USE OF THE INGRES RDBMS INSIDE THE NEW GANIL LINUX BASED CONTROL SYSTEM

E. Lécorché, P. Lermine
Ganil, Caen, France

ABSTRACT

The Ganil control system has been entirely relying on the Ingres relational database management system (RDBMS) for almost fifteen years, its use gradually increased during this period so that it is really now a critical point and a key issue of the system.

The control system has been upgraded from the previous VMS one during the last winter shutdown so that both clients and servers are now running within the Linux operating system. The whole existing application software programmed in Ada has been quite easily migrated, keeping the Motif graphical user interface as the man machine interface and still addressing the on-line Ingres databases through Ada/Sql interfaces. To provide high availability capabilities, the Hewlett-Packard service guard package has been adopted featuring a fail-over cluster environment among which the Ingres configuration has been fully integrated.

This paper gives a short overview of the rejuvenated control system then describes the main uses of the relational database management system (naming service, alarms archiving, pieces of equipment configuration, beam parameters management …). Lastly it focuses on the Ingres integration and implementation into the new architecture.

INTRODUCTION

The Ganil facility located in Caen (France) consists of cyclotrons in cascade thus allowing to accelerate heavy ion beams for nuclear physics, atomic physics, radiobiology and material irradiation ; since December 1997, its Spiral extension has been producing radioactive ion beams using the so-called ISOL technique. A new project named Spiral2 has been approved in May 2005 and aims to produce rare ion beams using a Uranium carbide fission process : a Linac accelerator will accelerate deuteron beams impinging on a carbon converter to produce the neutrons to get the fission.

Since its very beginning, the Ganil control system integrates a relational database management system (RDBMS) to implement many functionalities. In a first approach, the RDBMS has been used at the background level where it can be stopped so making the control done within a degraded but functional mode ; then, it became progressively a critical component of the control system so that now most of the control software relies on the use of the RDBMS package.

The Ingres product has been adopted more than fifteen years ago and is still in use even if its whole environment deeply evolved since the first implementation. Ingres offers all the basic capabilities of a relational database management system and is currently now provided by Computer Associates which also delivered an open source release by the end of 2004.

A major evolution recently migrated the control system from the former VMS context to a new Linux based environment. After briefly describing the control system, the presentation focuses on the use of the databases inside the control system and then presents the RDBMS implementation into the renewed system.

THE REJUVENATED CONTROL SYSTEM

The Ganil control system has to handle more than 4500 pieces of equipment in order to send the beam to the experimental areas. It has been deeply renewed by the end of 2004 in order to migrate from the VMS operating system to the Linux environment [1].

It is a three layers system following a quite classical approach :

- A failover clustered server provides the central services : on one hand the off-line needs such as development tasks, code generation, beam preparation and analysis ; on the other hand the real-time level servers for the alarms handling, real-time archiving, data collectors, Web server .... The cluster itself consists of two HP Proliant ML 350 G4 machines running the Red Hat Enterprise Linux 3 operating system while the failover capability is achieved through the HP Service Guard package. An Ingres server is running on each of the two nodes of the cluster.

- Operator consoles consist of Linux PCs equipped with two flat screens and one knob box linked to the PC through a serial or USB port. They are able to run either a general purpose control program or any of the 50 dedicated tuning programs for the whole facility (magnetic field adjustment, cyclotrons isochronism, beam adaptation, emittance settings, RF handling …).

- Front-end real time crates are used to drive pieces of equipment, most of them are within the VME standard and running the VxWorks environment ; for historical reasons, there are still some CAMAC crates under the VaxELN operating system. Pieces of equipment are reached by direct I/O cards or interfaced through field buses (the most commonly used being the Jbus one while CANbus is under test for future use). Three VME crates are used as gateways to Profibus networks allowing to interface Siemens PLCs (S5 and S7 series).

