

SPARTA for the VLT: status and plans

Enrico Fedrigo^{1a}, Reynald Bourtembourg^a, Robert Donaldson^a, Christian Soenke^a, Marcos Suarez Valles^a, Stefano Zampieri^a

^aEuropean Southern Observatory, Karl-Schwarzschild-strasse 2, 85748 Garching, Germany;

ABSTRACT

SPARTATM, the ESO Standard Platform for Adaptive optics Real Time Applications, is the real time computing platform serving 3 major 2nd generation instruments at the VLT (SPHERE, GALACSI and GRAAL) with plans to serve more, smaller, instruments in the near future. SPARTA offers a very modular and fine-grained architecture which is generic enough to serve a variety of AO systems. SPARTA includes the definitions of all the interfaces between those modules and provides libraries and tools to implement and test the various modules as well as a map to technologies capable of delivering the required performance, most of them innovative with respect to ESO standards in use.

For the above mentioned instruments, SPARTA provides also a complete implementation of the AO application, with features customized for each of the 3 instruments. In this paper we present the architecture of SPARTA, its technologies, functions, performance and test tools as well as the plans to increase the reach of the platform to smaller system with what we call SPARTA Light.

Keywords: real time, real time computer, adaptive optics, control systems.

1. INTRODUCTION

ESO has started several projects to provide the four Very Large Telescope (VLT) telescopes with new state-of-the-art instruments to replace some of the existing ones. These new instruments are collectively called “second generation VLT instrumentation” and some will be equipped with Adaptive Optics (AO) facilities. The core component of any AO system is the Real Time Computer (RTC) which measures the incoming wave-front aberrations by means of a sensor and corrects for them by means of a deformable mirror. The requirements that this new generation of instrumentation poses on the Adaptive Optics facilities are challenging, in particular to the real-time computer.

At the same time ESO is investigating the next generation of giant telescopes where the Adaptive Optics system, a fundamental facility of the telescope, will have to face even harder requirements.

Currently no single-board computer is capable of processing the amount of real-time data required to run these AO systems. Extreme AO (XAO), Multi-conjugate AO (MCAO) or simply Multi-Target Ground Layer Correctors will require about one thousand of actuators for 8m class telescopes like the VLT running at a Kilohertz rate or more. Even more actuators and higher speed will be needed for the E-ELT.

The Adaptive Optics Real Time Computer has to be based on multi-processor multi-board computers in order to achieve the required computational power. The complexity of each of these systems and their number raised concerns about the complexity of their development, their reliability and their maintenance. Individual efforts aimed at developing different custom systems were simply unaffordable not only for the duplication of the design and development effort to build similar products for different systems, but also the amount of resources required to test, maintain and upgrade systems which are different in spite to being similar.

The selected solution was a common standard platform that could achieve all the goals of the AO systems for the second generation instrumentation and be capable of targeting the simplest AO systems for E-ELT.

This platform, named SPARTA for Standard Platform for Adaptive optics Real Time Applications, provides both a hardware and software common infrastructure in which all the previously mentioned applications can run.

ESO's experience of previous Adaptive Optics systems² delivered to the Paranal Observatory shows that the core functionality of each AO system is limited to about the 20% of the developed software, while the remaining 80% constitutes the infrastructure that performs the same tasks, but in a different way because it has been developed for different systems. It is therefore possible to develop a common software infrastructure that covers 80% of the application

* enrico.fedrigo@eso.org

² NAOS-CONICA, MACAO-VLTI, SINFONI, CRIRES

needs and is developed and maintained only once for all systems. A common development means also higher quality and reliability since the same code is tested multiple times in the various systems where it is installed.

However this becomes possible only if the hardware is standardized as well, in order for the software to run always within the same platform. This choice became the critical choice for SPARTA. The standard hardware platform had to be scalable to accommodate different needs of the different projects, easily maintainable and upgradeable to follow the technological evolution as much as possible, had to provide a powerful development environment to make the development easy and the final product reliable, and be easy to use strictly following standard operational scenarios defined by the observatory.

The first requirement, scalability, has been achieved through modularity: SPARTA can be run with a variable number of computational boards.

The second requirement, maintainability, has been fulfilled through the use of Commercial Off The Shelf (COTS) components that are easier to procure and repair than custom developed hardware. COTS components help also in terms of reliability since these are components used in other projects and other companies and so better tested than a custom component specifically developed for a single project.

The third requirement, upgradeability, is also fulfilled by the use of COTS products: in many cases it could be true that a custom development can achieve better performance or better integration than a collection of COTS elements. But COTS elements are faster in following the technological evolution and quickly surpass the custom development. In turns, the custom development requires a heavy investment to benefit from the same technological evolution, since normally this involves restarting the complete design / development / test process.

