

УДК: 004.45

## Computational task tracking complex in the scientific project informational support system

N. V. Yuzhanin, Yu. A. Tipikin, I. G. Gankevich, V. I. Zolotarev

Saint Petersburg State University, University ave. 35, St. Petersburg, Peterhof, 198504, Russia

E-mail: {yuzhanin, iutipikin, igankevich, viz}@cc.spbu.ru

*Received December 4, 2014*

This work describes the idea of the system of informational support for the scientific projects and the development of computational task tracking complex. Due to large requirements for computational experiments the problem of presentation of the information about HPC tasks becomes one of the most important. Nonstandard usage of the service desk system as a basis of the computational task tracking and support system can be the solution of this problem. Particular attention is paid to the analysis and the satisfaction of the conflicting requirements to the task tracking complex from the different user groups. Besides the web service kit used for the integration of the task tracking complex and the datacenter environment is considered. This service kit became the main interconnect between the parts of the scientific project support system and also this kit allows to reconfigure the whole system quickly and safely.

Keywords: service desk, task tracking, HPC, web service

## Комплекс слежения за вычислительными задачами в системе информационной поддержки научных проектов

Н. В. Южанин, Ю. А. Типикин, И. Г. Ганкевич, В. И. Золотарев

*Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35*

В данной работе рассматривается идея системы информационной поддержки научных проектов и построение комплекса слежения за вычислительными задачами. Ввиду больших потребностей в вычислительных экспериментах предоставление информации о вычислительных задачах на HPC-ресурсах становится одной из важнейших проблем. В качестве решения этой проблемы предлагается нестандартное использование системы service desk — построение на ее базе комплекса слежения за выполнением вычислительных задач на распределенной системе и ее сопровождения. Особое внимание в статье уделено анализу и удовлетворению противоречивых требований к комплексу со стороны разных групп пользователей. Помимо этого, рассмотрена система веб-служб, служащая для интеграции комплекса слежения с окружением датацентра. Данный набор веб-служб является основным связующим компонентом системы поддержки научных проектов и позволяет гибко изменять конфигурацию системы в целом в любое время с минимальными потерями.

Ключевые слова: служба поддержки, отслеживание статуса выполнения задач, HPC, веб-сервис

Research was carried out using computational resources provided by Resource Center “Computer Center of SPbU” (<http://cc.spbu.ru/>)

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 615–620.

## Introduction

As a result of the growth of an HPC industry many powerful supercomputers appeared in the last years. Their peak performance seems to be large in terms of GFLOPS, but one should always try to keep resources loaded at an acceptable level in order to achieve good average utilization coefficient of the resources. This implies an application of the Capacity Management to the supercomputer [Gayduchok et al., 2012]. Such approach consists of monitoring, application sizing, analysis and capacity planning, and it could be a really challenging task in modern hybrid systems including the heterogeneous resources (different CPU architectures, specialized accelerators like NVIDIA Tesla or Intel MIC). More than that multi-user environment brings the following problems: load balancing and security, logging and accounting tasks.

Problems mentioned above are solved partly by the resource managers such as Portable Batch System (PBS). There are many implementations of Portable Batch System that differ from each other in detail but with the same main idea. Such systems start jobs in accordance with a schedule that is based on user preferences, capacity plan, resources load and availability. The scheduler can be a part of the PBS system or can be installed from a separate software package. It makes a decision about job start time depending on its rules based on the capacity plan and system data collected by PBS. This data updates constantly with logging in the PBS log files. PBS also store log files with information about jobs: owner, timeframes, requested and used resources, etc. So, in order to get accounting data one needs to have a log parser that will retrieve the data from large PBS log files.

The PBS system has a possibility to send reports by e-mail to users when their jobs are started or stopped (with job status that indicates successful execution or error). But this is inconvenient too: there are too many e-mails with such information (users usually run many jobs), so such a large list of emails is hard to analyze. Console PBS commands usually require some Linux skills, so they are inconvenient for inexperienced users too. In this article some approaches for controlling of the tasks on shared HPC resources, which are quite convenient but not traditional, are reviewed.

These approaches are reviewed as a part of the scientific project informational support system. The support system regulates all relations between the supercomputer center and scientists that execute the projects. Such approach allows using information about computational jobs in analysis and continuous service improvement.

## Conflict of interests

Center for collective use of HPC resources as any social institute involves working with groups of users having different goals. Question of retrieving monitoring information and computational job tracking is answered differently by the two main groups — users and staff of HPC center which have different capabilities and different goals. As a result, they need different representations of the monitoring information about the computational jobs. But within these groups we can also distinguish smaller groups. Users can be divided into executors and leaders while computer center staff consists of administrators and managers. These subgroups have different requirements for the job tracking too.

