phase the operator is asked to set the local timezone.

During the whole installation procedure the operator **cannot obtain the system control**: this action has been foreseen to avoid completely any kind of accidental interruption problem, either pressing **Ctrl+C**, either **Ctrl+Alt+Del** !

Either HP and DELL servers, whose models are defined in the Platform Technical Baseline, are automatically recognized by the installation procedure and the relative drivers will be used by the system. To let the user knows which kind of server he is using, the “/root/bin/.hw_spec” hidden file has been generated with the server brand name.

Supported servers in Platforms 3.x.y are:

- ✓ HP Proliant DL380 G4
- ✓ HP Proliant DL385
- ✓ DELL PowerEdge 1850
- ✓ DELL PowerEdge 2850

Previous Platforms 2.x.y used both DELL servers and HP Proliant DL380 G4, plus HP Proliant DL360.

```
- tty1 -  
DWH AS32 Platform 3.0.0 for MC 6.1  
Kernel 2.6.9-42.ELhugemem  
HP i686 server  
  
o9ias login: █
```

Screenshot 7: The login prompt of the DWH32 AS Platform. Are clearly showed: the kind of Platform and its version, kernel version, server brand, OS type and hostname.

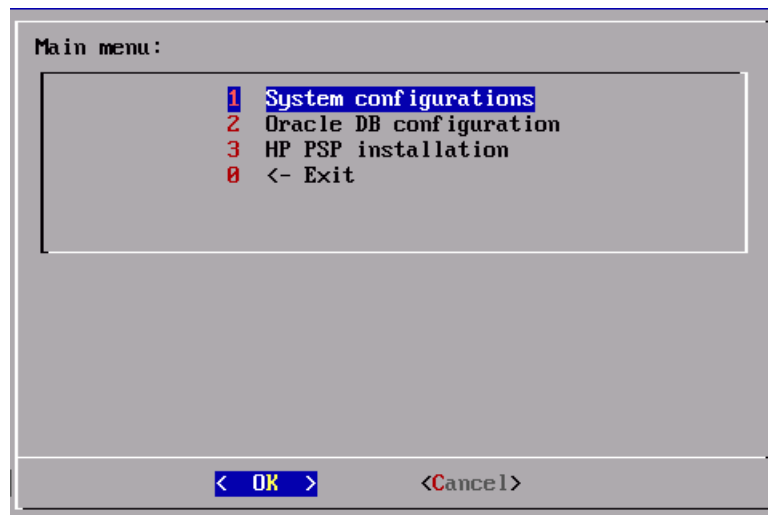
```
- tty1 -  
DWH DB64 Platform 3.0.1 for MC 6.1  
Kernel 2.6.9-34.0.2.ELsmp  
HP x86_64 server  
  
DB64U300 login:
```

Screenshot 8: The login prompt of the DWH64 DB Platform. Are clearly showed: the kind of Platform and its version, kernel version, server brand, OS type and hostname.

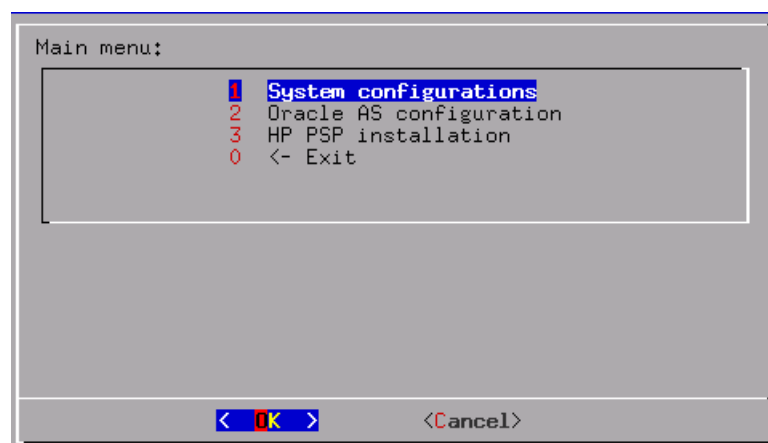
The operator is asked to not perform any kind of system administration, to avoid any little possibility to disease the system in any manner. Thus it has been realized a user friendly tool to execute all the needed server customizations: the key of the server management is the bash script named "PostInst.sh". It is a menu driven *ncurses* program whose power reside in its simplicity and the big versatility to be executed remotely via shell. That is the best way to have maintenance and control over a server. The program is menu contextual and checks every single operation avoiding the possibility to bring instability on the server. E.g., if the HP PSP installation has already been performed (see Screenshot 9), the menu will show that it is already done.

It has a log file in which is logged any operation performed through the script, specifically stressed with colors: green if it has been successful (visible in the installation log console, Screenshot 4), red if an error arose.

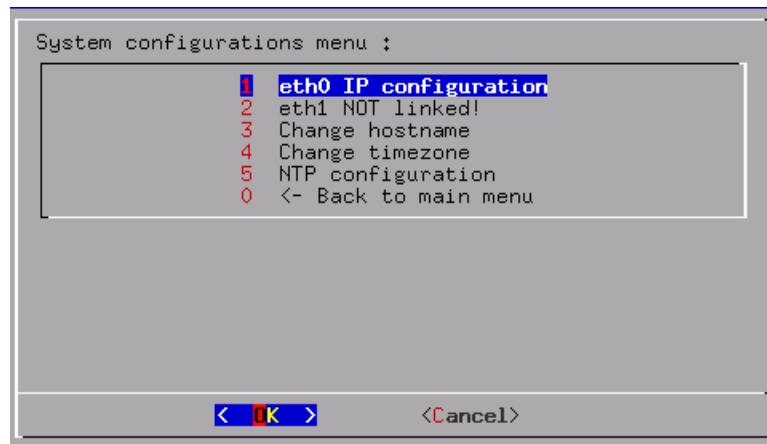
Hereinafter there are several screenshots showing some functions that could be done using PostInst.sh.



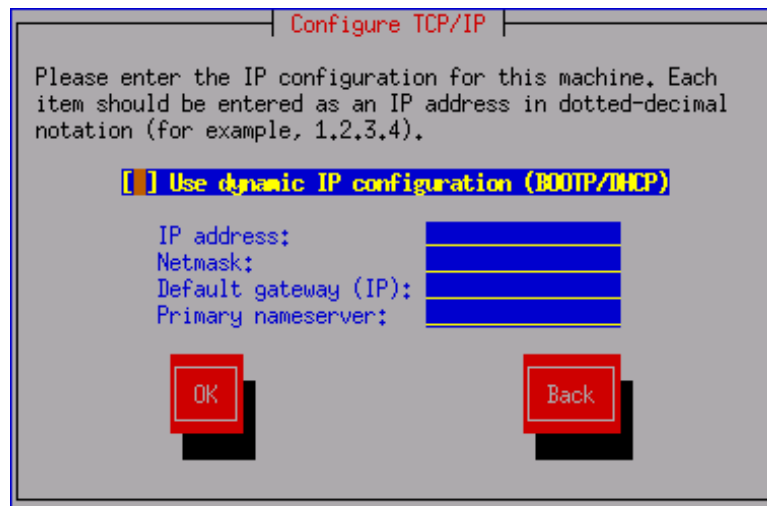
Screenshot 9: DWH64 DB Platform - PostInst.sh main menu.



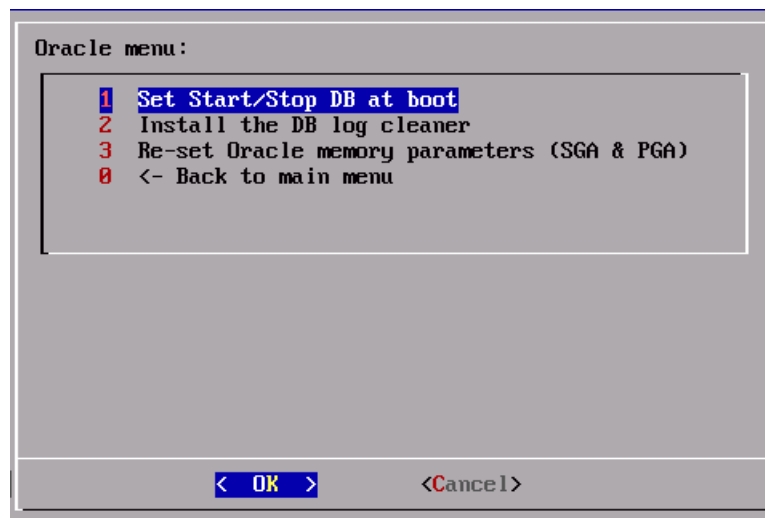
Screenshot 10: DWH32 AS Platform: PostInst.sh main menu.



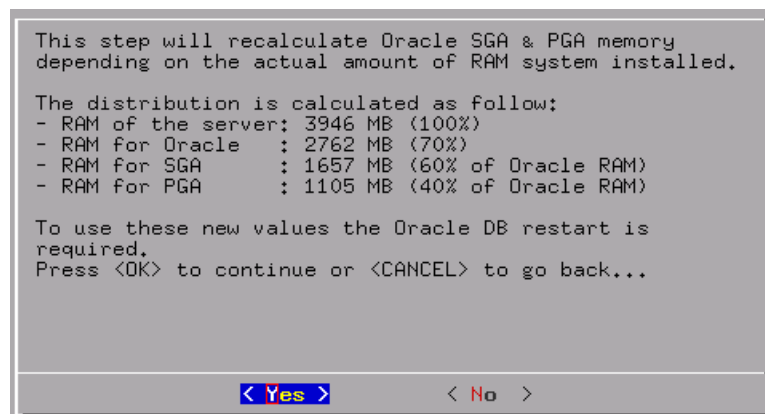
Screenshot 11: PostInst.sh: the system configuration sub-menu. Note that ethernet ports and the link on them is automatically recognized.



Screenshot 12: PostInst.sh: the ethernet configuration sub-menu.



Screenshot 13: PostInst.sh: the DWH64 DB Platform Oracle sub-menu.



