Journal of Computational Science 14 (2016) 78–84

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

In need of partnerships—An essay about the collaboration between computational sciences and IT services

Anton Frank[a], Matti Heikkurinen[b,*], Ferdinand Jamitzky[a], Helmut Satzger[a], Dieter Kranzlmüller[a,b,**]

[a] Leibniz Supercomputing Centre (LRZ), Garching n. Munich, Germany
[b] MNM-Team, Ludwig-Maximilians-Universität München (LMU), Munich, Germany

ARTICLE INFO

Article history:
Received 15 July 2015
Received in revised form
18 December 2015
Accepted 24 January 2016
Available online 29 January 2016

Keywords:
Computational science
IT services
HPC
Supercomputing
Partnership

ABSTRACT

Research in computational science (CS) poses several challenges, stemming both from the scientific domain being studied and the sophisticated IT infrastructures used by the CS researchers. IT infrastructures typically consist of the latest and most powerful supercomputers, high-performance networks and high-capacity data storage services that are provided, developed and operated by IT experts who do not have in-depth understanding of the CS problems being studied. This "standard service approach" has been a very useful tool for streamlining the interaction between IT and application domain users. However, in many cases it has started to become a limiting factor that may prevent realizing the full potential of the IT services when tackling the most challenging CS research topics. The partnership initiative π^{CS} established at the Leibniz Supercomputing Centre (LRZ) aims at transforming the relationship between Computational Scientists and IT service providers from a service-centered approach to an integrated partnership. The partnership between CS researchers and IT service provider allows IT services to better address the requirements of CS in an optimized and efficient manner. In addition to describing this model and its background and results achieved so far, the paper sheds some light on future development and assessment activities.

1. Introduction

Simulation is accepted today as "the third pillar of research", allowing scientists to advance science by using computational solutions in a wide range of disciplines. Scientific computing focuses on analyzing and solving problems that are impossible or impractical to study algorithmically. Scientific areas that are strongly dependent on scientific computing methods (and supporting data services) are often called "computational sciences" (CS). The use of plural is one of the indications that the term covers not only the computational aspects of the research (software algorithms or optimization of implementation), but also key research questions of the scientific fields that require large-scale computing resources. In most scientific fields, CS is accepted as the most powerful tool for accelerating progress. Hence, limitations of the computational power that can be used to solve the key research questions will have a direct impact on the overall progress of the discipline. This direct link between computing capacity and potential for scientific discovery has led to the installation of more and more powerful supercomputers around the world. One of the indicators of this growing interest is the turn-over rate between consecutive Top 500 supercomputer list[1] (ranking of supercomputer centers that is published every 6 months) that is usually about 50%.

But how are CS practitioners using these machines? In most cases, simulation codes are initially developed on desktop machines or on small-scale distributed computing systems (own departmental clusters or, more recently, using different Cloud services). In this initial stage, responsibility for operating and maintaining IT services (or procuring services, in case of commercial Cloud environments) lies typically within the research group. Thus, a computing expert who understands also the intricacies of the research domain being supported is practically always available for general support, modifying existing IT system, and even allocating (at least a modest amount of) additional resources when needed. The "IT service interface" tends to be quite informal, with a verbal

* Corresponding author at: MNM-Team, Ludwig-Maximilians-Universität München (LMU), Munich, Germany. Tel.: +49 8921809143.
** Corresponding author at: Leibniz Supercomputing Centre (LRZ), Munich, Germany. Tel.: +49 (0)89 2180-9146.
E-mail addresses: heikku@nm.ifi.lmu.de (M. Heikkurinen), kranzlmueller@mnm-team.org (D. Kranzlmüller).

[1] www.top500.org.

http://dx.doi.org/10.1016/j.jocs.2016.01.008
1877-7503/

face-to-face request or a free form email likely the most common approaches.

However, once the software runs correctly and efficiently, the scientific progress can quickly become limited by these locally available IT capacities and capabilities. To overcome these limits, scientist needs to request access to external services from local, regional, national or even international providers. This means approaching the academic IT resource owners either directly, or via joint efforts such as EGI[2], PRACE[3] or XSEDE[4], because in most cases the research group scientist belongs to will not have the financial resources and suitable administrative procedures to contact commercial providers directly. As a result, the scientist needs to usually adhere to a peer-review application process, where access to IT resources is justified by the scientific quality of the proposed research, as well as the quality of the application software. The evaluation of the proposals determines the amount of core hours and other resources (such as capacities in different levels of storage systems) to be provided to the applicant.

