Evolution and deployments of MCAS, a Metrics Correlation and Analysis Service

Ted Hesselroth , Gabriele Garzoglio, Tanya Levshina, Parag Mhashilkar, Andrew Baranovski

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510-5011

E-mail: {tdh, garzoglio, tlevshin, parag}@fnal.gov, abaranovskyi@gmail.com

Abstract. The Metrics Correlation and Analysis Service (MCAS) is a web-based REST service that provides an integral solution to warehousing, querying, transforming, and displaying of disjoint metrics. These metrics can be generated by common distributed middle-ware or user services. The solution implements a well defined model for organizing information processing and display. The core of the software infrastructure is the MuleESB data transformation engine, which implements a graph-based model for connecting data analysis, summarization, and reduction rules. The model is completed by a python API which maps the results of the data transformation workflow into a dashboard presentation, using the layout of a few well defined UI window elements. This paper describes the MCAS service and several applications that are using the MCAS infrastructure. These are a time series visualization toolkit for the MINOS experiment at Fermilab, a status analysis and monitoring workflow for the dCache storage system, and a demonstration portal solution for metrics integration for the CMS experiment at CERN. Finally, specific visualization forms and their use cases are described.

1. Operational Challenges in Grid Monitoring and Troubleshooting

In recent years, the Grid computing paradigm[1] has been adopted by a wide variety of communities. Virtual Organizations and computing sites continue to join the Grid. Because all participants have their own administrative boundary, each of them has their own requirements for the types of equipment to purchase, and the kinds of tools and services to be provided. Thus, system monitoring information varies from site-to-site, and from experiment-to-experiment. Meanwhile, the complexity of Grid workflow activities and their associated software stacks inevitably involves multiple organizations, ownership, and deployment domains. Traditionally, operational data are generated using legacy utilities and read through independently created web pages. An overall picture of the status of the system is difficult to obtain.

The variety and the amount of information provided by these distributed services, such as monitoring, discovery, troubleshooting, accounting, auditing, etc. is becoming increasingly difficult to manage in a coherent fashion. Furthermore, due to the disjoint nature of all these data sources, aggregating, transforming, and displaying this distributed information is particularly challenging. This makes for a difficult environment in which to perform grid operations, i.e. troubleshooting, incident investigation, trend spotting, and alarming. In this setting, common important tasks such as the correlation and

display of metrics and debugging information are challenged by the informational entropy inherent to independently maintained and operated software components.
In general, operational problems cannot be easily visualized and assimilated using any one particular display method. As a part of the decision-making process, the experts employ a variety of tools such as Nagios[2] or Zabbix[3] to report and display data patterns that pinpoint incidents of already known problems. This variety is often built on technologies that incorporate incompatible input/output data such as XML, CSV, or serialized objects, which cannot be used within the same context. These incompatibilities restrict the ability to perform analysis across data domains.

2. The MCAS Strategy

To address the challenges described in the previous section, Fermilab has initiated the Metrics Correlation and Analysis Service (MCAS) [4] project. MCAS allows the collection of operational data, transforming it into a common XML format which can be stored in a database. Visualization, correlation, and analysis of data from different sources can then be obtained by querying the database and displaying the data through various javascript portlets. A portlet server gives a unified, customizable view of the system state. Since the database stores operational data over a period of time, data mining techniques can be used for historical analyses of events. Thus, MCAS supports common standards in data presentation, representation, and manipulation with which users can interact.