Screenshot 14: PostInst.sh: the essential menu item which re-distribute the Oracle SGA and PGA memory settings present in the Oracle sub-menu.


```
This step will install HP ProLiant Support Pack (PSP).
To be activate PSP it could be required a reboot.

Do you want to continue?

< Yes > < No >
```

Screenshot 15: PostInst.sh: HP Proliant Support Pack installation step.

```
HP ProLiant Support Pack (PSP) installation starts...

The operation will take several minutes to complete,
depending on the speed of the server!

Switch to Virtual Console F12 (Alt+F12) or tail log file
"/root/PostInstall.log" to check if all the installation
steps will be completed successfully.

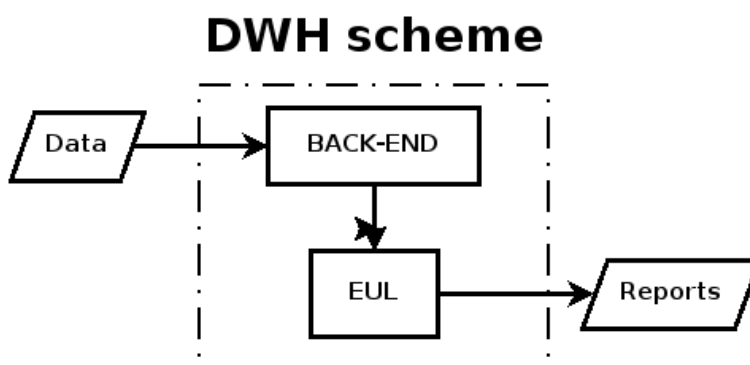
Please wait!
```

Screenshot 16: PostInst.sh: HP Proliant Support Pack installation step.

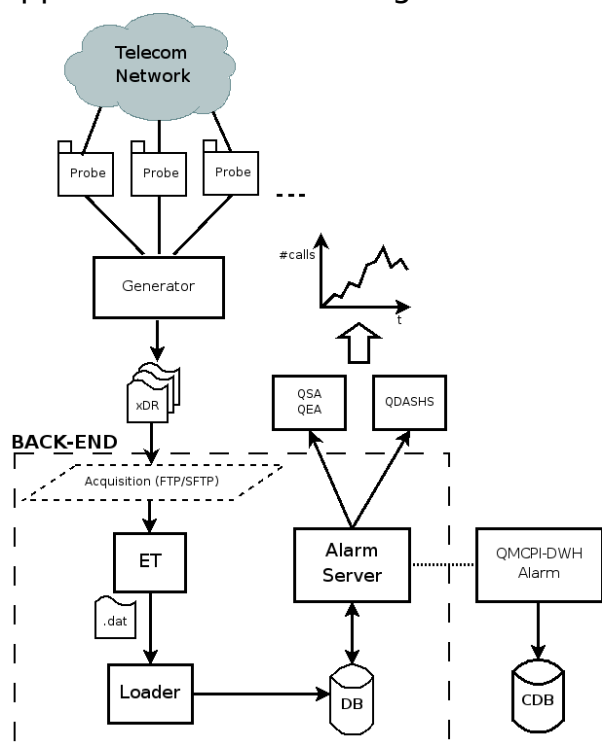
- **DWH-PTM: Data WareHouse Performance Test Manager**

The main purpose of this tool I developed is to test the many Anritsu Data WareHouses (DWH) performances on servers, in order to optimize the system overall performances and provide base indexes good to support company's sellers who can calibrate hardware furnishes on customer's needs and, at the same time, offering competitive and very high quality products.

The generic schema of a data warehouse (DWH) is represented hereinafter:



More in detail a DWH system, that the DWH-PTM have to reproduce, appears like the following:



Legend:

ET	Extraction, Transformation
CDB	Configuration DB
EUL	End-User Layer
QMCPi DWH	Plug-in for the management of the alarm server configuration files
QDASH S	Dash Server (shows alarm graphs)
QSA	Qauest7 Surveillance Application
QSA	Quest& Event Service

The DWH-PTM program is a menu-driven KSH script. The main reason of the choice of this scripting language is because:

- it can be used easily either in remote and in local servers without the need of a graphic engine, but a simple shell is sufficient
- it is simple to implement and to maintain
- it is integrable into other programs/scripts

In fact, the tool has been studied to be used as command line, for experienced users, or to be called from other applications. Anyway the most preferred way to be used is as a interactive menu driven mode.

Following screenshot shows the main menu, as soon as it is started:



As you can see some parameters are not defined, but in a standard working environment one or more of them are already set. Anyway they can be passed as parameters when the tool is called.

Pressing the “A” key in interactive mode, or passing the “-h” parameter on CLI mode, the tool shows a brief description and all parameters that it accept; following screenshot hereinafter shows it:

```
*****
***          D D D D D          W W H H          #####          ***
***          D  D  W W W H H          # # # # #          ***
***          D  D  W W W W W #####          = #####          ***
***          D  D  W W W H H          # # # # #          ***
***          D D D D D          W W H H          # # # # #          ***
*****
***          DWH Performance Test Manager - v. 1.1.0          ***
*****

Description:
"dwhtpm.sh" is a menu driven DWH Performance Test Manager that
collects CPU, RAM, SWAP, DISK and Oracle statistics for DWHs
providing CSV reports and generate system statistic graphics.
Can even be used in batch mode, as a shell command without user
menu interface, passing "-b" "-d" "-t" "-g" "-p" parameters.

Usage:
dwhtpm.sh # Full interactive menu mode
dwhtpm.sh [-q q7path] [-d dwh] [-p tpath] [-t #tdu] [-g #tgran] [-l logf]
dwhtpm.sh -b -q Q7PATH -d DWHNAME -p TRAFFICPATH -t 120 -g 5
dwhtpm.sh [-v|--version] [-h|--help]

Parameters:
-b l --batch # batch mode; do not use menu
              interface, but run the test
              directly (requires all params)
-q quest7_root_path # MQ7 environment root path
                    (not needed if QUEST7_ROOT is set)
-d dwh_name # DWH name to test
-p traffic_files_path # DWH path with traffic files
-t test_duration_in_minutes # test duration time
-g test_granularity_in_seconds # test granularity time
-l logfilename # replace default log file
-v l --version # print out the program version
-h l --help # print out this help

DWH test list available:
qipdwh qsasdwh

Examples:
dwhtpm.sh -d qipdwh -p /opt/traffic_path -t 120 -g 2
dwhtpm.sh -q /usr/quest7 -d qipdwh -t 180 -l /opt/mq/DWHPTM.log
dwhtpm.sh -b -q /usr/quest7 -d qsasdwh -p /dwhtpm/traffic -t 120 -g 5

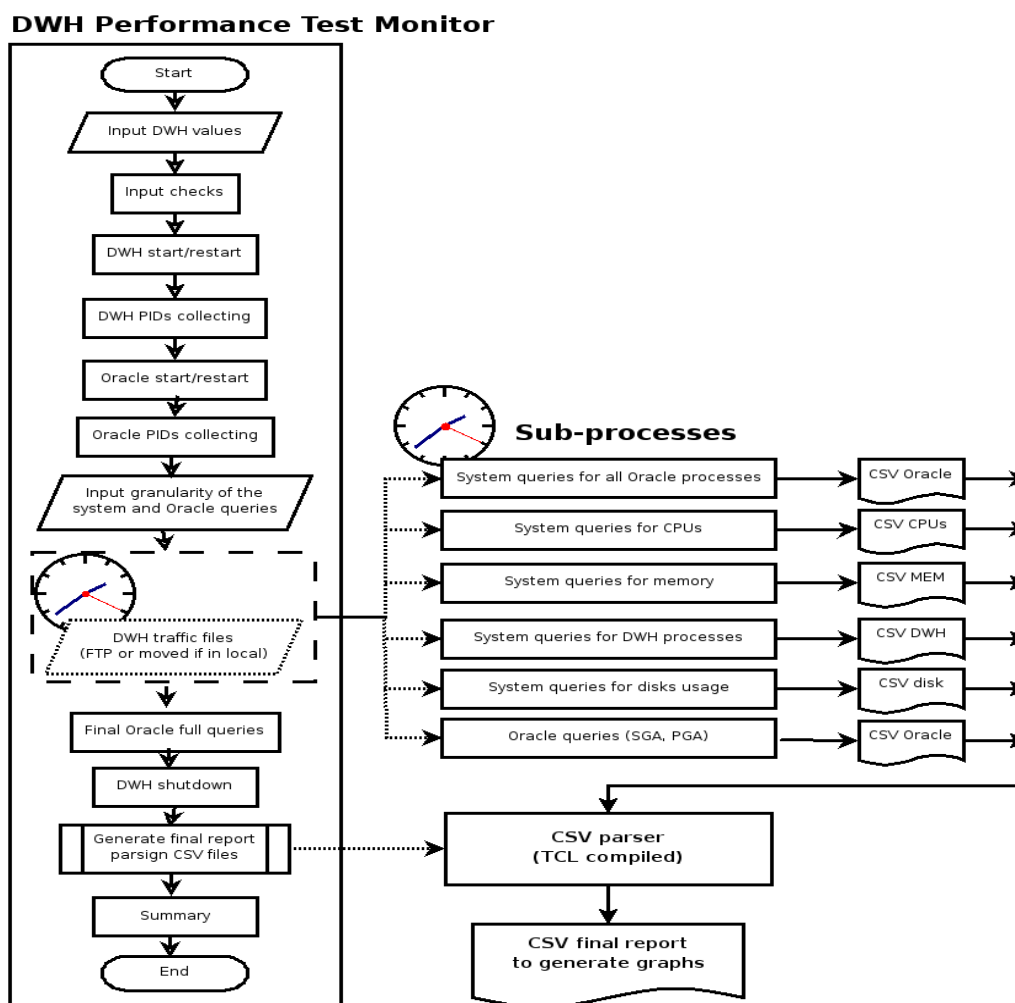
For any help or suggestion contact Emanuele.Petrucci@hetttest.com

Press ENTER to return to main menu ...
```

At the moment the DWH Performance Test Manager support just two data warehouses, but has been studied to be easy configurable simply adding the name of the new DWH and its loop ingestion time that could differ.

The DWH-PTM consist on more than one simple scripts. It has been packaged the same way all the other DWHs are been, a special tar.gz customized, with install and uninstall scripts, configuration file (.nin extension), SQL scripts to perform Oracle queries, bash script for system queries and a final Comma Separated Value (CSV) parsing TCL compiled program.

DWH-PTM requires all values to reproduce the functionality of a operational DWH system, that could be represented, very basically, with the following schema:



The program starts requesting for all parameters it needs and checks them; start (or restart) DWH to test and Oracle, collecting all their PIDs and checking if all processes are correctly started-up; request for the granularity of the system checks; loop for the desired time providing the traffic files to be ingested by the DWH and starting slave-programs which collects server system resources informations and Oracle informations, saving them in Comma Separated Value (CSV) files. When testing time passed away perform the final Oracle queries, DWH shutdown, create the final CSV file for graphics generation.

DWH-PTM uses a series of additional programs to complete its job. Those are:

- bash scripts to collect system resources data in CSV files
- SQL scripts to collect Oracle DB data in CSV files
- a TCL script compiled which parse previously collected CSV files into a bigger CSV report file which will be used by Oracle to generate graphics.

