Site Sonar -
A Flexible and Extensible Infrastructure Monitoring Tool for ALICE Computing Grid

Kalana Wijethunga[1,2,*], *Maksim* Storetvedt[2,**], *Costin* Grigoras[2,***], *Latchezar* Betev[2,****], *Maarten* Litmaath[2,†], *Gayashan* Amarasinghe[1,‡], and *Indika* Perera[1,§]

[1]University of Moratuwa, Bandaranayake Mawatha, 10400, Moratuwa, Sri Lanka
[2]CERN, Esplanade des Particules 1, 1217, Meyrin, Switzerland

Abstract. The ALICE experiment at the CERN Large Hadron Collider relies on a massive, distributed Computing Grid for its data processing. The ALICE Computing Grid is built by combining a large number of individual computing sites distributed globally. These Grid sites are maintained by different institutions across the world and contribute thousands of worker nodes possessing different capabilities and configurations. Developing software for Grid operations that works on all nodes while harnessing the maximum capabilities offered by any given Grid site is challenging without advance knowledge of what capabilities each site offers. Site Sonar is an architecture-independent Grid infrastructure monitoring framework developed by the ALICE Grid team to monitor the infrastructure capabilities and configurations of worker nodes at sites across the ALICE Grid without the need to contact local site administrators. Site Sonar is a highly flexible and extensible framework that offers infrastructure metric collection without local agent installations at Grid sites. This paper introduces the Site Sonar Grid infrastructure monitoring framework and reports significant findings acquired about the ALICE Computing Grid using Site Sonar.
Keywords - Grid computing, Grid monitoring, Grid infrastructure, infrastructure monitoring, Site Sonar

1 Introduction

ALICE (A Large Ion Collider Experiment)[1] is an experiment dedicated to heavy-ion physics at the Large Hadron Collider (LHC). The experiment produces tens of PB of new data every year[2] and requires a large amount of storage and CPU power for its computations. The experiment relies on the ALICE Computing Grid – a highly distributed computing Grid which which consists of 77 Grid sites around the world. The ALICE computing Grid has access to more than 150,000 computing cores, 350 petabytes of storage and on average

*e-mail: kalana.16@cse.mrt.ac.lk
**e-mail: mstoretv@cern.ch
***e-mail: Costin.Grigoras@cern.ch
****e-mail: Latchezar.Betev@cern.ch
†e-mail: Maarten.Litmaath@cern.ch
‡e-mail: gayashan@cse.mrt.ac.lk
§e-mail: indika@cse.mrt.ac.lk

runs more than 500,000 jobs daily. As a part of the upgrade process of the ALICE experiment for LHC Run 3, it was decided to aim for all jobs to run in Singularity/Apptainer containers[1] that provide job isolation as well as a uniform environment for the jobs to run in.

Each Grid site in the ALICE Computing Grid contributes its own resources to the Grid independently of other sites and different sites may have different hardware, software, configurations, security policies etc. The resulting Grid is largely heterogeneous, which makes it a challenge to develop software for the Grid which works on all Grid sites. Currently, software for running jobs on the Grid[3] is developed by assuming the worker nodes to have certain baseline capabilities that the vast majority of Grid sites can provide without difficulty. However, the ALICE Grid experts do not have explicit knowledge of how individual worker nodes in the Grid sites are configured and what capabilities they possess. On one hand, this may cause job failures when a feature that is not available in a worker node is presumed in developing software for the Grid. On the other hand, this limits utilizing the new features offered by Grid sites, as the Grid software developers are not aware of those features unless they are informed by Grid sites.

A new tool called "Site Sonar" has been developed by the ALICE Grid team as a solution to these problems. Site Sonar was initially used to collect information from Grid sites about their ability to support Singularity. It was later improved to become a flexible, extensible, and architecture-independent Grid infrastructure monitoring tool that is used to oversee the infrastructure information of worker nodes in ALICE Grid sites.

The specific contributions presented in this paper are:

- A methodology to identify the capabilities of individual worker nodes in a distributed computing Grid.
- A new Grid infrastructure monitoring tool which we refer to as Site Sonar that can be used to identify the capabilities of a computing Grid.
- Findings about the ALICE Computing Grid, acquired using Site Sonar.

2 Literature Review

GridICE[4] is a legacy Grid infrastructure monitoring system that was developed to monitor the Istituto Nazionale di Fisica Nucleare(INFN) Grid. The system consists of individual agents installed at each Grid site, which collect the infrastructure data and report to the central services daily. With GridICE, local site administrators can decide what should be exposed to the outside as monitoring data. However, this would require the ALICE Grid team to negotiate with each ALICE Grid site on the desired information to be exposed, making for an unnecessarily cumbersome operational model.

