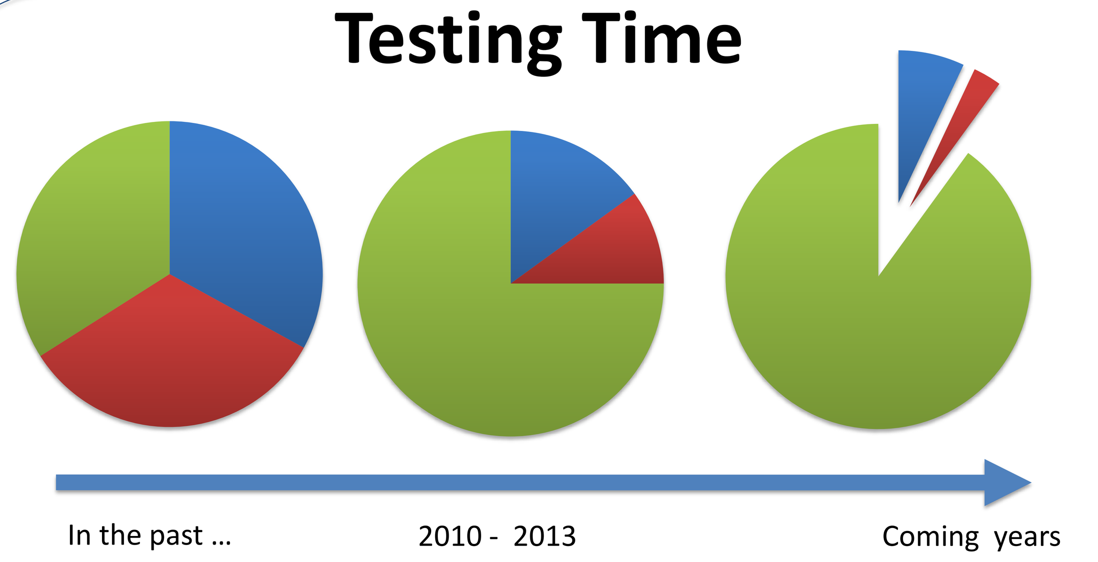
The Evolution of The Simulation Environment in ALMA

Tzu-Chiang Shen[a], Rubén Soto[a], Norman Sáez[a], Gastón Velez[a], Soledad Fuica[a], Tomás Staig[a], Nicolas Ovando[a], Jorge Ibsen[b]
[a]Atacama Large Millimeter/submillimeter Array, Av. Alonso de Córdova 3107, Santiago, Chile,
[b]European Southern Observatory, Av. Alonso de Córdova 3107, Santiago, Chile.

Abstract

The Atacama Large Millimeter /sub millimeter Array (ALMA) has entered into operation phase since 2014. This transition changed the priorities within the observatory, in which, most of the available time will be dedicated to science observations at the expense of technical time that software testing used to have available in abundance. The scarcity of the technical time surfaces one of the weakest points in the existent infrastructure available for software testing: the simulation environment of the ALMA software. The existent simulation focuses on the functionality aspect but not on the real operation scenarios with all the antennas. Therefore, scalability and performance problems introduced by new features or hidden in the current accepted software cannot be verified until the actual problem explodes during operation. Therefore, it was planned to design and implement a new simulation environment, which must be comparable, or at least, be representative of the production environment. In this paper we will review experiences gained and lessons learnt during the design and implementation of the new simulated environment.

Testing Time



Overview

The priorities have been changing in the observatory since entering into operations. This transition has lead to a reduction in the technical time both for software testing and engineering tasks such as maintenance and hardware integration in benefit of having more time for science operations, and this tendency will increase in the following year. The previous reasons have lead to realize one of the weakest areas in the existent infrastructure for software testing: The simulation environment of the ALMA software.

In ALMA, simulation capabilities were initially developed to satisfy Control and Correlator subsystem needs, supplying them with virtual devices to interface with the software components being developed. Later, additional simulation layers and capabilities were added, but focused on the functionality aspect instead of the real operation scenarios, leading to ignore or postpone awareness on scalability and performance problems introduced by new features or hidden in the current accepted software. The lack of a representative testing environment will seriously impact the efficiency of the ALMA incremental software release process.