For example, user who executes scientific task and directly submits computational jobs to a PBS cluster needs to know which of his jobs are completed and investigate job failures. He needs to know execution time of finished jobs and amount of consumed resources. Important factor is user-friendly, clear representation of the information. At the same time user does not need to know such an information as allocated slots or other technical details.

Unlike the executor, research work principal is not interested in every particular job. He must control executors, plan the use of the computational resources necessary for their research. Last, he must report to grantors. That is why he needs statistics about each subordinate: he needs statistics with the same or lower level of detail as executor and at the same time he needs general reports reflecting the overall progress of the research work of his scientific group. For him the tables are more suitable than graphs and diagrams. He usually needs more parameters than executor as he ought to assess the

overall process. He can also find automatic reports generation and prediction of resources usage very useful because he has to estimate the efficiency of the work. Also this way he can generate and submit capacity plans for oncoming quarter. He also has to submit requests for additional resources in case of lack of computational power.

On the other side of computational process there is the staff that supports the Computing Center services: system administrators and managers of the computer center. Administrators should have the possibility to get any available information about any job. At the same time they need to get selections from overall job statistics about cluster usage. For example, it can be necessary for capacity planning and load balancing. Also administrators have to track erroneous jobs and solve any problems related to jobs, system and hardware. So, they need a monitoring system that meets these requirements and also a service desk system which can be used to facilitate work with users and improve work flow. Such system can speed up overall work process, clear interaction with users by creating tickets and speed up the incident processing by registering the known errors and problems.

Manager of computational center, in turn, wants to get information about overall resources usage. At the same time, he wants the possibility to assess the resource usage of specific user groups and he does not need detailed reports but an overview of total resource usage during the different time frames. Also, he needs an opportunity to sort reports by each resource and each scientific group. In case of operation with many different supercomputers such separation of the statistics improves the capacity planning and work with computation services level.

As one can see, the requirements of mentioned groups have both intersections and differences. Universal system corresponding to the needs of all groups is difficult to implement and the problem is intended to be solved with a system based on a set of services attached to the service desk software.

## Web services as an interconnect

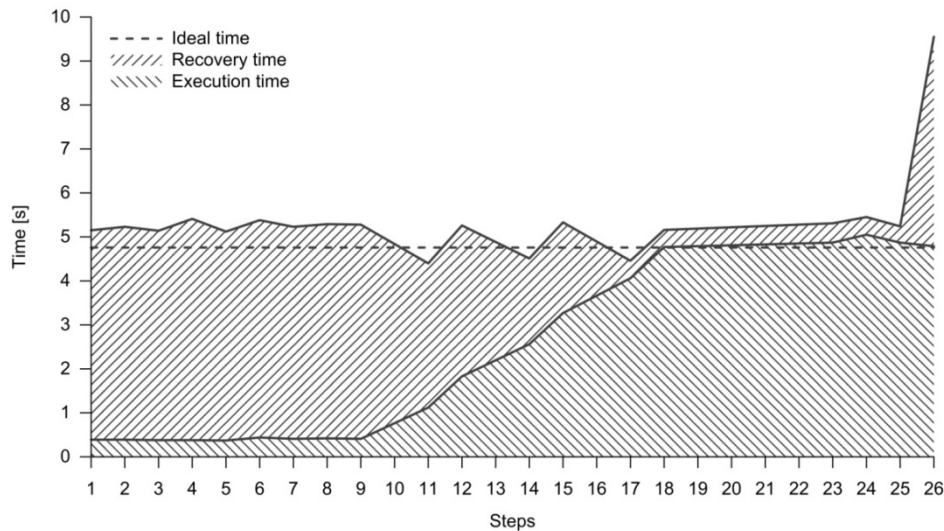
In case of complex information system the problem of connection between modules becomes crucial. Goals of an interconnect are not only correct functioning of data transfer but the ensuring the reliability and timeliness of command execution. Information systems in scientific HPC environment are sensitive to the speed of inside operation. In our case the collective usage center is considered. In such center efficiency of resource usage has a first priority. For the interconnect of information system of such center the best solution will be the web service set. This technology allows to connect the modules flexibly using de facto common network protocols. Also, the advantage of this approach is the desire of developers to provide an API of its products as web services, which facilitates the development of the complex as a whole. It is necessary to clarify that in this case we use web services architecture RESTful, it means strict adherence to the protocol HTTP / 1.1 and the idea of resource availability in the network [Fielding, 2000]. Special attention was paid to the speed of response to the occurrence of the problem while creating complex for task tracking. For this purpose we have developed an automated system for web service call processing. The basic idea is to reduce the recovery time of the settlement process through a mechanism of transactions.