Paryavekshanam[5] is a Grid infrastructure monitoring system developed by Prasad et al to monitor the GARUDA[6] computing Grid in India. Paryavekshanam uses a pull model to collect Grid information. The pull model usually requires a large resource-intensive process to run the data collection continuously, whereas most of the Grid monitoring systems use a push model to send data to the central servers only when necessary. Site Sonar was initially developed with a pull model[7], but later moved to use a push model, as the data collection across thousands of nodes across the Grid is too resource-intensive and takes a long time to complete.

MonALISA[8] is a popular monitoring system that has been used to monitor the ALICE Grid since more than 15 years. It specializes in collecting real-time performance data rather than infrastructure data that does not change often. It focuses on fast changing parameters

[1] Singularity was renamed as Apptainer along the course of the research

like CPU usage, latency, network round trip time etc. Each monitored parameter is typically reported once per minute. MonALISA cannot be used achieve the purpose of this paper because MonALISA is focused on reporting job data rather than infrastructure data. In addition, MonALISA analyzes the variation of a specific parameter across time whereas we are looking to analyze the variation of a specific parameter across worker nodes.

MONIT[9] is a monitoring framework developed by CERN IT to replace the LEMON[10] framework for CERN Data Center infrastructure monitoring as well as the Worldwide LHC Computing Grid (WLCG) Dashboards[11] which were used to monitor activities on the WLCG until a few years ago. MONIT collects high frequency data about the health of CERN Data Center hosts and their services use agent installations on those hosts. MONIT is focused on collecting information on the health of hosts and services, database activity, file transfer throughput etc. rather than infrastructure information that affects the running of jobs on the Grid, which prevents us from using MONIT to achieve the goals of this paper.

3 Site Sonar Architecture

Site Sonar is a new Grid infrastructure monitoring tool developed to monitor the ALICE Grid. Site Sonar collects information about the infrastructure and configuration of the worker nodes in the ALICE Grid, aggregates the data, and visualizes the results in monitoring dashboards with flexible filtering options. Site Sonar is set apart from the existing systems mainly by its high flexibility and extensibility. Site Sonar allows collection of any type of information from the worker nodes, from text strings up to complex JSON objects, and provides the ability to change or add any monitoring probe without having to make any changes to the Site Sonar framework. Additionally, it facilitates creating no-code visualization of the collected data with the help of an ELK(Elasticsearch[2], Logstash[3], Kibana[4]) stack. The architecture of Site Sonar is shown in Figure 1, which is explained in detail in the following sections.

3.1 Probe

The basic unit of Site Sonar is called a "Probe." A probe is a program that queries the worker node it runs on for some specific information and outputs the result as a JSON object. In practice, each probe is a shell script. Site Sonar consists of a number of such probes and more probes can be easily added with no code changes. Each probe can be configured with a TTL (Time To Live) after which its results are considered obsolete and need to be collected again.

3.2 Sonar

The "Sonar" is the main component used for data collection in Site Sonar. When a worker node starts running an ALICE Grid job, the latter will instantiate a number of "Job Agent" processes, each of which is responsible for running a payload task received from the ALICE Central Services. Before a Job Agent starts the execution of the payload, it runs the Sonar which calls the Central Services to identify which probes need to be run on the current node. This is done to avoid sending results from probes that already had results reported for the given worker node within the TTL values of those probes.

More than 280,000 such metrics are reported every day to the Central Services by Sonars running across the ALICE Grid, with an average data ingestion rate of 3.2 Hz.

[2] https://www.elastic.co/elasticsearch
[3] https://www.elastic.co/logstash
[4] https://www.elastic.co/kibana

3.3 Central Services

The Central Services act as the controller for data inflow and outflow to the Site Sonar tool. They control which probes should be run, what their TTLs should be and which parts of the collected data should be visible to the users.

Central Services store the data reported by Sonars directly in a database for long term persistence. A daily job then aggregates the collected metrics per node and prepares JSON documents of metrics reported per node. These JSON documents are then uploaded to Elasticsearch through Logstash.

Site Sonar injects more than 7,000 documents into the Site Sonar ELK stack every day, with each document containing the more than 25 metrics collected on a worker node in the ALICE Grid. The data is visualized through preconfigured Kibana dashboards for scrutiny by the ALICE Grid team. In Site Sonar we use an ELK stack due to its fast response time, complex querying capability, presence of visualization tools out-of-the-box and the ability to create no-code visualization.