Following text saved into DWH-PTM log file can give a good idea of the steps performed by DWH-PTM and of the results it provide for the test of "qipdwh" DWH (note that the log is about not the final release, but of a previous development release):

```
petrus@Kysmor-d:/DOCS/devel$ m ./DWH_PerfTest/log/dwhptm.log_20060320_0900
20060320 09:53:57 AM - Start program dwhptm.sh V.0.6.4
=====

*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
=== 20060320 09:54:05 AM - start_dwh_perftest: Entered into procedure.
=== 20060320 09:54:05 AM - start_dwh_perftest: Check if the DWH "qipdwh" is installed
=== 20060320 09:54:05 AM - start_dwh_perftest: DWH nin file of "qipdwh" is: /usr/quest7/nin/qipdwh.nin
*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
Using traffic files path passed: ./traffic

=== 20060320 09:54:05 AM - start_dwh_perftest: Starting DWH Performance Test for "qipdwh" using date 20060320-095405
(QIPDWH) Service Manager already down
=== 20060320 09:54:06 AM - start_dwh_perftest: qipdwh shutdown done
Using jdk ORB.....
lor stored on file: /usr/quest7/packages/qipdwh-backend@III-1.5.0/lib/QIPDWH_serviceManager.ior
(QIPDWH) Service Manager running
=== 20060320 09:54:13 AM - start_dwh_perftest: qipdwh start done
*** 20060320 09:54:17 AM - start_dwh_perftest: Service Manager running for qipdwh
=== 20060320 09:54:17 AM - start_dwh_perftest: List all process actual running through a "ps -eafww" command:

=== start_dwh_perftest: PS START HERE *****
UID      PID  PPID  C  STIME TTY      TIME CMD
root     1    0  0 Mar07 ?      00:00:04 init
root     2    1  0 Mar07 ?      00:00:00 [keventd]
root     3    1  0 Mar07 ?      00:00:00 [keventd]
root     4    1  0 Mar07 ?      00:00:00 [keventd]
root     5    1  0 Mar07 ?      00:00:00 [keventd]
root     6    0  0 Mar07 ?      00:00:00 [ksoftirqd_CPU0]
root     7    0  0 Mar07 ?      00:00:00 [ksoftirqd_CPU1]
root     8    0  0 Mar07 ?      00:00:00 [ksoftirqd_CPU2]
root     9    0  0 Mar07 ?      00:00:00 [ksoftirqd_CPU3]
root    10    0  0 Mar07 ?      00:00:31 [kswapd]
root    11    0  0 Mar07 ?      00:00:00 [kreclaimd]
root    12    0  0 Mar07 ?      00:00:02 [krefilld]
root    13    0  0 Mar07 ?      00:00:00 [bdflush]
```

```
root    14   0 0 Mar07 ?    00:00:00 [kupdated]
root    15   1 0 Mar07 ?    00:00:00 [mdrecoveryd]
root    23   1 0 Mar07 ?    00:00:03 [kjournald]
root   159   1 0 Mar07 ?    00:00:00 [kjournald]
root   160   1 0 Mar07 ?    00:00:08 [kjournald]
root   161   1 0 Mar07 ?    00:00:00 [kjournald]
root   162   1 0 Mar07 ?    00:00:05 [kjournald]
root   163   1 0 Mar07 ?    00:00:16 [kjournald]
root   531   1 0 Mar07 ?    00:00:00 syslogd -m 0
root   536   1 0 Mar07 ?    00:00:00 klogd -2
rpc     556   1 0 Mar07 ?    00:00:00 portmap
rpcuser 584   1 0 Mar07 ?    00:00:00 rpc.statd
root    735   1 0 Mar07 ?    00:00:01 /usr/sbin/sshd
root    769   1 0 Mar07 ?    00:00:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid
root    801   1 0 Mar07 ?    00:00:00 gpm -t ps/2 -m /dev/mouse
root    819   1 0 Mar07 ?    00:00:00 crond
xfs     875   1 0 Mar07 ?    00:00:00 xfs -droppriv -daemon
daemon 911   1 0 Mar07 ?    00:00:00 /usr/sbin/atd
root    938   1 0 Mar07 tty1    00:00:00 /sbin/mingetty tty1
root    939   1 0 Mar07 tty2    00:00:00 /sbin/mingetty tty2
root    940   1 0 Mar07 tty3    00:00:00 /sbin/mingetty tty3
root    941   1 0 Mar07 tty4    00:00:00 /sbin/mingetty tty4
root    942   1 0 Mar07 tty5    00:00:00 /sbin/mingetty tty5
root    943   1 0 Mar07 tty6    00:00:00 /sbin/mingetty tty6
root    944   1 0 Mar07 ?    00:00:00 /usr/bin/kdm -nodaemon
root    945   1 0 Mar07 ?    00:00:00 /bin/su oracle -c exec /opt/oracle/product/RDBMS10g/bin/ocssd
root   963  944 0 Mar07 ?    00:00:00 /etc/X11/X -auth /var/lib/kdm/authfiles/A:0-2vI65P
root   972  944 0 Mar07 ?    00:00:00 -:0
root   975  972 0 Mar07 ?    00:00:00 /usr/bin/kdm_greet
oracle  982  945 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  984  982 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  985  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  986  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  987  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  988  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  989  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  990  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  991  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  993  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  994  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  995  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  996  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  997  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle  998  984 0 Mar07 ?    00:00:00 /opt/oracle/product/RDBMS10g/bin/ocssd.bin
oracle 3410   1 0 Mar08 ?    00:00:00 ora_pmon_dwh
oracle 3412   1 0 Mar08 ?    00:00:00 ora_mman_dwh
oracle 3414   1 0 Mar08 ?    00:00:17 ora_dbw0_dwh
oracle 3416   1 0 Mar08 ?    00:00:28 ora_lgwr_dwh
oracle 3418   1 0 Mar08 ?    00:00:27 ora_ckpt_dwh
oracle 3420   1 0 Mar08 ?    00:00:33 ora_smon_dwh
oracle 3422   1 0 Mar08 ?    00:00:00 ora_reco_dwh
oracle 3424   1 0 Mar08 ?    00:00:00 ora_cjq0_dwh
oracle 3426   1 0 Mar08 ?    00:00:00 ora_d000_dwh
oracle 3428   1 0 Mar08 ?    00:00:00 ora_s000_dwh
```

```
oracle 3434 1 0 Mar08 ? 00:00:00 ora_qmnc_dwh
oracle 3436 1 0 Mar08 ? 00:09:42 ora_mmon_dwh
oracle 3438 1 0 Mar08 ? 00:00:00 ora_mmmnl_dwh
oracle 3441 1 0 Mar08 ? 00:00:00 /opt/oracle/product/RDBMS10g/bin/tnslsnr LISTENER -inherit
root 10919 735 0 Mar10 ? 00:00:00 /usr/sbin/sshd
root 10920 10919 0 Mar10 pts/6 00:00:00 -ksh
petrus 12450 1 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12452 12450 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12453 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12454 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12455 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12460 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12461 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12462 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12463 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
petrus 12470 12452 0 Mar10 ? 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
oracle 12491 1 0 Mar10 ? 00:00:00 oraclerdwh (LOCAL=NO)
petrus 12492 12452 0 Mar10 ? 00:00:02 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 21012 735 0 Mar14 ? 00:00:01 /usr/sbin/sshd
root 21013 21012 0 Mar14 pts/2 00:00:00 -ksh
root 946 735 0 Mar14 ? 00:00:00 /usr/sbin/sshd
root 947 946 0 Mar14 pts/7 00:00:00 -ksh
root 4839 735 0 Mar14 ? 00:00:00 /usr/sbin/sshd
root 4840 4839 0 Mar14 pts/0 00:00:00 -ksh
root 9045 1 0 Mar14 ? 00:00:00 tail -f reports/report_20060314-164808_CPU.sar reports/report_20060314-
164808_DU.rep reports/report_20060314-164808_MEM.sar reports/report_20060314-164808_ORA.sar
oracle 20530 1 0 Mar15 ? 00:02:22 ora_j000_dwh
root 20758 947 0 Mar16 pts/7 00:00:00 tail -f stat.log qipdwh.log /usr/quest7/log/qipdwh/error.log
/usr/quest7/log/qipdwh/trace.log /usr/quest7/log/qipdwh/normal.log
root 20790 735 0 Mar16 ? 00:00:00 /usr/sbin/sshd
root 20791 20790 0 Mar16 pts/1 00:00:00 -ksh
root 25149 20791 0 Mar17 pts/1 00:00:00 tail -f log/dwhptm.log
oracle 12209 1 0 Mar18 ? 00:01:17 ora_q001_dwh
root 17949 1 0 Mar18 ? 00:00:00 tail -f log/dwhptm.log
oper 23653 769 0 09:41 ? 00:00:00 ftpd: 10.105.2.3: oper: IDLE
root 23658 735 0 09:42 ? 00:00:00 /usr/sbin/sshd
root 23659 23658 0 09:42 pts/5 00:00:00 -ksh
root 23692 23659 0 09:43 pts/5 00:00:00 vi dwhptm.sh
```



```
root 23730 21013 0 09:53 pts/2 00:00:00 /bin/ksh ./dwhptm.sh -d qipdwh -t 60 -g 1 -p ./traffic
root 24696 1 3 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24698 24696 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24699 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24700 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24701 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24706 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24707 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24708 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24709 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24716 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
root 24720 24698 0 09:54 pts/2 00:00:00 java -ms512m -mx512m -Dquest7.root=/usr/quest7 -Duser.timezone=CET
-Dapplication.name=QIPDWH -Dversion=1.0 -Djava.rmi.server.codebase=/localhost
com.nettest.masterquest.qipdwh.server.Main QIPDWH -run
=== start_dwh_perftest: PS STOPS HERE *****

Collecting "qipdwh" PIDS:
=== 20060320 09:54:17 AM - start_dwh_perftest: Uppercase of qipdwh = QIPDWH
-> DWH qipdwh Server started: 22 processes
Server PIDS = 12450 12452 12453 12454 12455 12460 12461 12462 12463 12470 12492 24696 24698 24699 24700 24701
24706 24707 24708 24709 24716 24720
(QIPDWH) Executed 'START' on ET
-> DWH qipdwh ET started: PID = 25485
(QIPDWH) Executed 'START' on Loader
-> DWH qipdwh Loader started: PID = 25853
(QIPDWH) Executed 'START' on Cleaner
-> DWH qipdwh Cleaner started: PID = 26221
Using duration test time passed: Not defined
The program adds a 10 percent more
=== start_dwh_perftest: Test duration = 72 min (including +10 percent)
=== start_dwh_perftest: Checking if Oracle is up

Oracle is up and running
*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
Start system and Oracle query data collection for "DWH_NAME":
*** start_dwh_perftest: sar ORA string = "sar -x 3410 -x 3412 -x 3414 -x 3416 -x 3418 -x 3420 -x 3422 -x 3424 -x 3426 -x 3428
-x 3434 -x 3436 -x 3438 -x 20530 -x 3441 1 4321 2>&1 >>/opt/DWH_PerfTest/report/report_20060320-095405_ORA.sar &"
-> System data collection started for ORACLE processes
*** start_dwh_perftest: sar CPU string = "sar -U ALL 1 4321 2>&1 >>/opt/DWH_PerfTest/report/report_20060320-
095405_CPU.sar &"
```

```
-> System data collection started for CPU(s)
*** start_dwh_perftest: sar MEM string = "sar -r 1 4321 2>&1 >>/opt/DWH_PerfTest/report/report_20060320-095405_MEM.sar
&"
-> System data collection started for MEMORY usage (RAM + SWAP)
*** start_dwh_perftest: sar DWH string = "sar -x 12450 -x 12452 -x 12453 -x 12454 -x 12455 -x 12460 -x 12461 -x 12462 -x
12463 -x 12470 -x 12492 -x 24696 -x 24698 -x 24699 -x 24700 -x 24701 -x 24706 -x 24707 -x 24708 -x 24709 -x 24716 -x
24720 -x 25485 -x 25853 -x 26221 1 4321 2>&1 >>/opt/DWH_PerfTest/report/report_20060320-095405_DWH.sar &"
-> System data collection started for DWH processes
*** start_dwh_perftest: disk usage string = "/opt/DWH_PerfTest/bin/query_du -t 72 -l 5
>/opt/DWH_PerfTest/report/report_20060320-095405_DU.rep &"
-> System data collection started for Disk Usage processes
File nin: /usr/quest7/nin/qipdwh.nin
*** start_dwh_perftest: Oracle connection string: "qipdwh/qipdwh"
-> Oracle query collection started for PGA
-> Oracle query collection started for SGA
-> Oracle query collection started for SGA1

=== 20060320 09:54:56 AM - start_dwh_perftest:
List of all processes started by the program and actually in running state:
=== PS LIST START HERE *****
root 26365 23730 0 09:54 pts/2 00:00:00 /bin/ksh ./dwhptm.sh -d qipdwh -t 60 -g 1 -p ./traffic
=== PS LIST STOPS HERE *****

=== 20060320 09:55:20 AM - start_dwh_perftest: ./traffic/IP-TDR-2006.03.18.1400-2006.03.18.1415 has 520833 data lines
=== 20060320 09:55:20 AM - start_dwh_perftest: Moving ./traffic/IP-TDR-2006.03.18.1400-2006.03.18.1415 into ingestion
dir /usr/quest7/data/qipdwh/etl/ingestion_buffer
=== 20060320 10:10:23 AM - start_dwh_perftest: ./traffic/IP-TDR-2006.03.18.1415-2006.03.18.1430 has 520833 data lines
=== 20060320 10:10:23 AM - start_dwh_perftest: Moving ./traffic/IP-TDR-2006.03.18.1415-2006.03.18.1430 into ingestion
dir /usr/quest7/data/qipdwh/etl/ingestion_buffer
=== 20060320 10:25:26 AM - start_dwh_perftest: ./traffic/IP-TDR-2006.03.18.1430-2006.03.18.1445 has 520833 data lines
=== 20060320 10:25:26 AM - start_dwh_perftest: Moving ./traffic/IP-TDR-2006.03.18.1430-2006.03.18.1445 into ingestion
dir /usr/quest7/data/qipdwh/etl/ingestion_buffer
=== 20060320 10:40:28 AM - start_dwh_perftest: ./traffic/IP-TDR-2006.03.18.1445-2006.03.18.1500 has 520833 data lines
=== 20060320 10:40:28 AM - start_dwh_perftest: Moving ./traffic/IP-TDR-2006.03.18.1445-2006.03.18.1500 into ingestion
dir /usr/quest7/data/qipdwh/etl/ingestion_buffer
=== 20060320 10:55:31 AM - start_dwh_perftest: ./traffic/IP-TDR-2006.03.18.1500-2006.03.18.1515 has 520833 data lines
=== 20060320 10:55:31 AM - start_dwh_perftest: Moving ./traffic/IP-TDR-2006.03.18.1500-2006.03.18.1515 into ingestion
dir /usr/quest7/data/qipdwh/etl/ingestion_buffer
=== 20060320 11:08:31 AM - start_dwh_perftest: Traffic file "" is still being ingested
=== 20060320 11:08:31 AM - start_dwh_perftest:
* "qipdwh" after 72 min finished processing ingested files
*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
Start final query data collection for "qipdwh":
=== 20060320 11:08:42 AM - start_dwh_perftest: Starting Oracle query collection for Materialized View rows
=== 20060320 11:09:44 AM - start_dwh_perftest: Starting Oracle query collection for Table Space size
=== 20060320 11:10:44 AM - start_dwh_perftest: Starting Oracle query collection for Table Space system
=== 20060320 11:11:45 AM - start_dwh_perftest:
"qipdwh" test finished. === 20060320 11:11:55 AM - generate_reports: Start generation reports
*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
=== 20060320 11:11:55 AM - generate_reports: Start parsing DU SAR report file: /opt/DWH_PerfTest/report/report_20060320-
095405_DU.rep
=== 20060320 11:11:55 AM - generate_reports: Start parsing CPU SAR report file:
/opt/DWH_PerfTest/report/report_20060320-095405_CPU.sar
=== 20060320 11:12:24 AM - generate_reports: Start parsing MEM SAR report file:
/opt/DWH_PerfTest/report/report_20060320-095405_MEM.sar
=== 20060320 11:12:32 AM - generate_reports: Start parsing DWH SAR report file:
/opt/DWH_PerfTest/report/report_20060320-095405_DWH.sar
```

```
=== 20060320 11:15:28 AM - generate_reports: Start parsing Oracle SAR report file:
/opt/DWH_PerfTest/report/report_20060320-095405_ORA.sar
=== Passing Oracle pids association string: " ora_pmon ora_pmon ora_mman ora_mman ora_dbw0 ora_dbw0 ora_lgwr
ora_lgwr ora_ckpt ora_ckpt ora_smon ora_smon ora_reco ora_reco ora_cjq0 ora_cjq0 ora_d000 ora_d000 ora_s000 ora_s000
ora_qmnc ora_qmnc ora_mmon or_mmon ora_mml ora_mml ora_j000 ora_j000 ora_q000 ora_q000 ora_lsnr ora_lsnr"
=== 20060320 11:17:21 AM - generate_reports: Start parsing SQL report file: /opt/DWH_PerfTest/report/report_20060320-
095405_SQL_PGA.rep
=== 20060320 11:17:21 AM - generate_reports: Start parsing SQL report file: /opt/DWH_PerfTest/report/report_20060320-
095405_SQL_SGA.rep
=== 20060320 11:17:21 AM - generate_reports: Start parsing SQL report file: /opt/DWH_PerfTest/report/report_20060320-
095405_SQL_SGA1.rep
=== 20060320 11:17:21 AM - generate_reports: Start parsing SQL report file: /opt/DWH_PerfTest/report/report_20060320-
095405_SQL_MVR.rep
=== 20060320 11:17:21 AM - generate_reports: Start parsing SQL report file: /opt/DWH_PerfTest/report/report_20060320-
095405_SQL_TSZ.rep
=== 20060320 11:17:21 AM - generate_reports: Start parsing SQL report file: /opt/DWH_PerfTest/report/report_20060320-
095405_SQL_TSY.rep
=== 20060320 11:17:21 AM - generate_reports:"qipdwh" generation report finished.

*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
SUMMARY
=====
Test duration      : 72 minutes
Additional ingesting time : 0 minutes
Total test duration   : 72 minutes
Traffic files ingested  : 5
Esteemated files/day   : 100
Total data lines ingested : 2604165
Esteemated data lines/day : 52083300

Press ENTER to return to main menu ...=== 20060320 11:18:42 AM - generate_reports: Exiting from procedure

=== 20060320 11:18:42 AM - start_dwh_perfctest: Exiting from procedure

*** Terminal: "rxvt terminal emulator (X Window System)": col=80, lines=24, w=63, spc=8
petrus@Kysmor-d:/DOCS/devel$
```