Once the resources are granted, there are several help services and solutions available to aid in the use of resources: a support hotline addressable via e-mail, phone, or a ticket system to log problem reports or change requests. However, compared to the smaller scale "local" scenario, a long-term, personal relationship with the service provider either doesn't exist (i.e. support tickets are assigned randomly among the support staff), or it takes longer to build due to lack of face-to-face contact. Combined with the fact that the IT support staff most likely won't have a same degree of knowledge about the scientific domain as the local one, this can make the communication between CS researcher and IT service provider considerably less efficient.

Despite its limitations, this standardized and somewhat impersonal approach has successfully supported the delivery of the latest scientific breakthroughs. Once the CS solution has been successfully deployed, the operations are reliable and performance achieved can be predicted relatively accurately. However, today the situation is changing in several ways:

- The efficiency and ease of using remote resources has dramatically increased with the near-ubiquitous availability of high-speed networks, which allow accessing computational resources almost anywhere on the planet. This increases the number of systems a user is able to access, making portability of the software and data important earlier in the CS research process lifecycle. However, achieving predictable and reliable performance in this multi-system environment is more challenging.
- Using the high-end machines is getting more and more complicated, due to large numbers of cores, sophisticated and interconnected software stacks, and other system-specific characteristics. In the mainstream IT, changes in the hard- and software architecture happen relatively rarely and tend to offer long periods of compatibility (e.g. in the case of Apple's switch of hardware architecture from PowerPC to Intel in 2006, the latest versions of the operating system could support old PowerPC binaries all the way to mid-2011[5]). In contrast, a switch from a shared memory systems to massively parallel systems required considerable rewriting effort. In case of high-end HPC, major software development efforts are often necessary just to retain application level performance achieved with the previous generation machine.
- Increased complexity means that code porting and optimization may take up increasing amounts of researchers' time. Supercomputers are often also oversubscribed, leading to increased waiting times and thus further increasing the time and effort needed in the code optimization phase.
- Operating and running IT resources is often more costly (over the lifetime of the machine) than the original capital investment was. Thus, in many cases it is cheaper to buy a new machine than to run an old one for its whole projected life time. This increases the frequency of the code porting and optimization phases, further limiting the time researchers can dedicate to new CS research activities.
- The costs of supercomputers are often determined by their energy and environmental costs. As a result, execution of a single CS simulation can have considerable financial and environmental cost. It is difficult to make users aware of the issue while at the same time encouraging them to take the necessary calculated, experimental risks to generate novel innovations.

All of these issues have to be taken into consideration for both computational scientists using these machines and IT service providers. The limitations of the current standard approach, especially in the context for state-of-the-art research, were the key observations that led to the development of the partnership approach. Further analysis led to the conclusion that the current customer-provider relationship needs to be replaced by a CS-IT partnership where both collaborators work on equal footing. In contrast, impersonal approach with mechanical efficiency-related metrics was deemed to carry risk of stifling innovation, due to factors on both CS and IT service provider sides. For example, testing a new algorithm represent a risk for both parties: needing to resubmit jobs may lead to delays and short-term loss of "efficiency". From the supercomputing center's point of view these downsides can be especially problematic if they are analyzed based on oversimplified efficiency metrics. They e.g. increase the load (and costs) of the support hotline, while at the same time dropping the utilization rate of the computer systems. However, these costs should be seen as necessary investments in the innovation process, with an understanding that capturing the return on investment requires taking a long-term view.

We believe that the partnership-oriented approach is the way to maximize the innovation potential of the computational sciences. Only through such an integrated approach it is possible to ensure that computational scientists get access to the most appropriate tool for their research, while at the same time allowing IT services providers to optimize the operations of the infrastructures to maximize the innovation potential.

This paper describes the partnership initiative π^{CS} established at the Leibniz Supercomputing Centre (LRZ) as a concrete way to reach this goal. The paper is structured as follows: In Section 2 we introduce the petaflop-scale machine SuperMUC as a representative of one of the most powerful IT infrastructures today. The architecture and specifics of SuperMUC are presented, focusing on the issues that are most relevant for the computational scientist. Section 3 describes the Extreme Scaling Workshop series at LRZ, which has been established as an incentive for scientists to bring their codes to the largest machines available. These events and their results also demonstrate the importance of CS – IT collaboration. Section 4 describes the concept of the partnership initiative, its current implementation as well as some first examples. The paper can be seen as less of a research paper in scientific computing, but more an essay that highlights difficulties and solutions related to computational sciences. We hope to stimulate further reflection related to these issues to identify further improvements for future ecosystem around CS applications and IT infrastructures.

[2] www.egi.eu.
[3] www.prace-project.eu.
[4] www.xsede.org.
[5] https://web.archive.org/web/20110107211041/http://www.apple.com/rosetta.