In summary, Site Sonar consists of 2 main components: the data collection framework and the data analysis framework. The data collection framework runs inside the Job Agent, which is responsible for executing the job on the worker node. Before the Job Agent runs the payload, it calls the Sonar, which checks the probes that need to run on the given node, then invokes those probes one by one. The resulting output is sent to Central Services by the Sonar for long-term persistence.

The data analysis framework is a separate component that is based on an ELK stack that ingests the data reported by the data collection framework, and allow users to create analysis and visualization of that data without the need to write code.

4 Features

A computing Grid is inherently heterogeneous owing to the fact that it is a collection of independent Grid sites. The possibility of knowing as much information as possible about each Grid site is helpful on multiple fronts: to ensure that the developed Grid software is compatible with each Grid site; to identify new capabilities offered by the Grid sites that can be utilized to improve the functionalities and the performance of Grid software; and to identify incorrectly configured nodes which cause job failures.

With the introduction of Site Sonar, the ALICE Grid team now benefits from a lot of interesting information that they were not aware of before. Special features of Site Sonar, which make it stand out from other systems are explained in this section.

4.1 Advanced Data Collection Capabilities

Existing monitoring systems focus on collecting and reporting a structured set of metrics which are often numeric values or short strings. This limits the amount of information we can collect from worker nodes, as all the information has to be preprocessed on the node and only the exact information that is foreseen is then sent to the monitoring system.

Site Sonar follows a data-first approach to this problem by allowing the Grid team to collect any kind of data desired from the worker node and then process it later at the time of analysis or visualization. In other systems, Grid administrators or operators need to be sure about what data needs to be collected and how it should be processed before the data collection starts. Important information can easily be missed in this approach which also make it hard to identify anomalies beyond the absence of some expected functionality: as Grid operators cannot know from the outset what kinds of anomalies might be present in a largely heterogeneous Grid, monitoring probes will only be able to check if known functionality is in working order or not. Site Sonar provides the flexibility to the Grid team to collect any data of interest first, look at the collected data, do a preliminary analysis first and finally create visualizations or arrive at conclusions using a selected subset of the data, unlike how other systems work.

4.2 Flexibility

While other systems only support structured data, Site Sonar provides unstructured data collection capabilities. To enable this, Site Sonar uses JSON objects as the model for collecting reported metrics. Using JSON allows the Grid team to collect information of any data type, including string, integer, boolean, arrays and even nested objects. It also provides the possibility to change the structure of the data to be collected whenever required, as Site Sonar does not enforce a schema on the resulting objects.

Site Sonar also provides a TTL for each test result, which allows the Grid team to collect different data at different frequencies. For example, rarely changing values like operating system versions can be collected once a week, while more dynamic values like installed software versions can be collected daily.

4.3 Extensibility

Other Grid monitoring systems are designed to monitor a set of predefined metrics and report them periodically. Considerable work has to be done to add new data collection probes to these systems. In most systems, the user has to write code to add a new probe as well as develop a new interface for visualizing the data collected by that probe. This usually requires deploying a new software version across the Grid and hence probes cannot be changed easily on demand.

Site Sonar has had a different architecture from the outset to circumvent the problem. Site Sonar probes are added to a specific directory in CVMFS[12] which is a distributed, read-only file system mounted on all the nodes that are part of the ALICE Grid. The Central Services instruct Sonar which of the probes that are present in this directory should be run on the relevant node. As a result, there is no need to change any code in Site Sonar or do any software releases to add a new probe. Adding the new probe in the relevant directory of CVMFS and adding a single entry in the database to enable the probe are enough to collect a new metric through Site Sonar, as shown in Figure 2.

4.4 Powerful visualizations

With the support for collecting any information of interest rather than predefined metrics, it is necessary to provide a powerful data analysis and visualization framework which can be used to process and analyze the data efficiently and selectively on demand. Other systems only provide a set of basic predefined filters and visualizations to analyze the collected data. If a new visualization is desired, the code for the monitoring system has to be changed to add it.

This is addressed in Site Sonar by using the ELK stack. Usage of the ELK stack for the analysis and visualization of data in Site Sonar helps to increase the flexibility offered to Site Sonar users. Kibana allows the creation of data visualizations without the need for writing code. This allows the user to freely experiment with the data as it permits creating multiple types of visualizations in different formats with different filters, leading to a rich analysis palette.