The following schema gives an overview of the renewed control system :

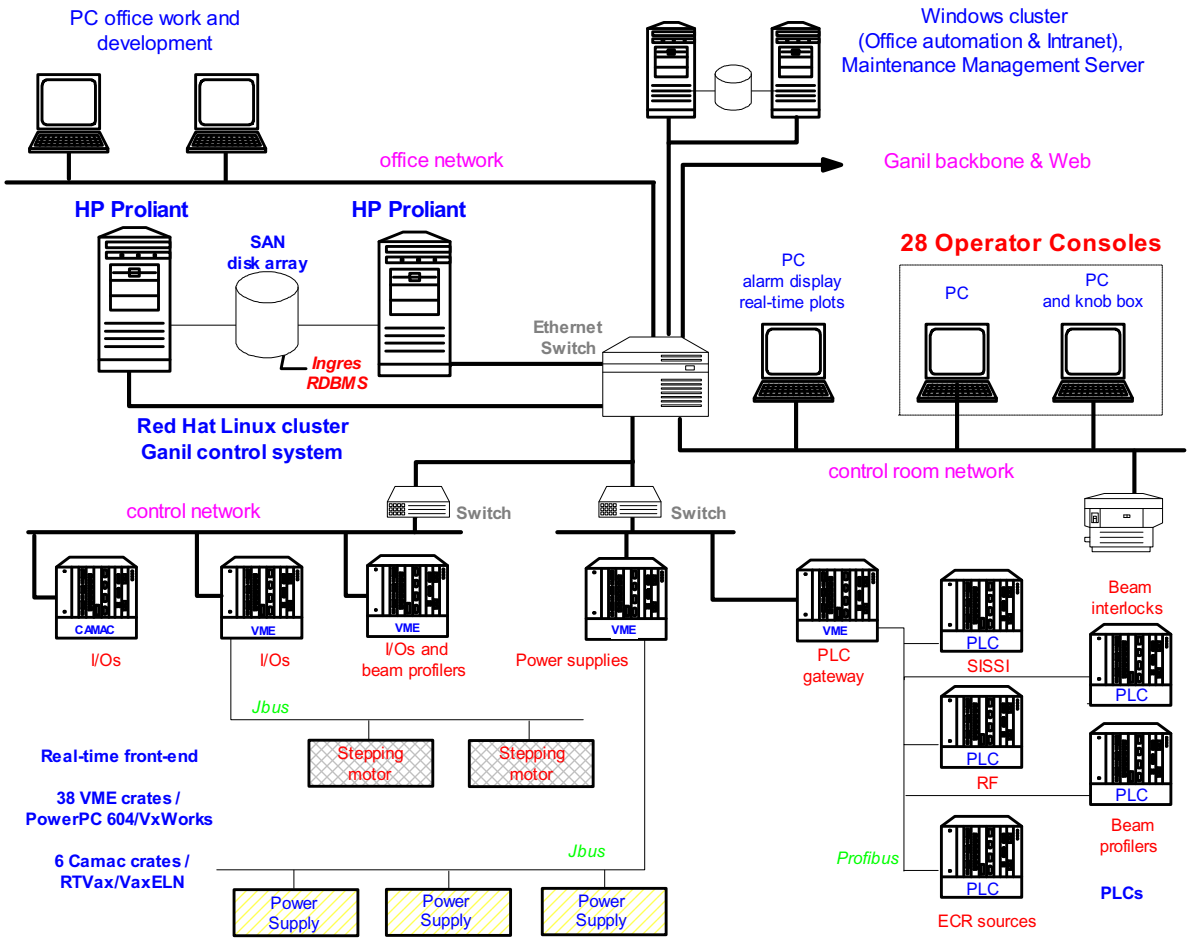

Figure 1: Schematic overview of the Ganil control system

As far as software is concerned, Ada 95 language is used over the whole control system, being the programming language both for the central servers, the operator consoles and the front-end crates.

DATABASES USE AMONG THE CONTROL SYSTEM

The RDBMS therefore provides all the data management facilities commonly involved into a typical control system [2] :

Pieces of equipment data configuration

All the 4500 pieces of equipment are listed into an equipment database referencing all data related to their behaviour and the way to control them. They are sorted into 20 types according to the real-time handler driving them inside the front-end crates : "power-supplies", "stepping motors", "generic analog input / ouput", "generic digital input / output", "vacuum measurement", "analog PLC variable", "digital PLC variable" … Typically one piece of equipment consists of one to several Input / Outputs channels. Data related to any piece of equipment are of several kinds :

- Physical path to access the equipment : front-end name, slot number or memory address, sub-address, field-bus slave number ...

- Equipment specification : lower and upper bounds, scaling factors, offset values, stability ...

- Real-time handling : alarms, thresholds, scanning periods, tolerance and survey ranges ...