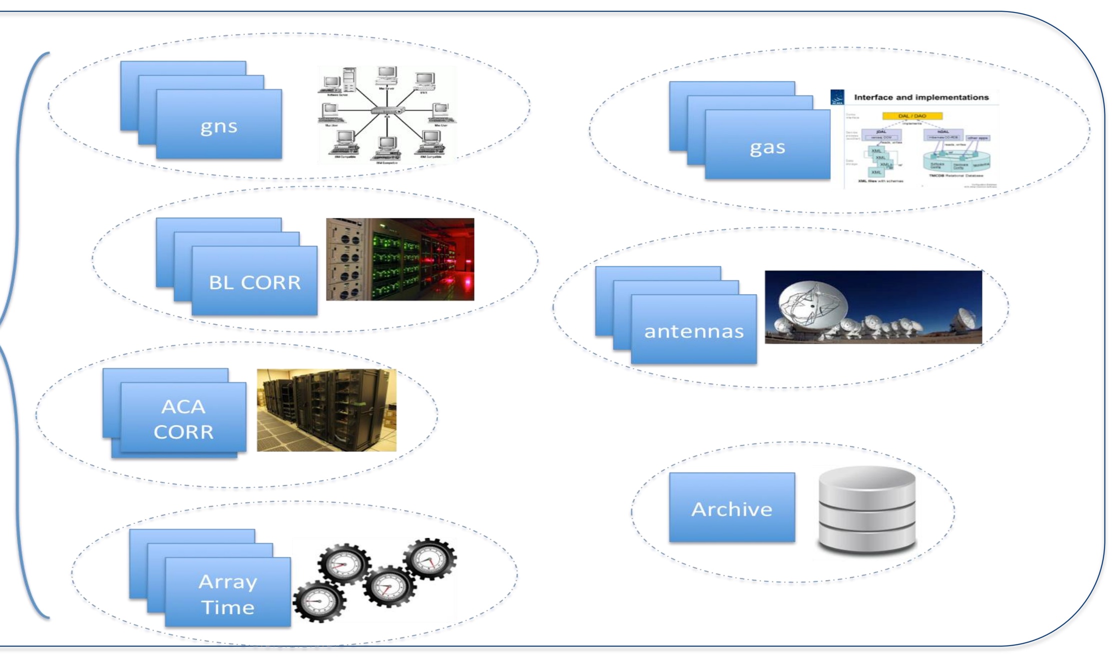
It was planned to design and implement a new simulation environment, which must be comparable, or at least representative of the production environment. Duplicating the production environment was not an option given the magnitude of the associated costs. As a consequence, adjustments in the current simulation architecture had to be introduced, taking special care in having a comparable simulation environment with regards to the production environment in terms of CPU load, network bandwidth throughput, memory usage, software configurations, etc.

The selected platform to provide computing power is based on blade technology of Cisco Unified Computing System (UCS). The new simulation platform will provide the required amount of time for testing purposes and, at same time, it will allow us to maximize the efficiency of the reduced technical time available in the production environment. This time will be dedicated only for the final validation of a new release and to test a small set of features that interact directly with hardware.

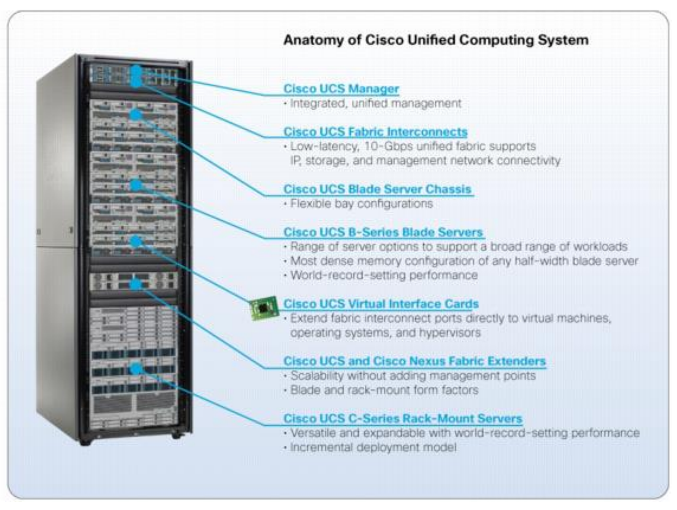
STEs

- STE
  - Standard Test Environment: STE is a set of computers, which controls the entire telescope using a distributed framework called ACS (ALMA Common Software). The testing environment model was later consolidated and used for production, yet the name remained for historical reasons
- Production STE
  - AOS : 66 antennas, BL Correlator, ACA Correlator, Central LO
- Verification STE
  - TFINT: 2 antennas, BL Correlator (2 antenna input), temporal Central LO
  - TFENG: 2 antennas, BL Correlator (2 antenna input), temporal Central LO shared with TFINT
  - TFOHG: 1 antenna
  - TFSD: 1 antenna, SDTR (Central LO pre production)

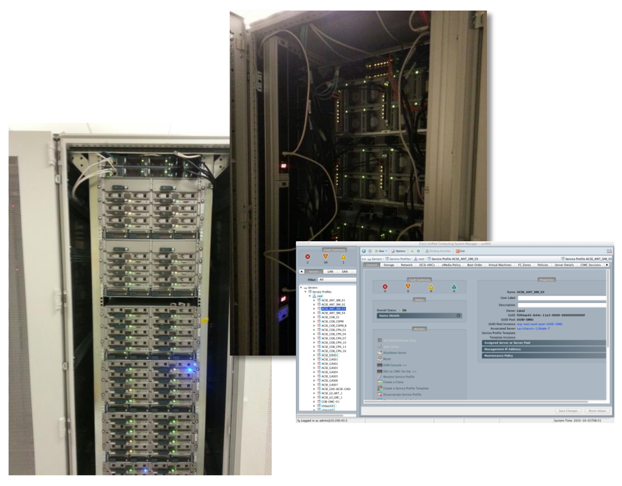
STE



Simulation



- NEW Simulated STE
  - ACSE : 66 antennas, BL Correlator, ACA Correlator, Central LO
  - AOS64: 25 antennas, BL Correlator, Central LO. Connection to production hardware is available.