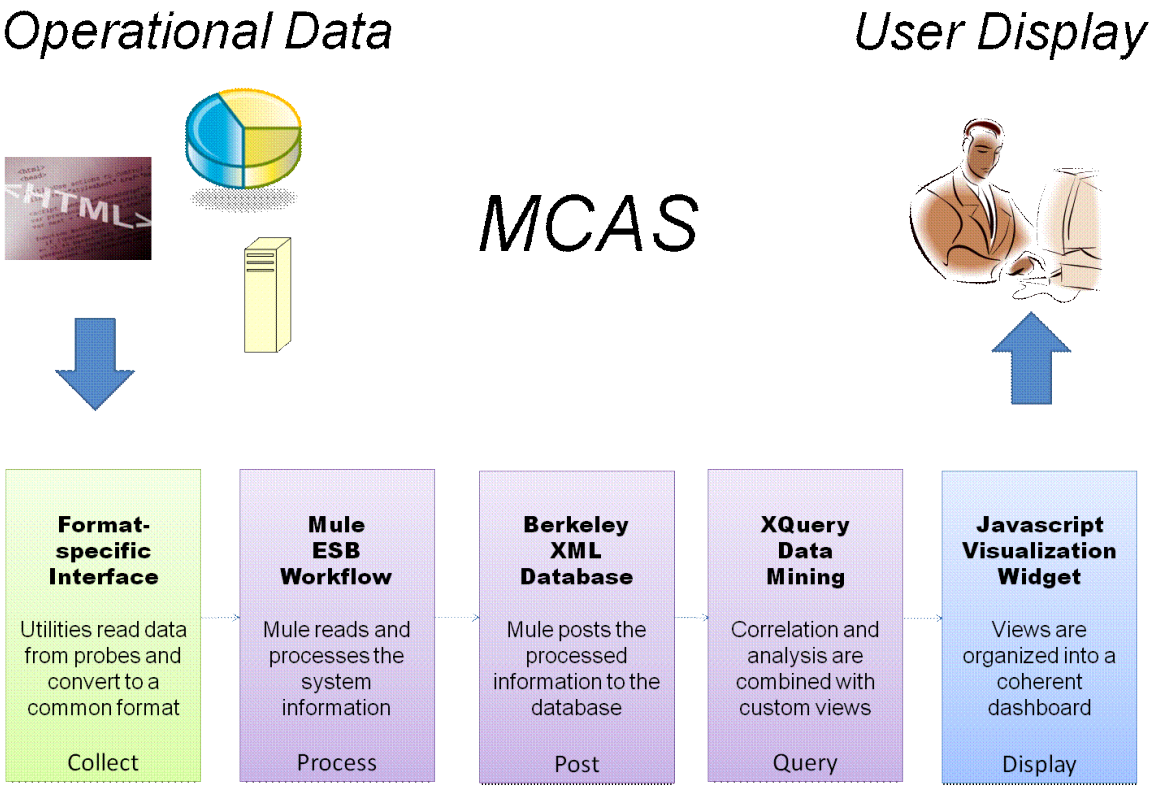

Figure 1. MCAS system diagram, showing processing steps from the collection of operational data to its graphical display for the user. Operational data is collected from probes, administrative interfaces, web pages, etc. At the user end, the operator or system administrator views a unified, coherent, customizable display of system state and history.

2.1. Data Collection

Data collection components provide an integration layer between MCAS and a site resource via a collection of utilities with format-specific interfaces. The interfaces process raw data provided by various tools using their specific formats and protocols, and output the collected data using a unified data format. Possible formats for input data include HTML, XML, Round-Robin Database (RRD), RDBMS/SQL, simple text, and image files. The output format is XML, with select schema provided by MCAS, depending on the data type.

2.2. MCAS Workflows

MCAS uses a publicly-available Mule[5] version to provide endpoints and perform workflows. Mule provides a lightweight Enterprise Service Bus architecture which connects input and output endpoints through customizable event-driven processing. Data can be either pushed or pulled into the Mule instance of MCAS. In the push model, probes periodically post system information to a service endpoint. Mule processes the data in a workflow, for example, by converting it to XML, converting the schema, or aggregating data from multiple sources. In the pull model, operational data may be acquired through periodic queries performed by Mule, or triggered from user actions in the graphical interface.

The transformations that occur in workflows are based on models that are specific to each operational system. MCAS provides tools to manage data source and workflow characteristics. In particular, the characteristics of the workflow, including its dependencies, are specified as parameters in python objects. The workflow configuration tool maps these objects into runnable Mule workflow configurations.

2.3. Data Warehousing

Workflow components are available to store transformed data. Historical or aggregated data can be queried and retrieved for analysis or visualization purposes. For example, users may want to combine different types of data and display them together, or examine the historical trend of a metric. The data mining components allow user to define and perform such operations. Depending on the specific technology used by the data storage components, different mining mechanisms may be used. The MCAS project has chosen the Berkeley XML Database[6] for persistence of the XML data. A Round Robin Database (RRD)[7] is also available. Queries to the data warehouse can be made using XQuery or XPath for the former, or RRD commands for the latter. Further processing of query results may also be done in a workflow context using data analysis tools such as R.

For the Berkeley XML Database persistence, automatic rotation of the database files is built into the software. When a database reaches a specified size or number of entries, a new instance of it is created. Previous data from rotated versions is still transparently accessible. This allows gradual deletion or archiving of the oldest, non-used data without affecting the functionality of the software. Automatic backup is also supported.

2.4. Rendering and Visualization

Visually displaying data together is one of the simplest ways to cross-analyze the data. MCAS enables a unified display of data by aggregating independently designed and managed frames in a single web browser window. Each frame renders a different aspect of the input data. Not only does this approach decouple development and testing of the display components, but it also allows users to choose and change the presentation layout along with configuration details of each component, in a way that best represents the state of the system.

Each visual frame can be autonomously designed with some unique perspective of a particular system aspect. A collection of different frames, describing different perspective of the system, can be put together to form a dashboard. The dashboard view reflects a perspective of the states of the entire system at a given time. Each visual frame is independently rendered using information provided by the data mining components.