The fourth requirement, a rich development environment, has been met by using industry standard environments (including operating systems) that are widely used and thus feature a large portfolio of tools.

Finally, usability can be achieved by capitalizing on the previous experience and by collaborating with the observatory personnel since the early stages of design.

The final result of this project is not only a powerful standard real-time computer for adaptive optics applications, but also a platform for high-performance parallel computing with fast I/O. The software toolkit designed to fulfill AO-related requirements, in its building blocks can also be used for other applications.

2. OVERVIEW

The aim of the SPARTA project is to provide a standard platform to build adaptive optics real-time computers based on the requirements of its client projects. SPARTA is not an AO RTC, but provides the tools to build one. Therefore, SPARTA provides the following components:

- **Interface definitions**
 - **External interfaces** to connect to sensors (ESO NGC and Andor iXon supported) and actuators (ESO-defined CODE interface)
 - **Internal interfaces** to interconnect the different HW components
- **Test tools** that can be used to extensively test the platform and its instances at various interface levels
- **Hardware** components to build the system
- A **Software** framework that provides the functionality which is common to all AO systems (e.g. data communication, configuration, storage, math, etc.)

The SPARTA software executes over a number of different platforms depending upon the performance requirements which have to be met, these platforms are:

- A low latency real-time processing pipeline known as the **RTC box**
- A Linux **co-processing cluster**
- A standard ESO **Instrument workstation**

Figure 1 shows a schematic representation of the SPARTA components and the platform in which they run.

The RTC box handles all tasks directly involved in the hard real time low latency AO control loop; acquisition of wavefront sensor data, reconstruction of the incoming wavefront and control of the final mirror actuator positions. The RTC box is based on a 19" VXS system hosting several specialized embedded computing boards. As well as implementing the core AO loop the RTC box continuously sends sensor and actuator data to the co-processing cluster for additional processing. All wavefront sensors and actuators are directly connected to the RTC box via fiber links.

The co-processing cluster handles tasks which do not require low latency but still require a significant amount of computing power. These tasks include: atmospheric statistics estimation, calibration, loop optimisation and data recording. In addition, one application in the co-processing cluster acts as a supervisor of the entire RTC system coordinating all of the activities. The co-processing cluster comprises a number of multi-CPU multi-core computers connected together and to the RTC box by means of a dedicated isolated GBit Ethernet LAN.

An ESO-standard Instrument Workstation hosts the processes which implement the external interface to the RTC, provide coordination with other software operating in the VLT, e.g. wavefront sensor controllers and TCS and hosts the user interfaces for the RTC.

An example hardware configuration for a SPARTA RTC using LGS (GALACSI) is shown in Figure 2.

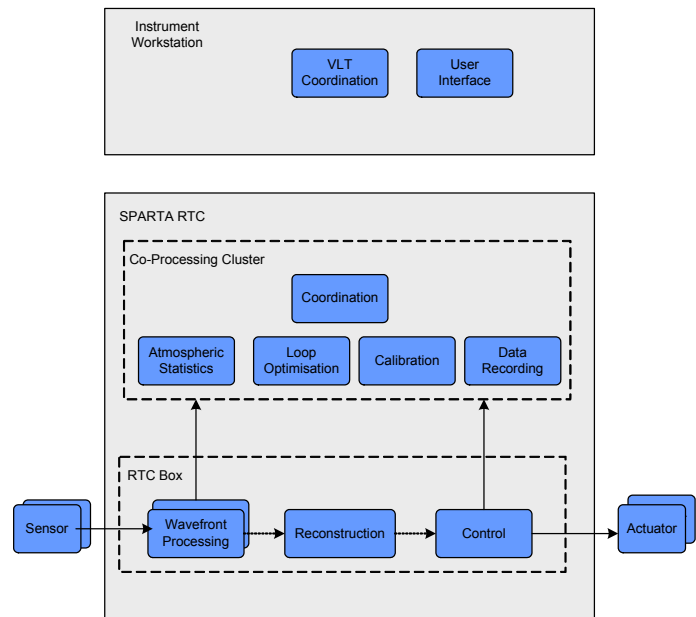


Figure 1 SPARTA Architecture Schematic

2.1 External & Internal Interfaces

As a goal SPARTA attempts to reduce the number of different interface types to be supported to the minimum. For real-time deterministic high-throughput data communication, a single standard protocol has been adopted, SerialFPDP. This protocol runs over copper within the RTC box and over fiber for the external interfaces at distances up to several hundreds of meters. It provides a throughput of 2.5Gbit/s together with low latency.