Principle of the transactions of RESTful web services is similar to that of the work of the transaction in ACID model, but with important exceptions. Thus, the atomic system can be achieved if we have control over the operation of a specific web service, i.e. it is important to have the possibility of handling the information/command received. Common basis of all web services relating to a tracking complex achieves consistency properties, but not at any point in time. But as for the isolation property, everything is much more complicated.

This property can not be achieved in practice in case of asynchronous web requests. So, taking into attention these exclusions, we have a reliable system that can work with common RESTful web services.

In this case, each of the states of such an intermediate system is written synchronously into the journal, it allows recreate the state at the time of the emergency stop precisely. In this case, the system start the execution with the last stable point.

In case of failure of the transaction system, time that spent on the query execution is just slightly different from the time spent on the request without the recovery process, as illustrated in the Fig. 1. In the case presented the operation is restored, it makes 26 notes in journal in case of the successful implementation of 26 records. The time is compared with that obtained without the recovery process. The Fig. 1 shows that the recovery time is not enough. With regards to the peak on the 26th step, it is due to the imperfection of the algorithm of analysis at the time of testing — restarts laden branches.



## Implementation

In its usual setting the service desk software provides the main point of incident and service request registration, ticket workflow for them and archiving after the solution was found and the ticket was closed [Kácha, 2010]. Also the service desk software provides the role-based permissions on the ticket workflow steps to the helpdesk staff, accounts the incident or service request solution time, provides reminders, notifications, ticket escalation mechanism, etc. Service desk system also usually has the knowledge base which contains most frequent incidents and its quick solutions.

The basis of our research is a model of supercomputer center which provides scientists with computational resources. Scientific work requires detailed reports to the grantors, that is why such supercomputer center has to provide detailed reports and statistics to its users. In such center the service desk system can be the central point of the interaction between supercomputer center staff and users. All correspondence about the incidents and service requests can be processed by service desk software.

On the assumption of such position of the service desk it is reasonable to make the website based on the open source CMS as a frontend and the service desk as a backend of the user support system. In our virtual testbed OTRS service desk and the website based on the Drupal CMS were integrated using the web services and the corresponding interfaces of both systems [Bakker et al., 2013]. To track the computational tasks on the supercomputers the idea of receiving the reports of PBS as tickets by the service desk was used.

It seems to be simpler to solve the problem without service desk and use direct transfer of the PBS reports to the CMS database instead with email transfer. In this way the processing can be done by a custom website module. However, the usage of service desk software in this case has some advantages. On the one hand the database schema development is already done. On the other hand the service desk software has a convenient object of ticket with set of fields and automatic state machine. This object allows to open a ticket by the keyword in the email header or field in data set received by web service automatically and to close it the same way. And finally the service desk allows the supercomputer center staff to watch the overall picture of the batch system and to operate with the PBS error as with an incident ticket.

Also usage of the service desk system as the core of the information system of supercomputer center corresponds the good practices of the ITIL [Potgieter, Botha, Lew, 2005]. The integration of service desk with PBS and the website realizes a part of the Capacity Management Process.

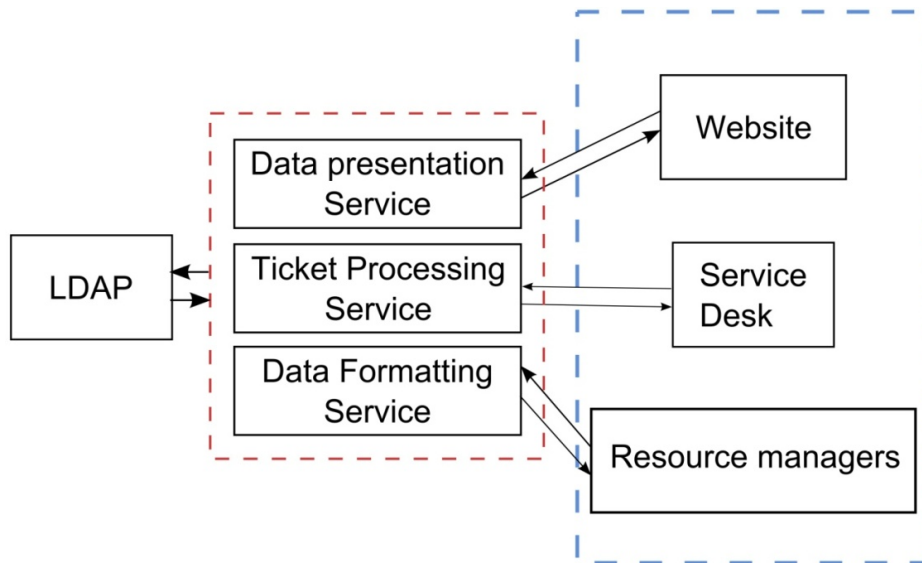


Fig. 1. Task tracker scheme

The first goal of our tracking complex is collecting an information about computational jobs, so the OTRS service desk and the PBS managers were connected via web services, placed on the duplicated transaction server. Every message from PBS becomes an article in a ticket: opening or closing one depending on job status in the message.