Rendering these visual frames flexibly requires a Content Management System. MCAS uses JBoss with portlets[8] to provide platform for visualization. Users can create a customized dashboard by inserting different visual frames into the dashboard. This capability provides a flexible way of visualize, compare and contrast data. Figure 2 shows an example of the use of JBoss.

Figure 2. Screen shot of demonstration Fermilab CMS Tier 1 monitoring as deployed in JBoss.

3. Applications

MCAS has been employed to monitor several experiments at Fermilab, either for production or demonstration purposes.

3.1. Minos

Minos[9] is one of the neutrino experiments at Fermilab from the High Energy Physics intensity frontier. It uses MCAS to display the principal elements of their computing production system. Information is organized in tabs on the dashboard dedicated to the major architectural components: Computing Cluster, Grid Computing Resources, Storage System, Local Disk. These pages display plots that summarize the status of the various components. Minos also uses a summary page to provide a status "at-a-glance" of the entire computing infrastructure. The page displays aggregate plots from the information already available in detail on the other pages.

3.2. DZero

DZero[10] is one of the "RunII" experiments on the Fermilab Tevatron accelerator. MCAS shows relevant metrics for their computing infrastructure for both Monte Carlo production and data processing. The views are also organized in separate pages ("tabs"), displaying statistics of the production and status of the system.

3.3. CMS

Fermilab is a Tier 1 site for the CMS[11] experiment of the Large Hadron Collider . Proof-of-concept MCAS work (shown in Figure 2, above) demonstrated displays for the Tier 1 computing infrastructure. In particular, the bargraph widget of MCAS shows an immediate view on the number of functioning data transport doors of their local storage. Alarm conditions based on the number of non-functioning nodes is shown in red. On the same page, there is an aggregated a view of several key nodes, each displaying a Ganglia plot pulled from the individual web pages of each system.

4. Specific visualizations and their use cases

In this section we focus on the presentation layer, and show some of the variety of display forms.

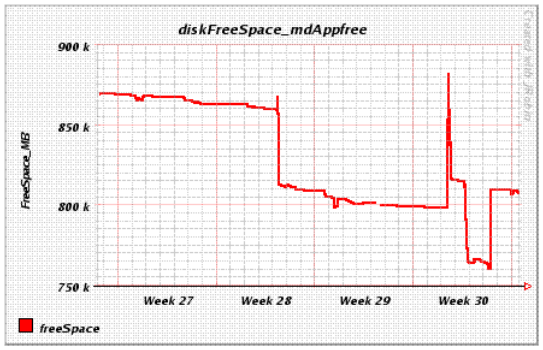

Figure 3. RRD plot example.

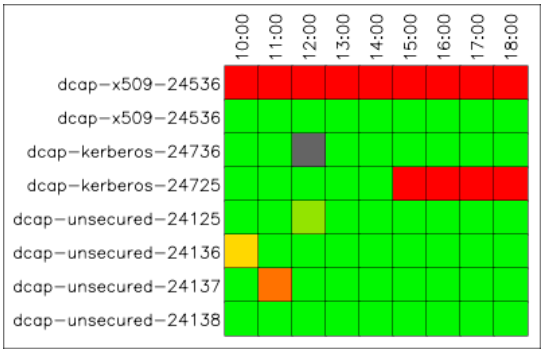

Figure 4. Heatmap example

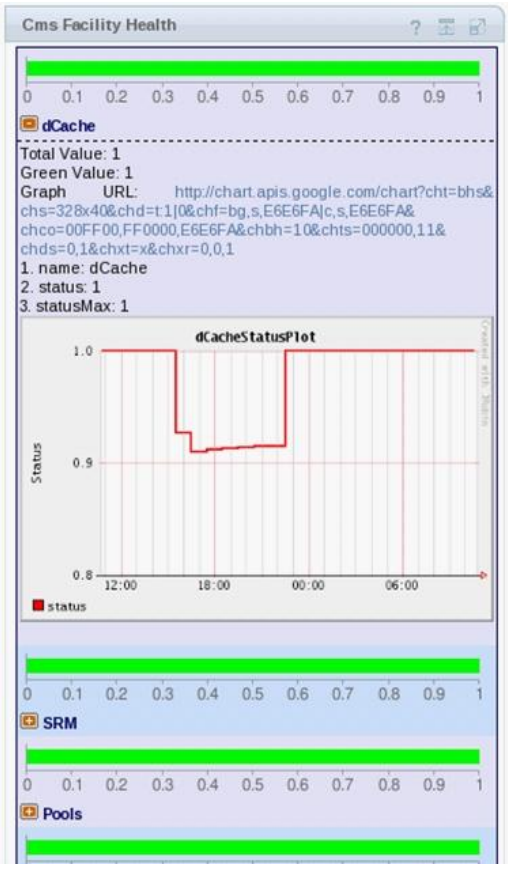

Figure 5. Bargraph example.