This interface is used to acquire pixels from the NGC, for sending actuator commands to the DM and for control of the LGS jitter actuators when required and RTC box internal communication.

A second high-throughput non deterministic protocol, Gigabit Ethernet, is used for communication between the real-time box and the co-processing cluster, within the cluster and for communication with the instrument workstation.

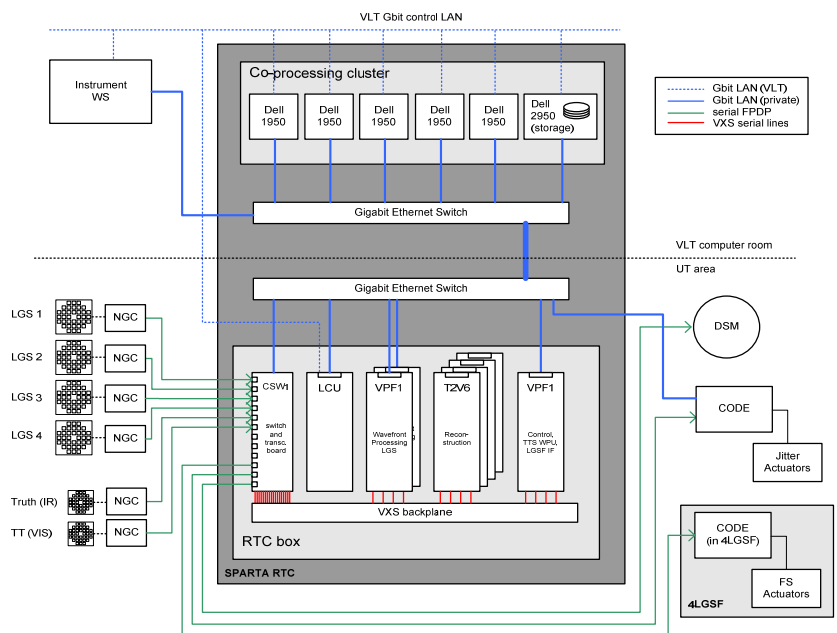


Figure 2 Example SPARTA Hardware Configuration

2.2 SPARTA RTC Box

The SPARTA RTC box implements the hard real-time low latency adaptive optics control loop. This loop comprises the following tasks:

- **Wavefront Processing**
Conversion of the input pixel stream into a wavefront measurement in the form of slopes.
- **Reconstruction**
Reconstruction of the wave-front and calculation of the delta actuator positions to be applied by means of matrix vector multiplication.
- **Control**
Control of the final actuator positions by application of a filter such as IIR or Kalman.

Within the RTC box different technologies are employed for each of these tasks depending upon the requirements in terms of processing and flexibility.

- **FPGAs**
Are used for input/output processing and operations which are largely integer based and highly parallel. Only operations which are considered well defined are hosted in FPGA as the development cycle is usually longer than for other types of processing elements. In SPARTA FPGAs are used as data routers and for wave-front processing.
- **DSPs**
Are used for floating point operations and operations which require significant memory. Since DSPs are programmed in “C” they offer a faster development cycle than FPGAs but development is still significantly more complicated than for CPUs. In SPARTA DSPs are used for reconstruction.
- **CPUs**
Are used for operations which require significant algorithmic complexity and whose definition is not well fixed. This environment has the fastest development cycle allowing rapid prototyping and modification of algorithms. In SPARTA CPUs are used for controller implementation and for monitoring and control of the other processing elements within the RTC box.

Physically the RTC takes the form of a 19” chassis with a VXS backplane. Data are distributed within the RTC box over a switched fabric implemented by the serial lines of the VXS backplane. VXS supports point to point full duplex serial links running up to 6.25Gb/s, but SPARTA uses them only at 2.5Gb/s. This use of point to point connections rather than traditional bus architectures has been found by benchmarking to be essential to achieve the latency and throughput requirements for SPARTA.

Development of hard real-time firmware and software for FPGAs and DSPs is significantly more complex than development of software in a traditional workstation environment due to the inherent complexity of the debugging process and development tool chain. Moreover, the development of hard real-time firmware and software for FPGAs, DSPs and CPUs is more complex as normal workstation applications due to the difficulties of hard real-time programming. This has led in the past to the RTC being one of the least reliable components in adaptive optics systems. To avoid this pitfall in SPARTA only the minimum set of operations are performed within the RTC box, the goal is to reduce the volume of code running in the RTC box to the absolute minimum.

Any operations which do not require low latency are offloaded to the co-processing cluster. To allow this co-processing to take place the loop data, slopes representing the wavefront and actuator positions representing the shape of the deformable mirror are sent to the co-processing cluster in real-time by means of a private GBit Ethernet LAN.