2005

**Intranet System
Administrator**

**TIM Spa
Rome, IT**

Hardware technology used:

- ✓ Two HP SuperDome clustered
- ✓ HP XP 1024 disk array with redundant controller and hardware RAID
- ✓ Four HP Proliant with SCSI disks, 2GB RAM and #4 Intel Xeon CPUs each (one used as backup server)

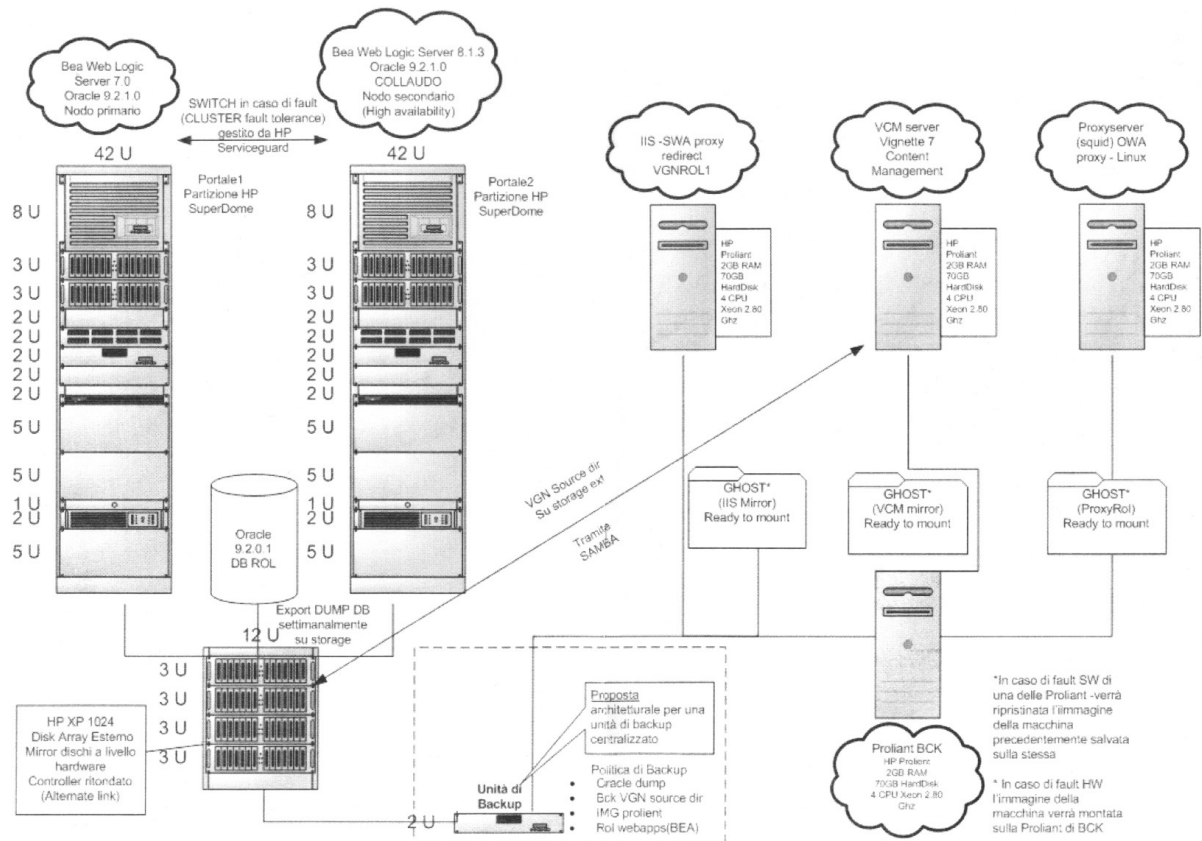
Software technology used:

- ✓ BEA WebLogic Server v.8.1
- ✓ BEA Administration Portal v.8.1
- ✓ BEA Workshop v.8.1
- ✓ Vignette v.7.2
- ✓ Oracle DB 9i
- ✓ MySQL v.3.23

TIM is Italy's largest supplier of cellular telecommunications products and services and was a wholly owned subsidiary of Telecom Italia until it was folded back into its parent company. Contracted to manage TIM's intranet during a relatively short transition phase.