- Presentation data : dynamic configuration for displaying Motif widgets on operator interfaces, engineering units …

Devices are under the control of the front-end processors embedded into either CAMAC or VME crates. For each front-end crate, three binary files are generated from the relational database : the first one describes the hardware configuration of the crate (I/O boards embedded into the chassis), the second one lists all the alarms able to be send from the crate (hardware address, local real-time handling ...) and the third one is the real-time equipment database describing all pieces of equipment to be controlled by the crate. Therefore, each crate is autonomous as it has its own "real time database" which consists of these three files being extracted from the Ingres piece of equipment database after any database modification. Then, in a second and well separated phase, these files are downloaded at front-end boot time into the processor live memory. For the devices handled by PLCs, a specific database is downloaded into the VME Profibus gateways, including the internal PLC addresses, pointers and data blocs information.

As described here, the equipment database is not directly reached on-line from the control system, the off-line configuration of the piece of equipment database is performed through a graphical application written within the Computer Associates Open Road environment. From this application people can consult, modify or add equipment ; they can also check connections data and cabling information attached to pieces of equipment ; lastly, they can get historical records or retrieve statistics concerning the boot of the front-end crates.

In order to provide a naming service, a specific table of this database is accessed from a dedicated Ada package (integrating a cache mechanism to reduce the frequency of queries performed to the Ingres database) : for each piece of equipment are retrieved the name of the front-end crate handling it, indexes allowing a quicker access to records of data associated to the piece of equipment, the software packages to be dynamically used to interface the device.

Alarms handling

The alarm database consists of two kinds of data : first of all, the alarm hardware configuration (interrupt signal address, piece of equipment concerned if any, dynamic parameters …) sent to the front-end crates as presented in the previous paragraph, these data allow each front-end to format and send on the network the appropriate alarm messages according to interrupts or software requests. Other data are read by the on-line central alarm server which will receive these messages to be able to process them in order to display and archive the alarms which occurred in the system : these data are mainly the meaningful Ascii alarm test, severity labels, formatting directives, display devices …

An off-line Open Road application allows to retrieve archived alarms according to several criteria (date, piece of equipment, severity, node or process having raised the alarm …).

Log book reference

The log book database stores data from the daily operation to record the chronology of the Ganil tuning process : beams accelerated and optics in use, experimental rooms into which the beam is delivered, occurring failures. The database is filled through an off-line Open Road application by the operators, some on-line programs also update automatically specific data such as the accelerator configuration.

These data are later extracted to office automation tools to generate operation statistics : beam time distribution, failures rate and distribution, beam availability …

Beam parameters management [3]

One major capability of the Ganil facility is its ability to accelerate a large range of beams depending on the ion (from He to U) and their acceleration characteristics (typically : isotope, charge, energy …). Furthermore, for each ion to be produced, the Ganil complex can be configured within various modes and beam optics.

In order to deal with this process, the beam parameters database has been designed to allow people to manage the beam configuration from the off-line beam preparation to the on-line use by tuning programs as shown on figure 2 :