2.3 SPARTA Workstation and Co-processing cluster

The SPARTA co-processing cluster is a set of multi CPU / multi core Linux workstations providing a comfortable development environment, significant computing density and good SW library support.

The co-processing cluster fulfils three major roles in the SPARTA architecture

- Provides a platform for CPU intensive mathematical processing
- Supports the recording in real-time of AO loop data
- Coordination

Mathematical processing is required for the tasks of atmospheric statistics estimation, calibration, and loop optimisation. To support these operations, highly optimized parallel mathematical libraries are available. Recording of AO loop data in real-time is required for system calibration and for diagnostic purposes. The volume of data produced by a multi-sensor AO system is in the region of 80Mbytes/s. To be able to stream these data to disk in real-time for periods of more than a few seconds a RAID array is required. One computer of the cluster is equipped with a RAID interface connected to an appropriately sized internal RAID array and perform the task of data recording.

In any distributed system the selection of the middleware supporting inter-process communication and data distribution is a key design choice. For a platform designed to be used by VLT 2nd generation instruments clearly a standard VLT SW interface for commands and on-line database is essential; however the usage of the standard VLT SW was not considered appropriate for highly multi-threaded systems such as SPARTA. As a result of a joint internal effort, the selected middleware for SPARTA is ACE/TAO with DDS providing a CORBA implementation for method invocation and an implementation of the OMG Data Distribution Service for publish-subscribe data distribution. However SPARTA is still a system intended to run within a VLT SW environment, therefore a number of gateways are provided. Those gateways are modular components that are added to the system to interface it to the observatory, but are not strictly part of the platform itself. In fact the SPARTA could run without gateways. Those are:

○ **CCS Command and Error gateway**

The gateway receives standard CCS commands specifying which object to address, the method to invoke and the parameters to use. Then the gateway translates the command to a CORBA command, including type conversion for the parameters, and waits for the reply of the object. The object replies with a CORBA sequence that is translated to a string and/or status (OK or FAILURE). In the event of an error, received as CORBA exceptions, the gateway packs together all the errors generated at the various stages of the computation into a CCS ErrorStack and returns it to the caller. At the same time, the error stack is also sent to the log system.

○ **Log Gateway**

The gateway subscribes to a dedicated DDS topic and posts to the standard log system all the log messages generated by the SPARTA objects, either simple text messages or FITS logs.

○ **OLDB Gateway**

The gateway subscribes to a dedicated DDS topic. Each packet carries the name of an object and the parameter that has been changed so that the OLDB Gateway can map this event to a On-line DB point with structure “<alias>SPARTA:<object name>.<parameter name>” to which the specified value is assigned. In this way parameters like *state*, *substate*, *initialised*, *error* and others are visible from the On-line DB and can be shown in standard panels.

○ **Display Server**

The Display Server attaches to RTDF sources (DDS high performance topics) to display various dynamical information like the pixels on the detector or the voltages of the mirror through either a standard RTD at low speed (maximum few Hz).

○ **NGC Gateway**

This gateway offers a CORBA interface to drive the NGC workstation process (which is a VLT SW process) from the SPARTA CORBA objects.

○ **FITS Header/Table generator**

This component subscribes to specialised DDS topics that collects period information about the system, typically performance figures. Those data are used to populate a FITS table that is initialised at the beginning of the

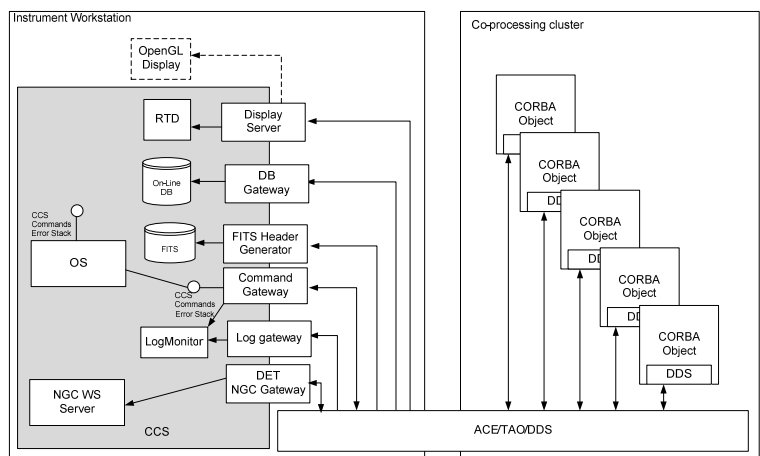


Figure 3: SPARTA Workstation Software architecture

observation and closed at the end. At the end of the observation this process also generates an AO FITS Header that adds extra global information on the performance of the AO system during the observation. Those FITS files are then passed to the upper level software components to be merged with the final scientific files. Content of the table and header are system dependent.