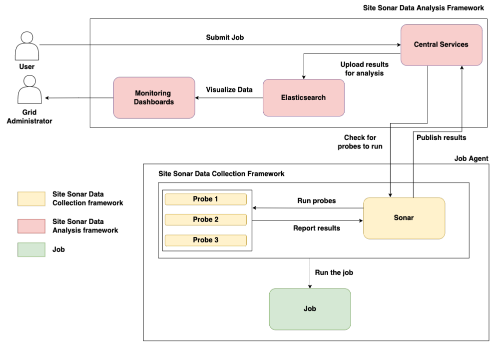

Figure 1: Site Sonar Architecture

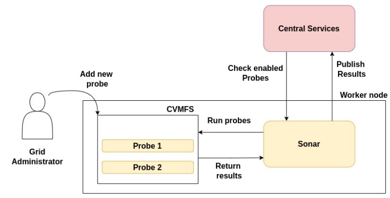

Figure 2: Adding new probes to Site Sonar

ELK also provides important features like multiple indexing options, data filters and processors, and multiple visualization options out of the box. Therefore we believe it is best to do our analysis and visualizations using the ELK stack rather than developing a new in-house data analysis framework. With the help of ELK, Site Sonar provides its users with the ability to add new probes to monitor the Grid and define how they should be visualized without the need to write any code beyond the probes themselves.

5 Results and Analysis

Currently Site Sonar monitors a large number of infrastructure metrics in the ALICE Grid. These metrics range from basic information like the Operating system, CPU information (no. of cores, CPU architecture, etc.), memory information (total RAM size, swap size, etc.) up to more complex information like open file limits, virtualization information (hypervisor information), or in which ways containers are supported etc. Site Sonar is capable of collecting any information that is available to be read by a unprivileged user in a worker node and even execute a complex function like a benchmarking script to measure and report the performance of a worker node.

Following is an analysis of Operating systems and Singularity support in the ALICE Grid which are few of the many metrics that are collected from the ALICE Grid worker nodes.

5.1 Operating System Version Distribution

Figure 3 shows one of the monitoring dashboards of Site Sonar presenting the operating system version distribution of worker nodes in the ALICE Grid on 2023-04-22. Meta information like how many sites have reported the information and how many nodes have reported data is shown in the top charts, and the distribution of the data is shown in the bottom charts of the dashboard. Other dashboards follow a similar structure with the ability to customize the view and add new visualizations on demand without any code changes. Dashboards can be used to easily filter out data by sites and/or attributes and create visualizations as well as summaries.

5.2 Singularity Support

The initial goal of Site Sonar was to check which sites do not support Singularity on their worker nodes, as it was required for introducing job isolation[13] for LHC Run 3. On a

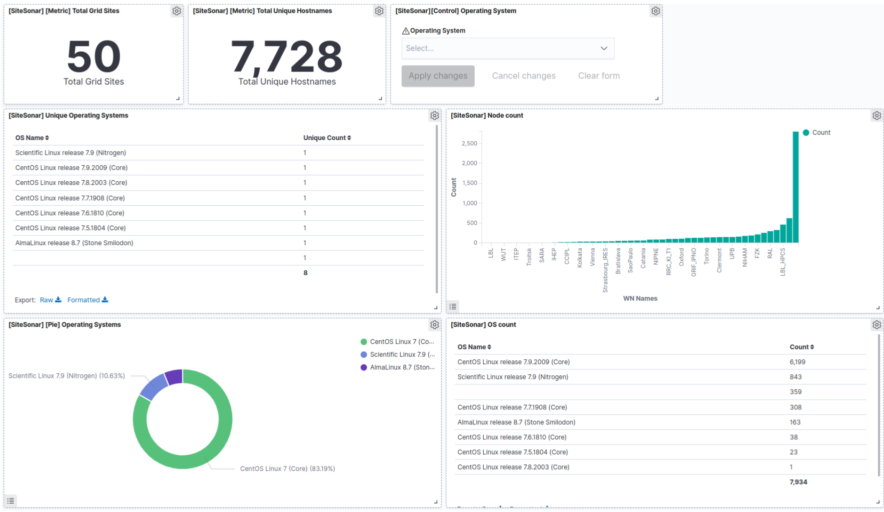

Figure 3: Operating System Monitoring Dashboard

number of sites, it was not supported via a local, privileged installation, or via the unprivileged installation provided on CVMFS by the ALICE Grid team. Contacting the respective site admins led to Singularity/Apptainer being supported at almost all sites. Where it is not supported, JAliEn runs its jobs without the benefits of containers. Site Sonar's "Singularity Support Dashboard" showed the percentage of nodes supporting Singularity/Apptainer either through local installation or CVMFS to be 96% on 2023-07-19.

5.3 Analysis

So far, the use of Site Sonar has led to multiple interesting discoveries about the ALICE Grid. This section presents a few of these discoveries.