My responsibilities were to manage the whole intranet using BEA for the development, administration and maintenance of the BEA nodes and of the dynamic portals managed with portlet, and Vignette for the dynamic content publication.

The schema hereinafter illustrate the system used to manage the entire intranet.



2005

**Portal System
Administrator**

**Telecom Italia Spa-
Rome, IT**

Hardware technology used:

- ✓ Two Sun Cluster Platform 15000

Software technology used:

- ✓ BEA WebLogic Server v.7
- ✓ BEA Administration Portal v.7
- ✓ Oracle DB 9i
- ✓ Apache v.1.3

Telecom is Italy's largest provider of telecommunications products and services. Contracted to manage and administrate Telecom's web portals, Rosso Alice Italia (see www.rossoalice.it) and Rosso Alice Francia (see www.aliceadsl.fr). These web portals provide news, customer and product information, games, e-commerce, and streaming audio/video (e.g., movies, soccer, music).

- Based on a very high-end multiprocessor hardware platform (Sun Cluster Platform 15k's) with several million page hits per day and hundreds of thousands of unique users.
- Developed a complete set of scripts which automated the process of portal administration thereby saving significant employee time and company resources.

Emanuele Petrucci

<http://kysmor.com>

Work Portfolio

kysmor@kysmor.com

**2001 -
2004**

**Team Leader &
Spacecraft Engineer /
Administrator**

**European Space
Agency (ESA)
Rome, IT**



Alcatel Space was the main contractor for the European Space Agency “Envisat-1” project, an 8-ton polar satellite equipped with 10 instruments for the study & analysis of the Earth's environment (see www.envisat.esa.int), the most complex and complete never built of its category. This highly acclaimed satellite continues to provide environmental data to agencies, universities, and research centers throughout the world.

Hardware technology used:

- ✓ IBM SP2 clusters with dual PPC nodes (thin & wide nodes)
- ✓ IBM RS/6000 H70, B80, B50, R50
- ✓ IBM P630, P670, P690
- ✓ IBM 43P with AIX o.s.
- ✓ IBM PC NetVista & X-Series X205, X330, X335
- ✓ DELL PowerEdge servers
- ✓ HP Proliant servers
- ✓ DIGITAL Alpha Station 600 with PPC

Network technology used:

- ✓ CISCO routers series 7200/4000/3600/2500
- ✓ CISCO switches Catalyst series 3500/2900 XL
- ✓ FORE switches ATM ASX-1000/ASX-200BX

Storage technology used:

- ✓ IBM NTP Tape Libraries 3494
- ✓ IBM Magstar Tape Drives 3590
- ✓ IBM SSA Disk Drawers 7133-020 / 7133-D40
- ✓ SONY D1 Digital Recorder DIR-1000M
- ✓ NAS storage systems
- ✓ CIPRICO Disk Arrays
- ✓ Barracuda Disk Arrays
- ✓ Storage Concepts Disk Arrays
- ✓ Young Minds CD Writer
- ✓ DLT 7000/8000 (IBM, Compaq, HP)
- ✓ Exabyte 5GB / 20GB
- ✓ DAT 2GB/4GB

Software technology used:

- ✓ Oracle DB v.7.3.4
- ✓ Several GPL system/network monitoring tools

Operating system used:

- ✓ IBM AIX v.4.3 & v.5 on all IBM hardware
- ✓ IBM OS/2 on the IBM NTP Tape Library 3494
- ✓ Windows 2000 and 2000 Server on IBM PC NetVista & X-series
- ✓ Windows NT4 on DEC Alpha 600
- ✓ Red Hat Linux 7.2 on the NAS

Miscellaneous hardware components:

- ✓ Rimage Autostar II robot (CD/DVD burner and labeller)

- ✓ HP laserjet printers (various models)
- ✓ Lexmark OpraColor 1200 color laser printer (A3 format)
- ✓ Tektronix X-terminal
- ✓ IBM text terminal 3153
- ✓ Datum Time Code Generator/Translator (Time Server)
- ✓ Datum Switch & Distribution
- ✓ Novotronic Base Band Matrix
- ✓ MacDonald-Dettwiler RFC Direct Archive System
- ✓ ATM Media Conversion Center
- ✓ Zebra label printer
- ✓ Several Router / Switch / Hub (Cisco, CentreCom, Fore)
- ✓ Miscellaneous spare parts (workstation & antenna H/W)

Platforms managed and relative hardware:

- ESF-HSE Platform (ESA-Esrin - Rome-IT)
 - #21 workstations
 - #43 dual PPC nodes of the IBM SP2 cluster (thin & wide nodes)
 - #1 Rimage Autostar II robot
- ESF-PDC Platform (ESA-Esrin - Rome-IT)
 - #10 workstations
- PDHSE Platform (ESA-Esrin - Rome-IT)
 - #12 workstations
 - #32 dual PPC nodes of the IBM SP2 cluster (thin & wide nodes)
 - #1 Rimage Autostar II robot
- PDCC Platform (ESA-Esrin - Rome-IT)
 - #16 workstations
- GSOV Platform (ESA-Esrin - Rome-IT)
 - #5 workstations
- STAND-ALONE Platform (ESA-Esrin - Rome-IT)
 - #7 workstations
- Telespazio Platform (Telespazio - Matera-IT)
 - #5 workstations
- PDHSK Platform (ESA-Kiruna - Sweden)
 - #20 workstations
 - #42 dual PPC nodes of the IBM SP2 cluster (thin & wide nodes)
 - #2 Rimage Autostar II robot
- PDHCC Platform (ESA-Kiruna - Sweden)
 - #15 workstations
- LRAC Platform (ESA-Kiruna - Sweden)
 - #20 workstations
 - #11 dual PPC nodes of the IBM SP2 cluster (thin & wide nodes)
- E-PAC Platform (ESA-Canary Islands - Spain)
 - #8 workstations
 - #9 dual PPC nodes of the IBM SP2 cluster (thin & wide nodes)

Summary of the operating hardware managed (except spares):

Total # workstations:	139
Total # IBM SP2 nodes:	137
Total # Rimage Autostar II robots:	4
Total:	280

Following table provides the identikit of Envisat-1 satellite.

Type	Earth observation ENV ironmental SAT ellite
Purpose	To gather data to study and monitor the Earth and its environment
Satellite	users the European Polar Platform as satellite bus
Specs	Launch mass: 8.2 Tons, of which Instruments 2.2 Tons Power: 6.6 KW (End of life) of which Instruments 2.0 KW Dimensions: Deployed in orbit (meters): 25 x 10 x 7 Launch Configuration (main body, in meters): 10 x 4 x 4
Lifetime	At least 5 years in orbit
Orbit	Near polar sun-synchronous with a mean altitude of 800 Km and 35 days repeat cycle
Data	On board recording capability 160 Gbit in total Ground Links at 100 Mbit/sec direct to stations or relayed via the European Data Relay Satellite (ARTEMIS)
Launch	March 1st, 2002, by Ariane 5 from Europe's Spaceport at Kourou, French Guyana



Flight module alignment test



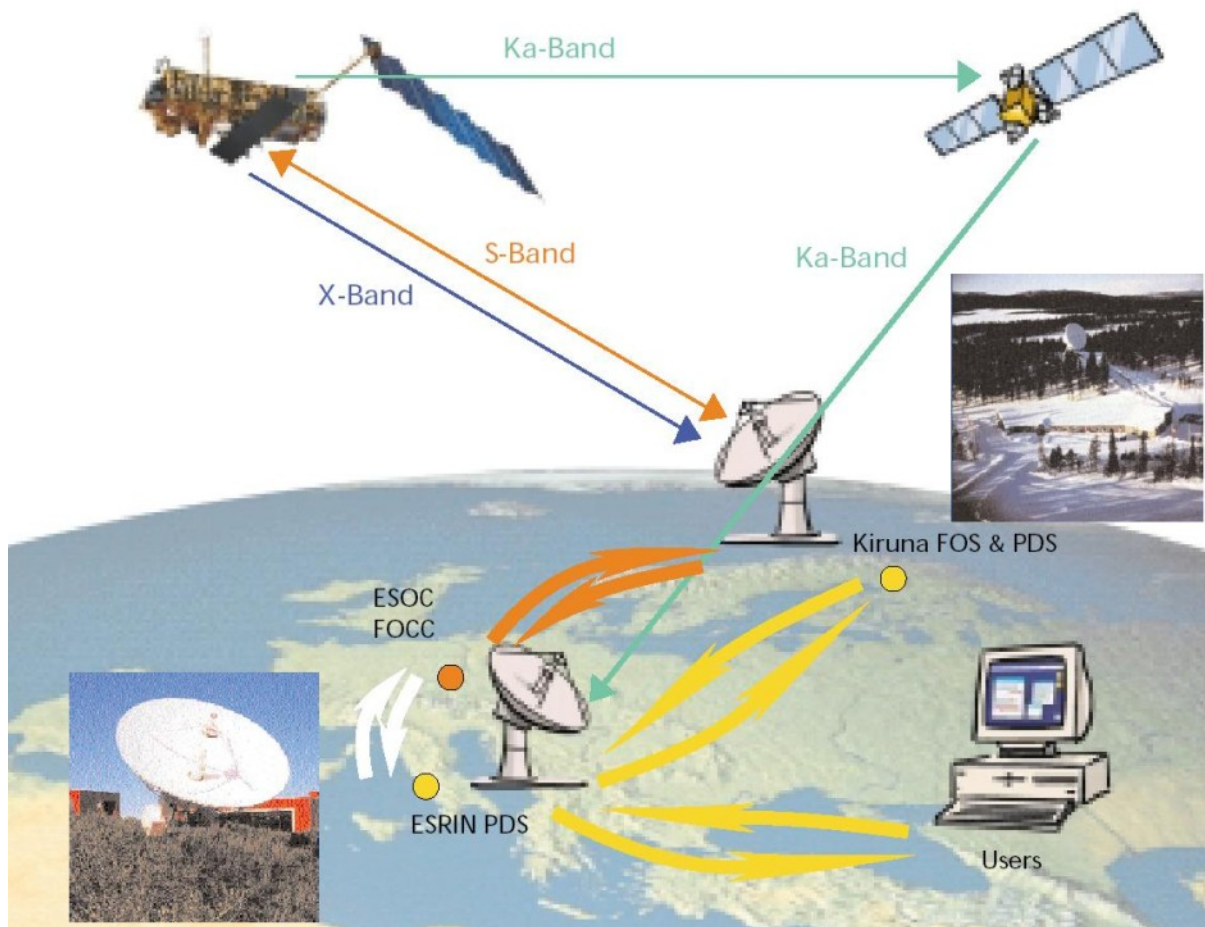
Solar array deployed

Still now, Envisat-1 is the larger, most complex and most complete environment satellite ever built. Its big power and complexity reside in the amount of instruments it has aboard and the multitude of different measurements they can operate.