All the gateways and VLT SW processes run in the Instrument Workstation.

The SPARTA co-processing cluster implements a number of soft real-time applications which do not require low latency but do require a significant amount of computing power. These applications include calibration, atmospheric statistic estimation and loop optimization amongst others. One process running in the cluster coordinates the operations of the soft-realtime applications with those running in the hard-realtime box and provides a single interface to the RTC.

The hardware solution selected to implement the SPARTA co-processing cluster is a collection of multi CPU, multi core Linux computers connected to the SPARTA real-time box by means of a private Gigabit Ethernet.

The software running within the SPARTA supervisor is inherently distributed and requires the normal facilities of distributed systems: message passing, centralized logging, error reporting, distributed database access. Each application is implemented as a separate server derived from a standard SPARTA object class implementing a number of standard CORBA interfaces. The applications running within the cluster can be divided into the following categories:

○ **Coordination**

One computer within the cluster acts as the SPARTA supervisor. A single application running in the cluster coordinates all applications running within the cluster together with those running in the RTC box. Additional components operating in the cluster coordinate the acquisition, reconstruction and control components running in the RTC box.

○ **Calibration**

A calibration application implements the basic interaction matrix measurement techniques supported by SPARTA and provides a mechanism by which to invert interaction matrixes to create and store a control matrix. Additional calibration techniques can be provided which are specific to each instrument by extending the basic calibration task. Multiple calibration objects can run to serve different loops.

○ **Auxiliary Processing**

A number of auxiliary processing applications are required for loop optimization and atmospheric statistics estimation.

○ **Data Recording, Distribution and Measurement**

One computer within the cluster is sized appropriately to acquire all of the data from the RTC box in real time. This computer is equipped with a high throughput RAID array. Applications running on this computer are provided to stream data to the RAID array, distribute data to the instrument workstation for graphical display, and perform measurements such as averaging of RTC data frames.

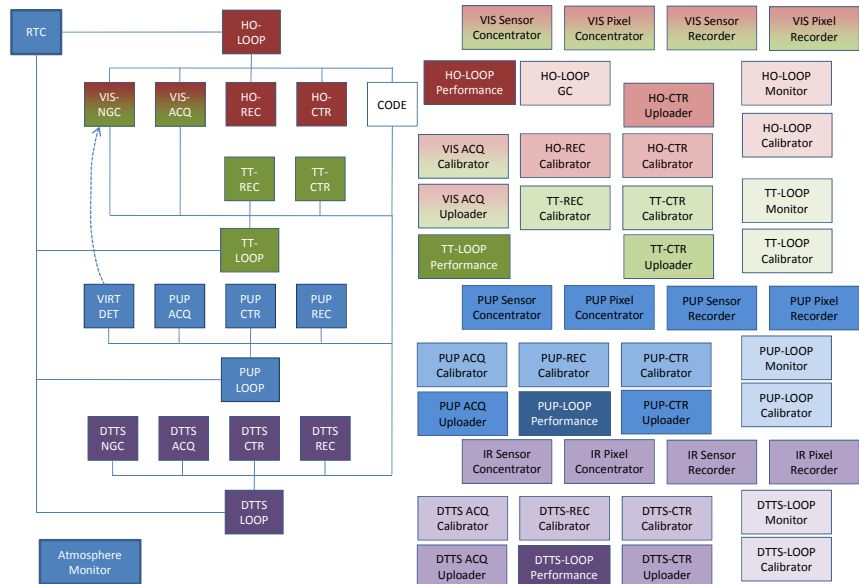


Figure 4: Planned deployment map for SPARTA for SPHERE/SAXO. Each box in the diagram is a CORBA object.

To support the development of the various cluster applications a number of support packages are provided, these packages are:

- **Configuration Data Management System (CDMS)**
All adaptive optics systems need to generate and share vectors and matrixes representing control matrixes, background maps etc. To support this functionality a tool is required implementing a distributed publish/subscribe interface, this tool is known as CDMS.
- **SPARTA Drivers**
To provide a clean interface separation between cluster and RTC box the functions within the RTC box are invoked by means of a network driver with a simple client server protocol supporting the operations of reading and writing memory and invoking commands. A set of classes implementing a basic SPARTA server and client interface is provided.
- **Mathematical Package**
The bulk of the processing performed by the auxiliary processing applications takes the form of FFT, SVD or MVM operations. To support these operations a set of mathematical libraries is included as part of the SPARTA platform.