The next problem is the report and statistics generation. Relevant information for the user depends on this user role: executor, research work principal, system administrator, computer center manager. Such dependency can be realized by the LDAP technology. LDAP account can contain role information and information about user's priority for the batch system. The OTRS, PBS and either the website uses the common LDAP server to associate the data relative with one user account. For the statistic report creation OTRS Stats module can be used. This tool can be controlled via SOAP, so user can choose reported parameters on the website and generate the report as a \*.csv or \*.pdf at any time interacting with simple GUI.

The question of ordering and recording the activities of various groups of users on the physical computing resources is not a simple question, especially when organizing access to the resource through a heterogeneous system of services. Stats module can be used to gather statistics, but it is not flexible tool and does not allow changing the configuration of the output data "on the fly". For fast and fine-tuning data output, a web service connected to OTRS via the corresponding API can be used.

Personal account is implemented as a module of Drupal CMS interacting with databases (CouchDB, ActiveDirectory) through REST web services and direct requests using Sag library. The summary report of work is made using datacenter resources contains resource utilization figures, however, datacenter systems that log activity of applications running on cluster are not linked directly to this account. To display on-demand computing resource usage statistics, a service that would collect all metrics and represent this data in the user account in accessible graphical form is required.

Information about tasks executed on a supercomputer is stored as chains of tickets: one for the beginning of the computational task and one for the end. This information is attached to personal account through PHP extensions for CMS and different criteria can be chosen to be displayed. PHP extension then submits request to the web service that contains the data and then pre-generated chart is received as a static picture.

Based on received data web service creates a chart and returns it as a bitmap. The main difficulty is the fact that the request to the database has to be completed in a short period of time. That is why

the distributed database is a requirement. For chart creation free Java library JFreeChart is used. Such approach allows meeting the needs of executors and research work principal for the visual representation of data in the form of graphs and charts.

Graphical representation gives users more options for an estimation of resource consumption, which further helps to avoid excessive (or lack of) resource allocation, increases efficiency of computational services and prevents from conflicts between customers and computer center staff.

## Conclusions

The solution of the task of integrating multiple information systems with user-friendly environment for computational task tracking allows an executor of scientific project to make necessary quantitative evaluations of his work on the supercomputer. For example such system can be useful to generate the reports about numerical simulations performed within the timeframe of a research work or to track the computational resource consumption and make prognosis of future resource utilization.

Our system will be especially useful for research work principals. Using the system they can evaluate the overall progress of numerical simulations within the research work, the productivity of every subordinate, the measure and the uniformity of quotas consumption. Also the automated report generator eliminates the need for manual creation of report after the end of research work. Metrics of total resource consumption and its distribution into the timeframe are included into the report automatically.

Supercomputer center staff also will have their profit. Capacity Management can easily receive information about the total computational resource consumption and make the predictions and a capacity plan for services. System keeps the data in the service desk database and allows archiving and making backups to protect the information. Also thanks to the database the reports can cover any time period. Such an opportunity is very useful for the computing center managers.

Another advantage of solution based on the service desk system is a possibility to increase the efficiency of solving the program errors on the supercomputers. The time of error recognizing decreases thanks to automatic incident registration. That is why overall time costs also decreases and the helpdesk staff performance indicators increases.

## References

- Bakker R. et al.* OTRS 3.3 — Admin Manual. 2013. [http://ftp.otrs.org/pub/otrs/doc/doc-admin/3.3/en/pdf/otrs\\_admin\\_book.pdf](http://ftp.otrs.org/pub/otrs/doc/doc-admin/3.3/en/pdf/otrs_admin_book.pdf) (retrieved on 2014-04-07).
- Fielding R. T.* Architectural styles and the design of network-based software architectures: Ph. D. thesis / University of California. 2000. P. 162.
- Gayduchok V. Yu., Bogdanov A. V., Degtyarev A. B., Gankevich I. G., Zolotarev V. I.* Virtual Workspace as a Basis of Supercomputer Center // Proceedings of the 5th Intern. Conf. “Distributed Computing and Grid-Technologies in Science and Education” (Dubna, 16–21 July, 2012) / Joint Institute for Nuclear Research (Dubna). — 2012. — P. 60–66.
- Kácha P.* OTRS: CSIRT WorkFlow Improvements. — CESNET, Tech. Rep., 10 2010.
- Potgieter B. C., Botha J. H., Lew C.* Evidence that use of the ITIL framework is effective // 18th Annual conference of the national advisory committee on computing qualifications, Tauranga, NZ. 2005. — P. 160–167.