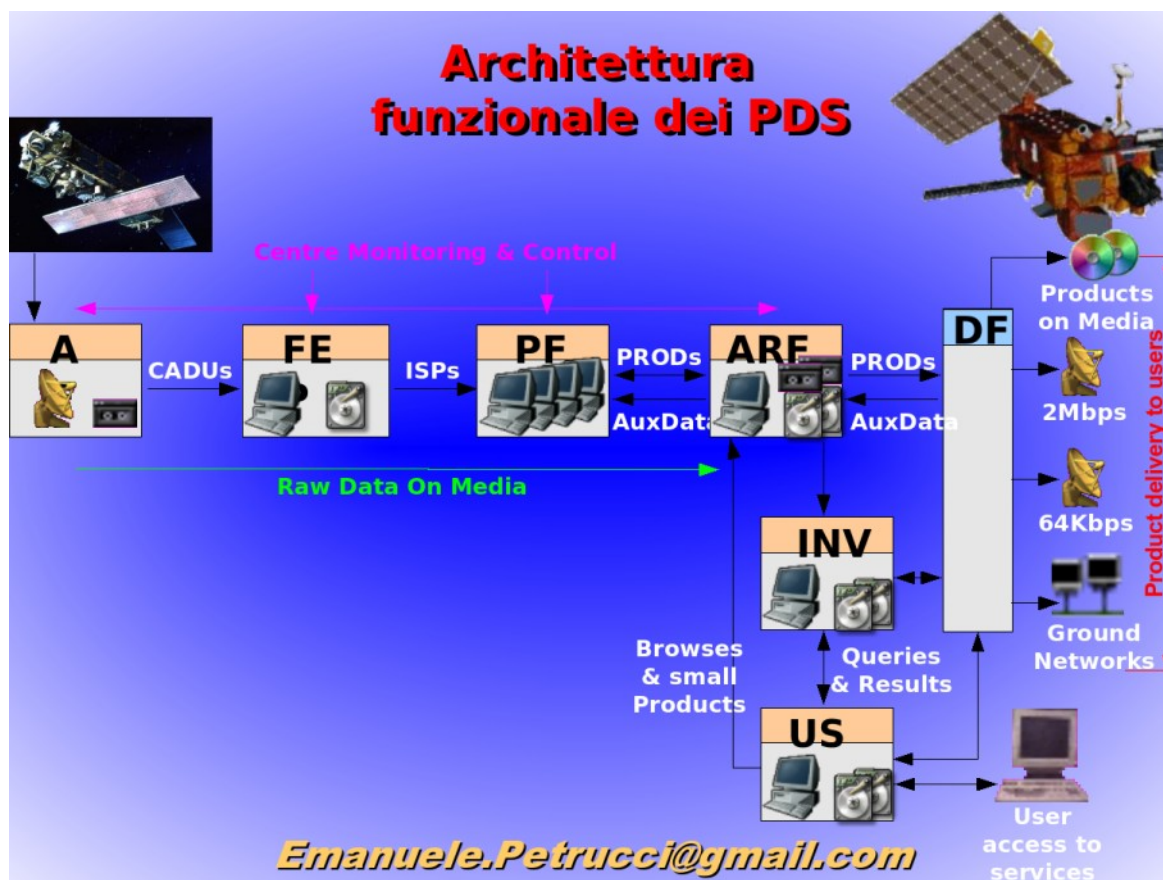
Following illustrations can give the idea.

	ASAR	Advanced Synthetic Aperture Radar
	MERIS	Medium Resolution Imaging Spectrometer
	AATSR	Advanced Along Track Scanning Radiometer
	RA-2	Radar Altimeter 2
	MWR	Microwave Radiometer
	DORIS	Doppler Orbitography and Radiopositioning integrated by Satellite
	LRR	Laser Retro Reflector
	MIPAS	Michelson Interferometer for Passive Atmospheric Sounding
	GOMOS	Global Ozone Monitoring by Occultation of Stars
	SCIAMACHY	Scanning Imaging Absorption spectroMeter for Atmospheric CHartography

Disciplines	Instruments									
	ASAR	GOMOS	RA-2	MERIS	MIPAS	MWR	LR	SCIAMACHY	DORIS	AATSR
Atmosphere										
Clouds				■				■		■
Humidity						■				
Radiative Fluxes				■						■
Temperature					■			■		
Trace Gases		■			■			■		
Aerosols		■		■	■			■		■
Land										
Surface Temperature										■
Vegetation Characteristics	■			■						■
Surface Elevation	■		■				■		■	
Ocean										
Ocean Colour				■						
Sea Surface Temperature										■
Surface Topography			■				■			
Turbidity				■						
Wave Characteristics	■		■							
Marine Geoid			■							
Ice										
Extent	■			■						■
Snow Cover	■			■						■
Topography	■		■				■			
Temperature										■



Envisat-1 operating system illustration.



As showed in the schema above, the PDS' operational architecture, the complete chain start, of course, from the satellite acquired data received from the antenna (or sent from Kiruna main receiving center recorded as raw data on Sony D1 tape) and have a first processing step by the Front End Processor (FEP). Those are DEC Alpha servers with MS Windows NT 4 with a mirrored main SCSI disk (RAID 1) and Storage Concepts disk arrays on SCSI chain, on first instance, after renewed with Ciprico 6500/6900 disk arrays with SCSI III disks set in RAID 5.

- **Prior to launch worked as Spacecraft AIT Engineer involved in all AIT phases (assembly, integration, and test) of the PDS (payload data stations) in Italy and Sweden.**

These were some of the responsibilities:

- ✓ PDS software integration
- ✓ Facility installation, configuration, testing and ultimate delivery
- ✓ PDS run-tests
- ✓ Preparation and validation of operational procedures

- **Post launch worked as system, network, and hardware administrators for all four payload data centers (i.e., 2 Italy, 1 Sweden, 1 Spain).**

Since the launch of the satellite on March 1st, 2002 ("Ariane 5" launcher - Kourou, French Guyana):

Team Leader of a team of 7 people.

After having trained them for the initial period, was responsible for their work as Operation System Administrators on the operating platforms PDHS-E and PDCC at ESA-ESRIN.

System Administrator for both Operating Platforms (production) and Maintenance Platforms (testing and integration) in the following locations:

- ✓ ESA-ESRIN (Rome, Italy), platforms: ESF, PDHS-E, PDCC, GSOV, STAND-ALONE
- ✓ ESA-KIRUNA (Kiruna, Sweden), platforms: PDHS-K, LRAC, LRCC
- ✓ E-PAC (Spanish Payload Acquisition Center: remote managing), platform: EPAC
- ✓ I-PAC/TeleSpazio (Italian Payload Acquisition Center), facilities: AF, CDRF

In these locations I managed, remotely and locally, workstations and mainframes:

- ✓ IBM PSSP with PPC nodes, RS/6000 H70, B80, B50, R50, P630, P670, P690, 43P with AIX o.s.
- ✓ IBM NTP Tape Library 3494 with OS/2 o.s.
- ✓ IBM servers PC NetVista and X-Series X205, X330, X335 with WNT/W2K
- ✓ HP ProLiant and DELL PowerEdge servers

Network Administrator of both Operating and Maintenance platforms based on CISCO routers series 7200/4000/3600/2500, CISCO switches Catalyst series 3500/2900 XL and FORE switches ATM ASX-1000/ASX-200BX.

Hardware Administrator of whole project, from the workstations to the IBM Magstars, from the x-terminals to the impulse generators, from each every antennas' components to the Spare parts, either in the operation platforms and in the storage rooms.

CVS Administrator of all generation & installation documentation and configuration files.

Carried out 5 operating missions to the ESA-KIRUNA base, Sweden, for the installation and configuration of workstations and facilities.

Personal Projects

- **SRMON: System Resource Monitor**

(see <http://sourceforge.net/projects/srmon>
or <http://srmon.kysmor.com> for the live static demo)

System Resource Monitor is a GPL program to monitor local and remote workstations, easy to use, highly configurable, totally manageable through a browser even remotely, eat low system resources, providing multiple server monitoring at a glance in one browser page (SRMON Network Manager), or showing system resources of every server with more details. I conceived it to provide an easy, but powerful and complete tool to manage a whole data center in a browser page.

The idea borned from job needs. I was asked to develop a simple tool to scan system resources and save data in CSV format to import in a spreadsheet and realize stat graphs. But my head was already ahead of the request: I thought to provide directly the requested stat graphs avoiding a double passage. So I let my creativity fly away and I conceived a bespoke system which retrieve the servers' stats, store them, create graphs and make them available to be analysed directly through the user's favorite browser, regardless the OS used!

That's why I decided to realize SRMON. The program is an Open Source project (GPL), hosted by SourceForge and currently under a deep development phases, written in C/C++.