The intended deployment of the SPARTA objects for the SPHERE case is shown in Figure 4. This diagram shows 4 different loops: HO for High-Order, TT for Tip/Tilt, PUP for Pupil control, DTTS for Differential TipTilt Sensor (a loop running with an infrared sensor). Each loop has 5 core objects, a DET (detector interface, also a gateway), ACQ (acquisition), REC (reconstructor) CTR (Controller) and CODE (Corrective Optics Drive Electronics) and a number of auxiliary tasks to perform other functions like calibration or parameters tracking.

2.4 Data Distribution

A key component of the SPARTA architecture is certainly the data distribution. SPARTA uses DDS (Data Distribution Service) for 2 purposes: real time high-throughput data distribution and event notification.

The former is clearly the most important application. In MACAO and MAD this layer had been developed using a custom TCP/IP based protocol. SPARTA added the requirements of multiple clients on different machines all receiving the same stream, up to 80 MB/s. Multicast was an obvious solution to this problem and DDS provided us with a off-the-shelf implementation of a multicast protocol.

Figure 5 shows an example of SPARTA data distribution. The real-time box modules (in yellow) publish data using the custom point-to-point RTDFL protocol (TCP-based). Data are collected by a “data concentrator” that correlates packets coming from different modules of the real time box but sharing the same frame counter, builds a “super-packet” and publishes it to the cloud via DDS. Several clients are listening for this packet for different purposes: calibration, performance estimation, loop optimization, etc. Clients can be added transparently without affecting the rest of the system thanks to the multicast implementation

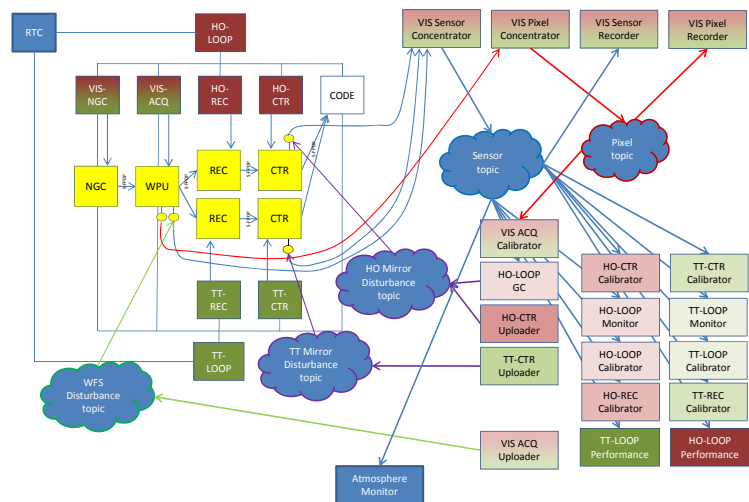


Figure 5: SPARTA Data distribution example

DDS is also used in SPARTA to replace the CORBA notification service. This has been done to experiment with the co-existence of CORBA and DDS and to show that events and logs can easily travel via DDS topics.

3. SPARTA LIGHT

SPARTA-Light is a simplified, scaled-down version of the SPARTA platform targeting smaller complexity systems. It has been created to re-use the SPARTA platform as much as possible at a significantly lower cost.

The aim of the SPARTA-Light project is to provide a standard platform to build small to medium complexity adaptive optics real-time computers. As well as SPARTA, SPARTA-Light is not an AO RTC, but provides the concept how to build one, a reusable infrastructure and a set of tools. As such, the SPARTA-Light platform is the combination of the items below:

- Hardware specifications defining the system building blocks
- A software framework providing functionality common to all AO systems – e.g. communication, configuration, storage, math, etc.
- Interface definitions:
 - External interfaces: connected to sensors, actuators, networks, etc.
 - Internal interfaces: interconnecting the different hardware and software components

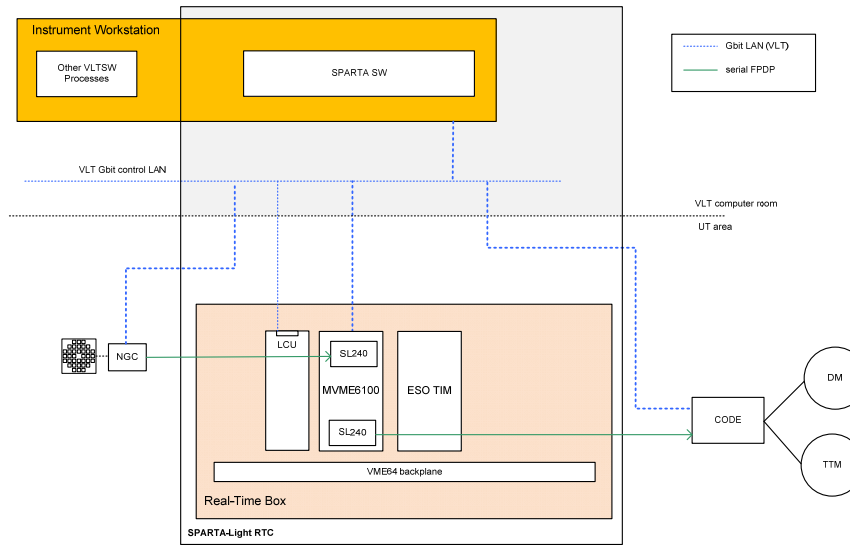


Figure 6: Example configuration of a SPARTA-Light system

A SPARTA-Light RTC consists of two main components (see Figure 6):