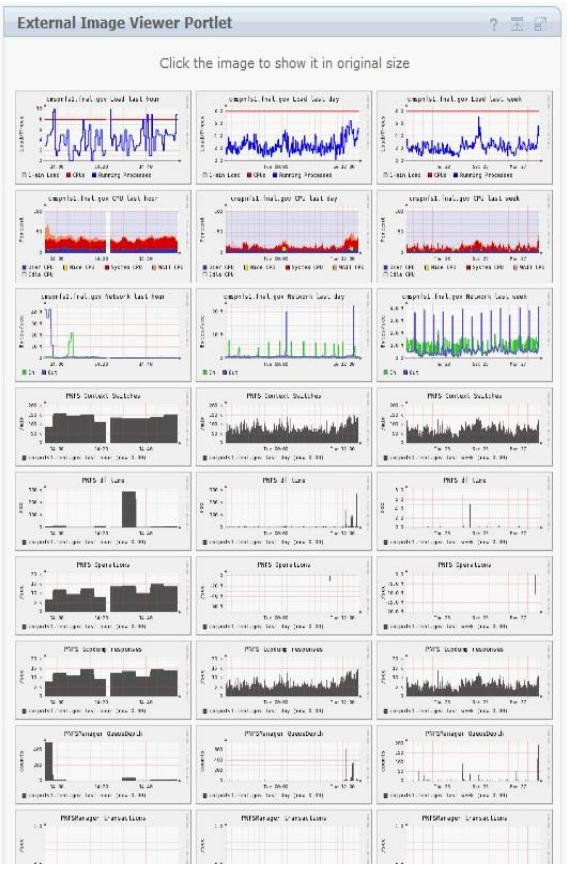

Figure 6. Image Viewer example.

4.1. RRD Plots

MCAS supports storing information in a Round Robin Database (RRD). Internally, MCAS uses the JRobin[12] implementation of RRD. In the push model, probes periodically post data in RRD format to the RRD service endpoint provided by Mule. Mule then processes the data in a workflow, storing it in one or more RRDs. In the pull model, operational data can be retrieved in RRD or XML format using appropriate queries. Users can create complex workflows that store the data in RRD databases and pull the data to create plots that are displayed using the Image Viewer portlet. See Figure 3.

4.2. Heatmaps

In this use case, the administrators of public dCache running at Fermilab would like to include in the dashboard the health status of gridftp doors over time. MCAS can display this information as a heatmap, where the color of each bin represents the status of a particular door during some period of time, e.g., the percentage of the time the door was up or down. The BioHeatMap javascript component[13] from Institute for Systems Biology is used to render the heatmap. See Figure 4.

4.3. Bargraphs

Storage administrators want to see the current health of multiple cells run by dCache. MCAS aggregates and correlates cell status then displays it as a bar graph. The bar graph shows the number of up/down gridftp and dcap doors as well as the number of up/down pools, aggregated by pool group. This allows the end user to quickly make generalizations about the data. See Figure 5.

4.4. Image Viewer

The administrators of the Minos experiment collect multiple graphs of various kinds (the number of running/idle jobs per user, read data rates on bluearc over time, etc). The MCAS image viewer portlet allows to organizing and displaying multiple graphs on one page as thumbnails. An end-user can enlarge a graph by clicking on it. See Figure 6.

5. Acknowledgments

This work was supported by the Fermi National Accelerator Laboratory. Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy. Fermilab is partially funded by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

References

[1] Anatomy of a Grid: Enabling Scalable Virtual Organizations. I. Foster, C. Kesselman, and T. Steven. Apr. 2001, International Journal of Super Computing Applications. 15(3):205–220.
[2] Nagios web site. http://www.nagios.org.
[3] Zabbix web site. http://www.zabbix.com.
[4] MCAS Project home page. http:/www.fnal.gov/docs/products/mcas.
[5] Mule web site. http://www.mulesoft.org.
[6] Berkely XML database web site. http://www.oracle.com/us/products/database/berkeley-db.
[7] RRD home page. http://www.mrtg.org/rrdtool.
[8] JBoss Portlet Container home page. http://jboss.org/portletcontainer.
[9] NuMI and MINOS web site. http://www-numi.fnal.gov.
[10] DZero web site. http://www-d0.fnal.gov.
[11] CMS web site. http://cms.web.cern.ch.
[12] JRobin web site. http://www.jrobin.org.
[13] BioHeatMap http://informatics.systemsbiology.net/visualizations/heatmap/bioheatmap.html