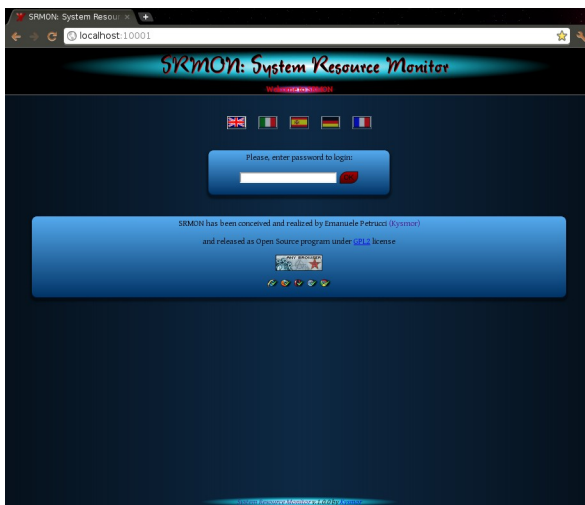
Here is a list of SRMON main features:

- runs as daemon
- integrates a tailored high-performance multi-concurrent web server
- monitoring of local/remote workstations
- 1/1 server monitoring at a glance - one server in each browser tab
- *n/1 server monitoring at a glance - all workstation in the network in one browser tab (SRMON Network Manager)
- remote installation/auto-replication (it clones itself as viruses do!) on one or more workstations with a single operation
- different graph views - time or resource based
- login with encrypted password
- http/https connection ports number customizable
- alarms - raise a pop-up, send a mail or run a program when configured events happen

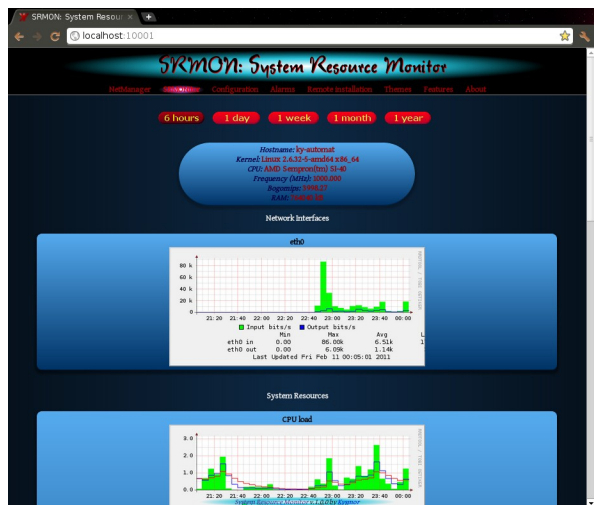
- easy configuration through the browser
- themes
- *multiple languages support
- *Mozilla Firefox add-on (planned)
- *Xscreensaver plug-in (planned)
- *Nagios plug-in (planned)

(*) Starred features are planned in future releases.

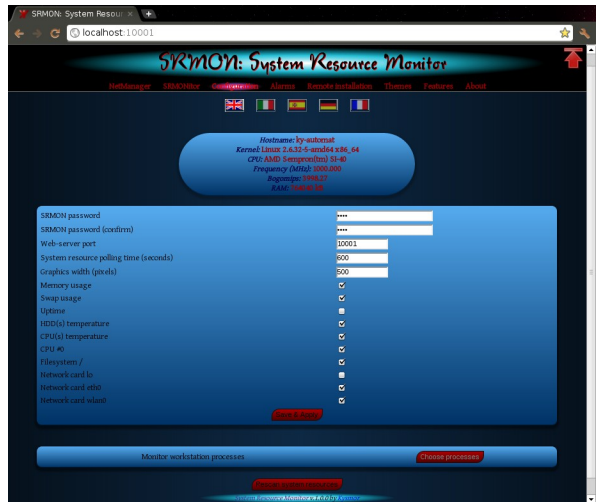
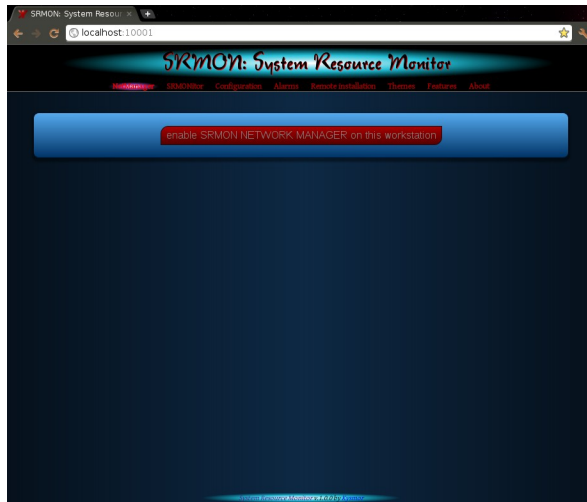
Hereinafter some SRMON screen shots:



Login/welcome page with encrypted password

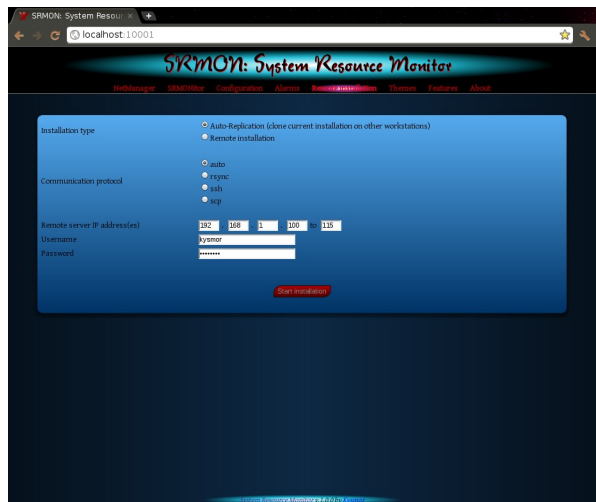
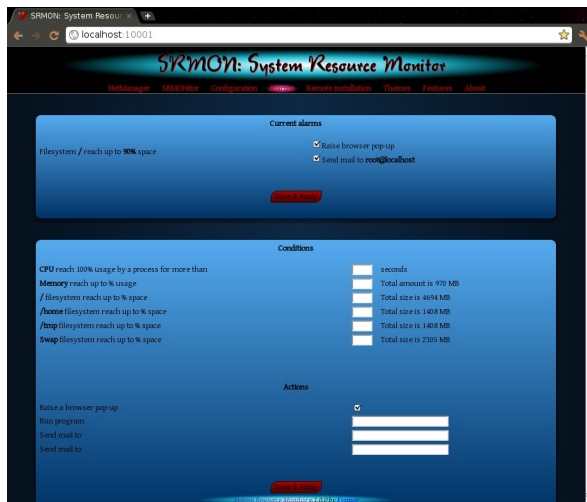


SRMON showing 6 hours graphs of a workstation on a single web page



SRMON Network Manager can run on any workstation: when enabled pressing the button it starts searching all the SRMON instances installed in the network and after having established the connection gathers the workstations resources' graphs (still to implement)

The configuration page. Note the workstation's specs showed at the top box (hostname, kernel, CPU, frequency, RAM) and all the CPU cores, filesystems and network cards identified in the middle box.



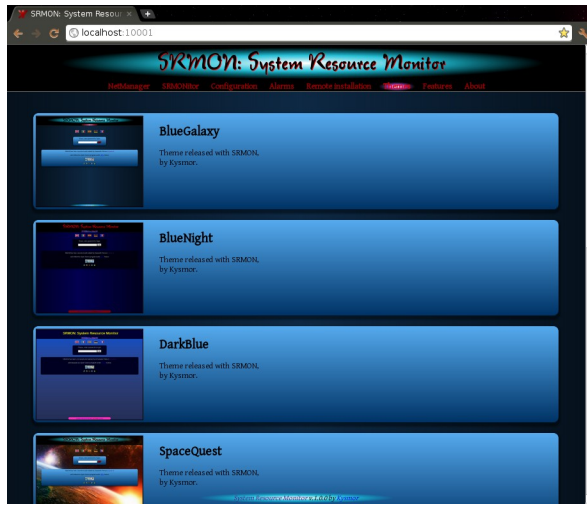
The alarms page: raise a pop-up, Remote installations performed in few seconds when configured events happen

Emanuele Petrucci

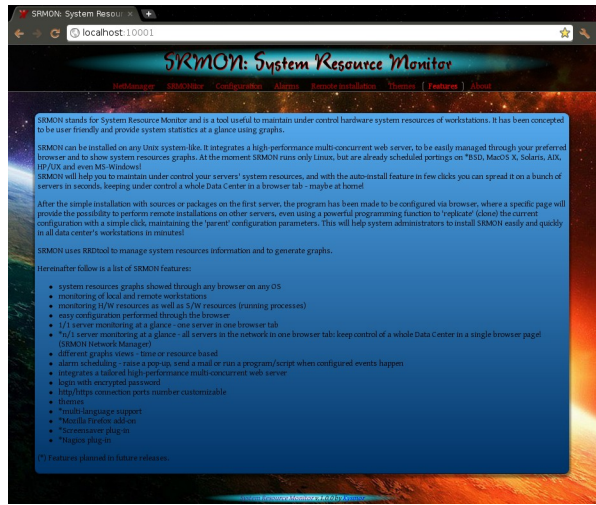
Work Portfolio

<http://kysmor.com>

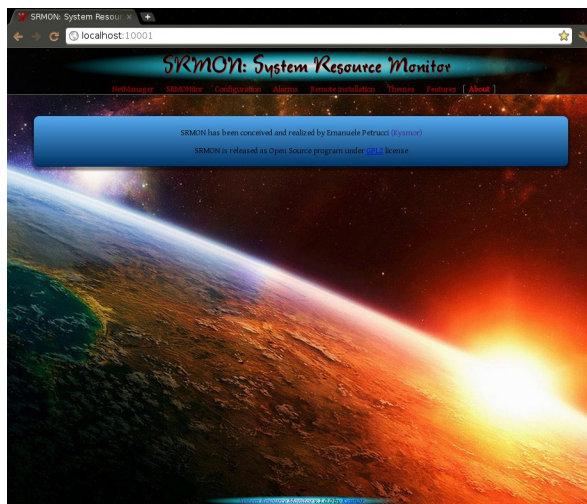
kysmor@kysmor.com



Themes (skins) applicable on-the-fly



SRMON features with with a different theme: "SpaceQuest"



The about page with the "SpaceQuest" theme

- **FlaCC: Flash Card Copier**

Design and implementation of an embedded system for duplicating proprietary PCMCIA memory cards used in industrial robotics facilities.

The program has been developed in C for low level I/O and TCL/TK for all control functions and Graphics User Interface (GUI).

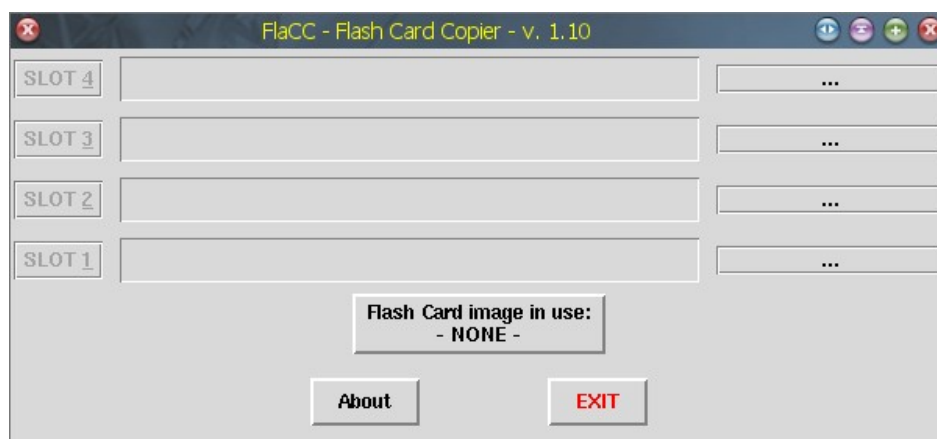
The system is a customized Debian based OS withing a special 4-slots PCMCIA card reader/writer.