- A real-time SW processing pipeline known as the Real-Time Box
- An Instrument Workstation hosting standard VLT SW as well as SPARTA-specific SW

3.1 SPARTA Light Real Time Box

The Real-Time Box handles all tasks directly involved in the AO real-time control. It is based on a 19" VME system hosting at least one embedded computing board. The Real-Time Box continuously sends sensor and actuator data to the Instrument Workstation for further processing using the Gigabit Ethernet control LAN. All sensors and actuators are connected to the Real-Time Box via 2,5Gbit/s sFPDP fibre links.

The SPARTA-Light reference design integrates all major AO loop processing blocks in a single real-time processing board:

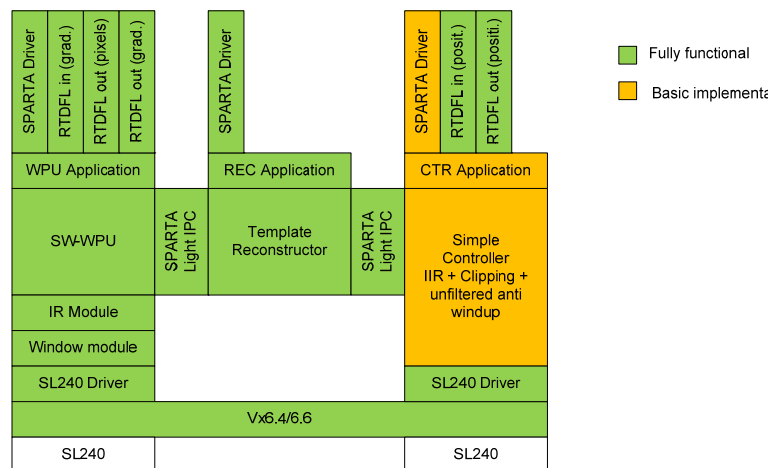


Figure 7: SPARTA Light Real Time Box Architecture

- ACQ: Acquisition and Wavefront Processing Stage, converting pixels to gradients
- REC: Wavefront Reconstruction Stage, projecting gradients to mirror space
- CTR: Actuator Control Stage, performing the control actions

The main SW components in the SPARTA-Light real-time processing board are presented in Figure 7: SPARTA Light Real Time Box Architecture following a hierarchical (i.e. layered) approach. Two main component types must be distinguished:

- The blocks in green are fully functional as provided by the SPARTA-Light Reference Implementation and do not require further specialization for use by the specific SPARTA-Light system. They share most of the code with the mother platform.
- The blocks in orange are to be replaced by specific components for each particular SPARTA-Light system. The SPARTA-Light Reference Implementation provides only a template implementation for them together with the specification of their API. It is the responsibility of each particular project using the platform to further develop/re-implement these components.

In the cases considered for SPARTA Light we expect that the reference implementation, which is derived directly from the SPARTA mother platform, will suffice for most of the blocks, with the notable exception of the controller. In fact SPARTA Light is intended to be used in the VLT Interferometer and therefore must be equipped with piston control algorithm as well as more sophisticated saturation management algorithms, which are not developed as part of the mother platform. The SPARTA Light IPC deserves a quick note: it is based on the mother platform RTDFL (Real Time Data Flow – Low level), which is a custom protocol resembling DDS. In fact one must define a topic that is sent across an RTDFL interface. This protocol is the same used to send the metrology to the supervisor via TCP/IP which is then further redistributed with DDS. The platform provides both a shared memory implementation as well as a TCP/IP one, with identical API. It is therefore easy to relocate one or more blocks on different machines therefore a configuration with 2 (or even 3) CPU boards can be used with SPARTA Light, where each board can run the ACQ or the REC or the CTR or a combination. Clearly at the expense of a higher latency, due to the communication over TCP/IP over a cross-over Ethernet cable.

3.2 SPARTA Light Supervisor

The SPARTA-Light platform reference design reuses the SPARTA High-Level SW components and deploys them on the Instrument Workstation together with the standard VLT SW, therefore using one single machine where the SPARTA mother platform used an Instrument Workstation to host the UI and an array of servers for the supervisor.