5.3.1 Sites running CentOS 6

CentOS 6 does not by default come with sufficient support for running Singularity. Hence, when checking whether the ALICE Grid is ready to support Singularity, one of the essential requirements was to check if all sites are running CentOS 7 or higher. With Site Sonar, we easily noticed that a few sites were still running CentOS 6. This information helped us narrow down those sites and get them upgraded to CentOS 7.

5.3.2 Reusing hostnames on different nodes

For logging and monitoring purposes, each worker node on the grid should have a unique identifier. The worker node hostname usually is sufficient to play that role, but through Site Sonar it was found that some sites have such high turnover rates of hostnames that Site Sonar results for those sites then eclipsed the results of much bigger sites, at the same time suggesting that the ALICE Grid was bigger than it actually is. After detecting these cases through Site Sonar, the affected sites were requested to define an extra environment variable that identifies the underlying physical machine and to let Site Sonar use that variable instead of the hostname.

6 Conclusions

Site Sonar has proven to be a useful tool to the ALICE Grid team so far. It has helped the team ensure that the Grid is ready for LHC Run 3 and to discover various "hidden" capabilities of the Grid. The ability to collect any information from the worker nodes and the ability to change monitoring probes on the go have allowed Site Sonar to be occasionally used as a debugging tool for the Grid as well. It is used in research work on the ALICE Grid to study how worker node capabilities may be taken advantage of across the Grid or at selected sites. With the flexibility and extensibility of Site Sonar, we believe it can easily be used as a multi-purpose tool to monitor the Grid and analyze its capabilities, which in turn will help prevent faults and allow jobs to take better advantage of features offered by the sites.

References

[1] K. Aamodt, A.A. Quintana, R. Achenbach, S. Acounis, D. Adamová, C. Adler, M. Aggarwal, F. Agnese, G.A. Rinella, Z. Ahammed et al., Journal of Instrumentation **3**, S08002 (2008)

[2] S. Pumma, W.c. Feng, P. Phunchongharn, S. Chapeland, T. Achalakul, Future Generation Computer Systems **72**, 65 (2017)

[3] M.M. Pedreira, C. Grigoras, V. Yurchenko, *JAliEn: the new ALICE high-performance and high-scalability Grid framework*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 03037

[4] S. Andreozzi, C. Aiftimiei, G. Cuscela, S. Dal Pra, G. Donvito, V. Dudhalkar, S. Fantinel, E. Fattibene, G. Maggi, G. Misurelli et al., *Next steps in the evolution of gridice: a monitoring tool for grid systems*, in *Journal of Physics: Conference Series* (IOP Publishing, 2008), Vol. 119, p. 062010

[5] D.H. Karuna, N. Mangala, B. Prahlada Rao, N. Mohan Ram, *Paryavekshanam: a status monitoring tool for Indian grid GARUDA*, in *24th NORDUnet2008 Conference–The Biosphere of Grids and Networks, Espoo, Finland* (2008), pp. 9–11

[6] B. Prahlada Rao, S. Ramakrishnan, M. Raja Gopalan, C. Subrata, N. Mangala, R. Sridharan, Computer Science-Research and Development **23**, 283 (2009)

[7] E.B. Sandvik, Master's thesis, The University of Bergen (2021)

[8] I. Legrand, C. Cirstoiu, C. Grigoras, R. Voicu, M. Toarta, C. Dobre, H. Newman (2005)

[9] A. Aimar, A.A. Corman, P. Andrade, J.D. Fernandez, B.G. Bear, E. Karavakis, D.M. Kulikowski, L. Magnoni, *MONIT: monitoring the CERN data centres and the WLCG infrastructure*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 08031

[10] M. Babik, I. Fedorko, N. Hook, H.T. Lansdale, D. Lenkes, M. Siket, D. Waldron, *LEMON-LHC era monitoring for large-scale infrastructures*, in *Journal of Physics: Conference Series* (IOP Publishing, 2011), Vol. 331, p. 052025

[11] J. Andreeva, M. Boehm, B. Gaidioz, E. Karavakis, L. Kokoszkiewicz, E. Lanciotti, G. Maier, W. Ollivier, R. Rocha, P. Saiz et al., Journal of Grid Computing **8**, 323 (2010)

[12] C. Aguado Sanchez, J. Blomer, P. Buncic, L. Franco, S. Klemer, P. Mato, XII Advanced Computing and Analysis Techniques in Physics Research p. 52 (2008)

[13] M. Storetvedt, L. Betev, H. Helstrup, K.F. Hetland, B. Kileng, *Running ALICE Grid Jobs in Containers A new approach to job execution for the next generation ALICE Grid framework*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 07052