Briefly, the program recognize when the operator insert PCMCIA memory cards into any slot enabling the whole line of the properly slot number. The rightest field (see screenshot below), visible with “...” when slot is empty, change status and appears a “read” button. When pressed the system disable from reading all others slots, and when the reading is finished and the PCMCIA memory image is ready, in all other active slots appears the “write” button.

This way the program allow to write contemporary all 4 slots. The central field is the progress bars which shows not a esteemed progress, but the real.

It is possible to load, save and remove any PCMCIA memory image through a button that indicate always memory image is being used.

Following screenshot display the program which recognize that no PCMCIA memory card are inserted into any slot.

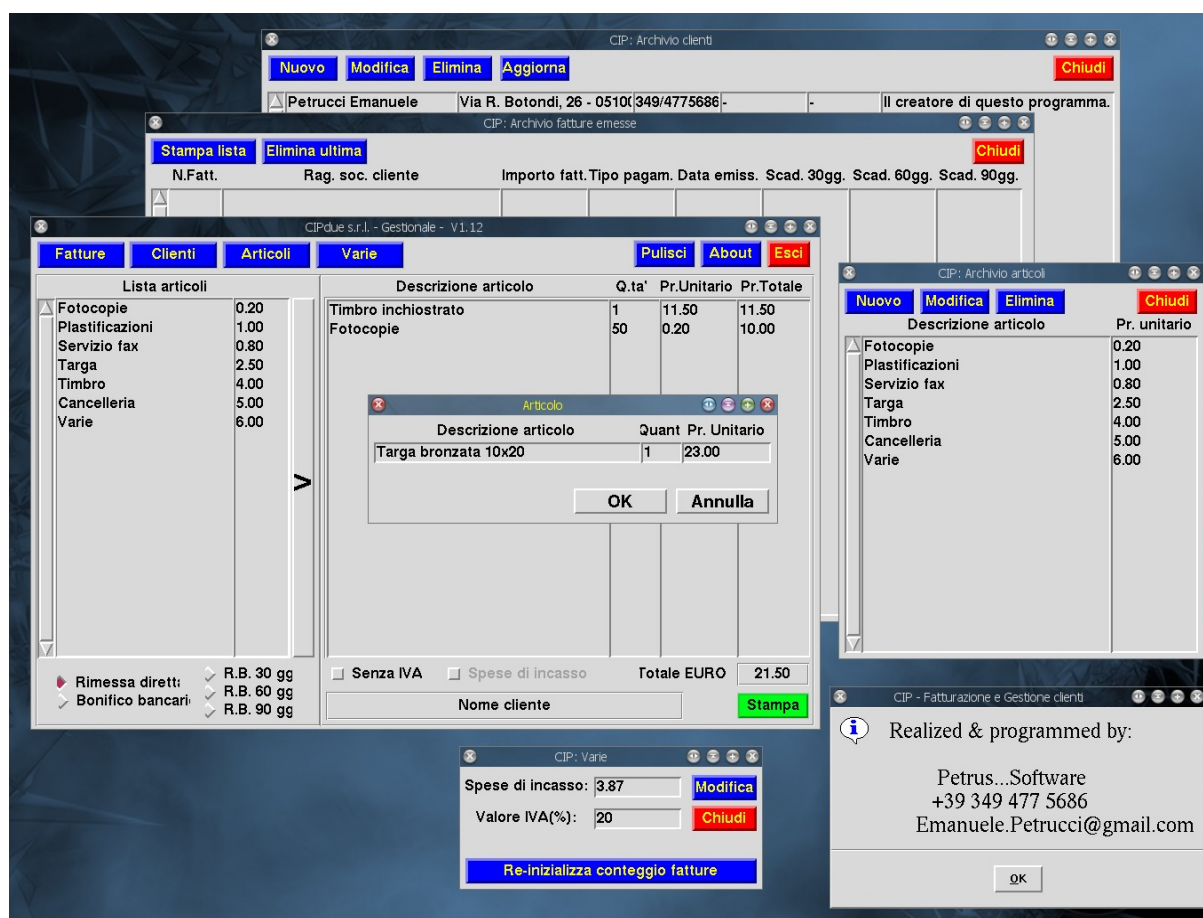


- CIP

Design and implementation of a bespoke retail Point-Of-Sale system (POS) called "CIP".

The program was developed in TCL/TK and compiled.

This screenshot shows the different program windows and components:



Hereinafter is reported a little snip of code, the function that retrieve and show the user all the details about a chosen customer's invoice, and if requested, print it to laser printer.

```
#####
# Procedura .mostraDettagliFattura
#
proc vTclWindow.mostraDettagliFattura {base} {
    global CIP_PATH;
    global BACKUP_PATH;
    global FONT_BUTTONS;
    global FONT_LABELS;
    global FONT_TEXTS;
    global numfatt;
    if {$base == ""} {
        set base .mostraDettagliFattura
    }
    if {[winfo exists $base]} {
        wm deiconify $base; return
    }
    #####
    # CREATING WIDGETS
    #####
    toplevel $base -class Toplevel
    wm focusmodel $base passive
    wm geometry $base 578x505+120+190
    wm maxsize $base 1024 768
    wm minsize $base 1 1
    wm overriddenirect $base 0
    wm resizable $base 0 0
    wm deiconify $base
    wm title $base "CIP: Dettaglio fattura"

    entry $base.ent_numfatt -font $FONT_TEXTS
    entry $base.ent_ragsoc -font $FONT_TEXTS
    label $base.lab_numfatt -borderwidth 1 -font $FONT_LABELS -text {N. Fattura}
    label $base.lab_ragsoc -borderwidth 1 -font $FONT_LABELS -text {Ragione sociale cliente}
    label $base.lab_desc -borderwidth 1 -font $FONT_LABELS -text {Descrizione articolo} -width 30
    label $base.lab_quan -borderwidth 1 -font $FONT_LABELS -text {Quantita} -width 12
    label $base.lab_puni -borderwidth 1 -font $FONT_LABELS -text {Prezzo unitario} -width 12
    listbox $base.lis_desc -font $FONT_TEXTS -selectmode single
    listbox $base.lis_quan -font $FONT_TEXTS -selectmode single
    listbox $base.lis_puni -font $FONT_TEXTS -selectmode single

    button $base.but_ristampa -activebackground #d9d9d9 -activeforeground #000000 \
        -background #0000fe -borderwidth 3 -foreground #fefe00 \
        -highlightcolor #000000 -padx 9 -pady 3 -font $FONT_BUTTONS \
        -text Ristampa \
        -command {
set risp [tk_messageBox -title "Attenzione!" -message "Ristampo la fattura?" \
    -icon question -type yesno -default no]
if {$risp == "yes"} {
    cd $CIP_PATH
```



```
catch [set fattura_clickata [.cipArchivioFatture.lis_numfatt curselection]]
# --- Prelevo il N. della fattura selezionata:
set indice [.cipArchivioFatture.lis_numfatt get $fattura_clickata]
# --- Trasformo il file HTML in PS...
catch [exec html2ps -d -c -i 0.25 $BACKUP_PATH/fattura-$indice.html > $CIP_PATH/fatt.ps]
catch [exec gs -q -dBATCH -dNOPAUSE -sPAPERSIZE=a4 -sDEVICE=ljet4 -sOutputFile=\\pr $CIP_PATH/fatt.ps]
catch [exec gs -q -dBATCH -dNOPAUSE -sPAPERSIZE=a4 -sDEVICE=ljet4 -sOutputFile=\\pr $CIP_PATH/fatt.ps]
file delete $CIP_PATH/fatt.ps
destroy .mostraDettagliFattura
destroy .cipArchivioFatture
}
}
button $base.but_chiudi -activebackground #d9d9d9 -activeforeground #000000 \
-background #fe0000 -borderwidth 3 -foreground #fe0000 \
-font $FONT_BUTTONS \
-highlightcolor #000000 -padx 9 -pady 3 -text Chiudi \
-command {destroy .mostraDettagliFattura}
#####
# SETTING GEOMETRY
#####
place $base.but_ristampa -x 420 -y 6 -width 86 -height 50 -anchor nw -bordermode ignore
place $base.but_chiudi -x 512 -y 6 -width 60 -height 50 -anchor nw -bordermode ignore
place $base.lab_numfatt -x 5 -y 10 -width 100 -anchor nw -bordermode ignore
place $base.ent_numfatt -x 5 -y 30 -width 100 -height 25 -anchor nw -bordermode ignore
place $base.lab_ragsoc -x 110 -y 10 -width 352 -anchor nw -bordermode ignore
place $base.ent_ragsoc -x 110 -y 30 -width 305 -height 25 -anchor nw -bordermode ignore
place $base.lab_desc -x 5 -y 75 -width 340 -height 25 -anchor nw -bordermode ignore
place $base.lab_quan -x 344 -y 75 -width 100 -height 25 -anchor nw -bordermode ignore
place $base.lab_puni -x 443 -y 75 -width 110 -height 25 -anchor nw -bordermode ignore
place $base.lis_desc -x 5 -y 100 -width 360 -height 400 -anchor nw -bordermode ignore
place $base.lis_quan -x 364 -y 100 -width 100 -height 400 -anchor nw -bordermode ignore
place $base.lis_puni -x 463 -y 100 -width 110 -height 400 -anchor nw -bordermode ignore
#####
# CODE
#####
catch [set fattura_clickata [.cipArchivioFatture.lis_numfatt curselection]]
# --- Prelevo il N. della fattura:
set indice [.cipArchivioFatture.lis_numfatt get $fattura_clickata]
set file_dettagli_fattura [open $BACKUP_PATH/fattura.$indice]
gets $file_dettagli_fattura linea
$base.ent_numfatt insert end $linea
gets $file_dettagli_fattura linea
$base.ent_ragsoc insert end $linea
while {[gets $file_dettagli_fattura linea] >= 0} {
    $base.lis_desc insert end $linea
    gets $file_dettagli_fattura linea
    $base.lis_quan insert end $linea
    gets $file_dettagli_fattura linea
    $base.lis_puni insert end $linea
}
close $file_dettagli_fattura
}
```