All the components of the SPARTA platform can be reused in SPARTA Light after customization to the size of the system (topic definition). Given the modularity of the supervisor architecture, additional components can be easily added to fulfill specific functions like piston management or saturation management, which is left to the particular project.

4. STATUS AND PERFORMANCE

SPARTA is designed to serve 2nd generation VLT instruments, in particular SPHERE, GALACSI and GRAAL. In addition SPARTA Light will serve a low-order system for the Auxiliary Telescopes (NAOMI) and probably GRAVITY as well (TBD). A fourth system serving the third focus of the AOF is under study.

Table 1 lists the systems of the main platform and their main parameters. The total complexity is computed as the product of the length of the gradient vector times the number of actuators of the mirrors times the frequency, without taking into account the latency.

Table 2 lists the portfolio for SPARTA Light, but in this case all the parameters in the table are TBC since those instruments are in early stage of definition/development.

Table 1: SPARTA main platform portfolio

System	Type	#s	DoF	#m	DoF	FRQ	Latency	Complexity
SPHERE	XAO	1	40x40	1	1320	1500	150μs	5 GMAC
GRAAL	GLAO	4	40x40	1	1170	1000	300μs	12 GMAC
GALACSI	LTAO	4	40x40	1	1170	1000	300μs	12 GMAC

Table 2: SPARTA Light portfolio

System	Type	#s	DoF	#m	DoF	FRQ	Latency	Complexity
NAOMI	SCAO	1	5x5	1	20	500	1ms	1 MMAC
GRAVITY	SCAO IR	1	9x9	1	60	700	1ms	5 MMAC

SPARTA is now in advance state of development. The first implementation, SPHERE, is scheduled to be completed by mid of 2011, but intermediate reduced-features/fully functional releases will be delivered, a first simple one being already done.

At the time of writing the major functions of the real time box have been completed and tested and all the critical components are in place. The real time FPGA+DSP pipeline has been completed and tested and features extremely low latency and almost no jitter. At the current stage of development, the SPARTA real time box for SPHERE/SAXO features about 300 μ s latency, 200 of which are spent in the controller part, which is implemented in a CPU. In fact most of the time spent there is to transfer data to and from the CPU. The control loop in the SPHERE/SAXO dimensions has been pushed up to 1.8 KHz.

At the cluster level the infrastructure features already all the critical components, from deployment tools, to CDMS, display tools, all base objects, data recorder. All 5 fundamental objects have been developed for 3 loops (HO, TT, DTTS). Auxiliary tasks are in development and in particular the calibration task is approaching completion. Atmospheric estimation and loop optimization are also in an advanced stage of development.

In the coming months the SPARTA team will complete all the features of the real-time box and will finalize the design of the co-ordination objects shown in Figure 4.

SPARTA Light, instead, has just started. As a sister project it will inherit most of the code base from the SPARTA mother platform. The core of the work on SPARTA Light, which will be performed in the next 24 months, will be the adaptation of the code of the real time box to run on a smaller system purely CPU-based.

5. CONCLUSIONS

In this paper we have briefly described the architecture of the Standard Platform for Adaptive optics Real-Time Applications (SPARTA), a development that ESO is undertaking to provide various high performance instruments with a common platform. We have shown the main components of the architecture and the technologies used to achieve the low latency and low jitter requirements of the various instruments in the SPARTA portfolio.

We have also introduced the SPARTA Light variation of the main platform, which is designed to address the need of smaller system at a lower cost.

We have also reported the current state of development of both projects, SPARTA being well advanced while SPARTA Light has just started.

REFERENCES

- [1] S.J. Goodsell, D. Geng, E. Fedrigo et al. "FPGA developments for the SPARTA project: Part 2", SPIE Conference Series 6272, (2006)
- [2] E. Fedrigo, R. Donaldson, C. Soenke, et al. "SPARTA: the ESO standard platform for adaptive optics real time applications", SPIE Conference Series 6272, (2006)
- [3] T. Fusco, C. Petit, G. Rousset et al, "Design of the extreme AO system for SPHERE, the planet finder instrument of the VLT", SPIE Conference Series 6272, (2006)
- [4] S. Strobele, R. Arsenault, R. Bacon et al, "The ESO Adaptive Optics Facility", SPIE Conference Series 6272, (2006)
- [5] J.L. Beuzit, M. Feldt, K. Dohlen et al., "SPHERE: a planet finder instrument for the VLT", SPIE Conference Series 7014, (2008)
- [6] R. Arsenault, P.Y. Madec, N. Hubin et al., "ESO adaptive optics facility", SPIE Conference Series 7015, (2006)