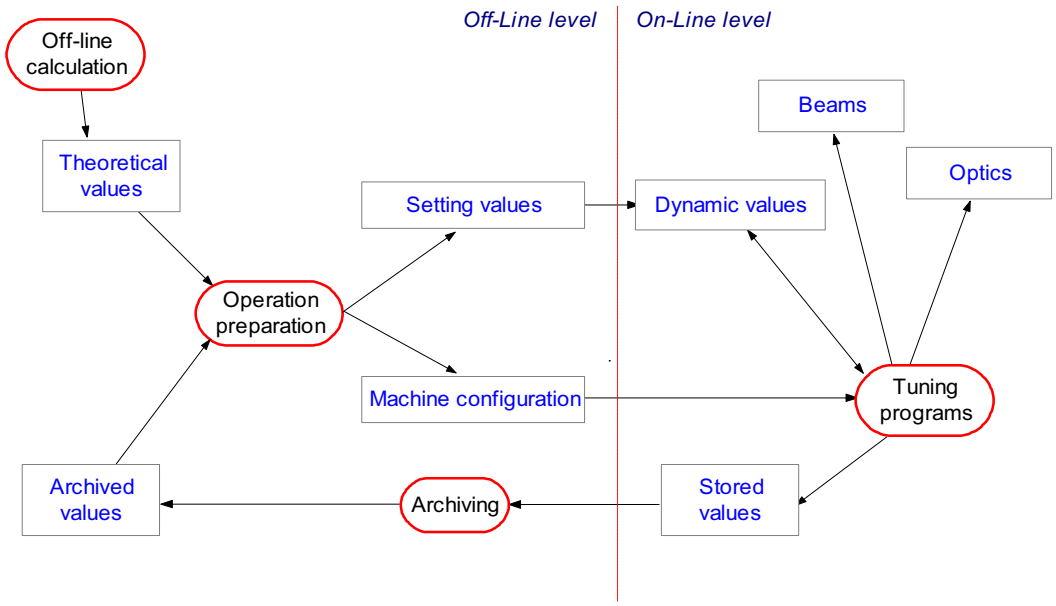

Figure 2: Beam parameters dataflow diagram

- An off-line program calculates the theoretical values for the beam to be accelerated. Then setting values to be later applied to pieces of equipment are prepared by integrating both these theoretical values and archived values from former beams previously tuned and recorded. Procedures to upload the output of the off-line calculation program into the database and to prepare the set of values of the beam to be accelerated are bash procedures invoking SQL routines ; for a simpler use, PHP applications also give people the ability to consult the database.

- On-line Ada programs read the setting values and manage a set of dynamic values which are updated when tuning the machine according to the beams and optics to be applied and the beam tuning phases (emittance settings, cyclotrons isochronism, beam lines adaptation ...). Stored values are extracted and eventually archived if of interest.

Reference data for tuning programs

The design of the beam parameters management database required to provide a complete description of the machine ; this has been achieved by adopting an object approach first specifying classes corresponding either to physical devices such as quadrupoles, dipoles, steerers, NMR probes, slits, beam profilers, RF systems, or to more conceptual concepts not necessary directly attached to any piece of equipment (beam data, emittance, beam adaptation, isochronism …). Then objects are defined belonging to these classes and having a fixed number of entities depending on their class ; as an example, an object of the quadrupole class has two entities : the current intensity delivered by the power supply connected to the quadrupole and the magnetic gradient inside the coils ; an object of the beam class has 15 entities (Z, A, Q, W, Bρ …) ...

By adding attributes to the objects and entities, the database has been extended and integrates new data to be used as reference for tuning programs (magnetic data, geometrical configuration, beam optic dependence, correlation parameters …). Many other data have been progressively introduced into the database so that most of the 50 Ada tuning programs are connected to this repository database.

Daily recommendations or warnings

A small database records the daily recommendations or warnings for the machine operation. Information can be created, updated, selected or destroyed from a PHP application.

INGRES ARCHITECTURE

Basic configuration

The current Ingres version in use is the 2.6 one, the introduction of the open source Ingres r3 release will be considered in a second step. As the replication process is not set up, only the database engine ("iidbms"), the recovery ("dmfrcp") and archiver processes ("dmfacp") are running to implement the server functionality ; to interface the client processes, the communication server ("iigcc") and name server ("iigcn") are configured.

For basic operations or development of built-in procedures, the common standard tools of Ingres are used ; graphical applications have been developed using the Open Road environment to provide a more user friendly interface. A PHP gateway interface also gives people the ability to access the databases from an Intranet context.

All the Ganil control software is written in Ada language and more than 200 Ada / SQL procedures have been written in order to interface the application programs with the relational database. So some work had to be done jointly with Computer Associates in order to adapt the Ada 83 / SQL gateway under VMS to the new environment based on the use of the Ada 95 Gnat compiler and this point was a key issue for the migration project to Linux. Once this has been solved, the specific package providing a re-entrant Ada multitasking access to the database had to be modified because of some differences of the multithreading handling.

Because of the functional needs, the Ingres installation consists of two separated servers, one for the off-line purposes and the other one for the on-line access from the control programs ; when required, specific home-made replication mechanisms are provided from one server to the other one.

Cluster implementation

The Linux cluster on which the Ingres database is running is based on the HP Service Guard software on top of the Red Hat Enterprise distribution. The cluster consists of two nodes accessing a SAN array with Raid 1 mirrored disks. The principle is to create software packages which can be switched from one node to the other one in case of problem, each package having its own IP address.

As previously said, for functional requirements, two different Ingres servers have to be run (for the so-called "off-line" and "on-line" needs) ; so, we decided to create two cluster packages "bddevt" (for development) and "bdoper" (for operation), each one integrates an Ingres server and normally executes on each of the two cluster nodes.