item	Description	type	# Equipment in production	Consolidation factor	# Of required blade server
1	Antenna Bus Master, Central LO RT computers	abm	68	8	8.5
2	CentralLO RT computers, DMCs	lo-x, cob-dmc	7	8	0.875
3	CDP nodes	cob-cdpn	16	4	4
4	CDPMaster, CCC, COJ-CDPMaster, COJ-CC	cob-cdpm, cob-cc, coj-cdpm, coj-cc	4	2	2
5	ACA CDP nodes	coj-cdpn	32	4	8
6	General Network Servers	gns	2	1	2
7	General Application Servers	gas	6	1	6
				Total # of blades	31.375

Use Cases

Since the moment the new simulation environment was put into production it has been the key platform for the verification phase of the incremental software release process [7]. The most common testing's scenarios are (i) regression tests, (ii) scalability tests, (iii) new features/functionality tests and (iv) bug fixes testing.

The simulation platform has also been very useful to verify the performance of key components of the ALMA software. For example:

- Concurrent access to the TMCDB database: Stressing the TMCDB database has allowed to introduce improvements that reduced the subsystems initialization time and initialization of the observations.
- Bulk data transfer: Transfer load testing allows to find throughput and timeout problems early in the testing process.
- System start-up parallelization: Having a comparable number of devices and machines allows to test start-up and online scalability to prevent the introduction unreasonable delays during software releases.

It has also been very useful to find and fix long-standing integrity issues that are only seen after long observations, such as memory leaks and memory corruption issues, which has allowed to increase the continuous uptime of the system by maintaining a low resources usage and protecting from unexpected crashes.

The simulation environment have also been used to test new deployment strategies before going into production.

Future Work

Future work is based in the enrichment of the simulation behavior of the hardware devices, such as antennas pointing, correlators modes, etc. We expect to incorporate concepts such as model in the loop or hardware in the loop, which gives us the advantage to use exactly the same software than production, therefore helping to achieve a better coverage with our testing process.

It is also considered to use the idle time of this simulation environment to run automatic testing. Our goal is to support continuous software integration, as part of the ALMA software delivery process [7], and this environment will be perfect to execute nightly builds.

Conclusion

A new simulation environment was designed and implemented. It fulfilled entirely the defined requirements, specially being a representative testing environment of the production environment. After its introduction, the amount of technical time requested on the production environment for software testing has been reduced considerably.

The testing environment has the same network configuration and, following the same idea, the servers and simulated devices are deployed exactly in the same way as in production, allowing for the same configuration schemes and tools to be used in both places.

Blade servers have demonstrated to be an excellent alternative to provide computing power, which scales, taking account that the observatory's data center is located in the Atacama Desert where providing power and space is not something trivial. Cisco UCS, as a complete solution, has the advantage that non-additional network devices are required to procure in order to put the blade servers chassis into production.

It was important for this solution to be fully compatible with our existent network design. Internally, Cisco UCS's Fabric Interconnect switches provide high availability and redundancy by design; therefore our former in house mechanism implemented by using network bonding at the O.S. level is not necessary anymore. Finally, with the experience learnt in this area, it is planned to upgrade to production environment using the same technology in the next year.

REFERENCES

[1] T.C. Shen, R. Soto et al., "ALMA Operations Support Software and Infrastructure", Vol. 9149 91492D-6, proceedings of SPIE, SPIE' 12, Amsterdam, Netherlands (2012).
[2] A. Farris, R. Marson, J. Kern et al. "The ALMA Telescope Control System", Proceedings of ICALEPCS, ICALEPCS' 05, Geneva, Switzerland (2005).
[3] T.C. Shen, N. Ovando, et al, "Virtualization in network and servers infrastructure to support dynamic system reconfiguration in ALMA", Vol. 8451, 84511N, proceedings of SPIE, SPIE' 12, Amsterdam, Netherlands (2012).
[4] A. M. Chavan, B.E. Glendenning, J. Ibsen et al, "The last mile of the ALMA software development", Vol. 8451 84510Q-1, proceedings of SPIE, SPIE' 12, Amsterdam, Netherlands (2012).
[5] M. Mora, J. Avarias, A. Tejeda, et al, "ALMA software scalability experience with growing number of antennas", Vol. 8451, 84510W, proceedings of SPIE, SPIE' 12, Amsterdam, Netherlands (2012).
[6] Cisco Unified Computing System website: http://www.cisco.com
[7] R. Soto, T.C. Shen, N. Saez, and others, "ALMA release management: a practical approach", proceedings of ICALEPCS, ICALEPCS' 15, Sidney, Australia (2015).
[8] M. Mora, J. Ibsen, J. Kern, and others, "Hardware device simulation framework in ALMA control subsystem", Vol. 411, p.462, proceedings of ADASS, ADASS XVIII, San Francisco, USA (2008).

ACKNOWLEDGEMENT
The authors acknowledge the contribution of all the members of the ALMA Department of Computing to the success of this project.