As these packages must be able to switch from one node to the other one, the Ingres servers must be first configured to be able to run on each of the two nodes. To work properly, the name of the physical node on which an Ingres server is running must be written at different levels into its configuration file ; so, this means in our case that each of the two Ingres server configuration files have to integrate the two node names and environments of the clustered machines. This configuration is simply done by configuring Ingres on the node as usually, then duplicate the whole configuration and lastly change the node name inside the copy/pasted part. A second problem comes from the fact that when commuted to a "degraded" mode, the two servers must be run simultaneously on the same node ; this is easily achieved by giving the servers specific installation codes ("OP" and "DV") instead of the standard one ("II").

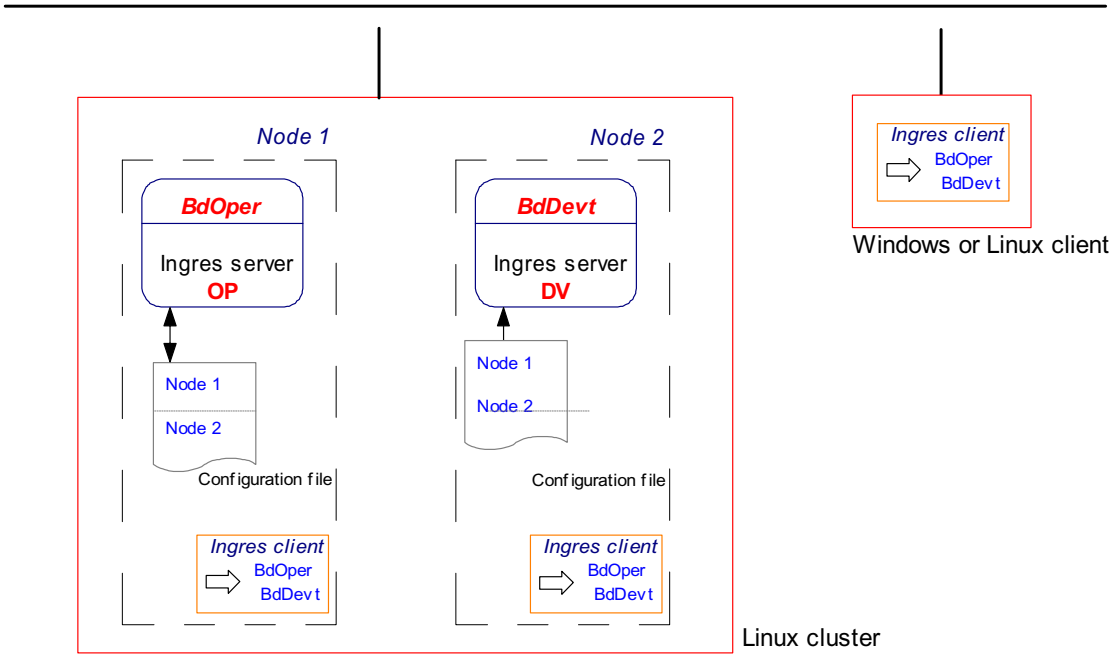

Figure 3: Ingres clustered installation

On the client side, the client/server configuration specifies as remote access the package name instead of the node name as commonly done so the communication is established with the package which can be switched from one node to the other one.

Note that we didn't implement a recovery mechanism into the application programs so that in case of package switching, application programs have to be stopped and re run as communications have been lost. Because of the nature of the application, we don't consider that to be a real problem.

CONCLUSION

After almost 15 years of use, the Ingres RDBMS and applications related to it have been successfully migrated from the VMS environment to the new Linux one. Taking benefit of the clustered environment, reliability has been increased so providing a more secure platform.

Therefore, it is quite encouraging for future evolution, mainly for the Spiral 2 project integration we will have to cope with for years to come.

REFERENCES

[1] L. David and the Ganil control group, “The new Linux based control system for Ganil”, this conference.

[2] E. Lécorché, P. Lermine, “Database management in the new Ganil control system”, ICALEPCS’93, Berlin, Germany, October 1993.

[3] E. Lécorché et al, “Use of an Ingres database to implement the beam parameter management at Ganil”, ICALEPCS’95, Chicago, USA, October 1995.