Future Generation Computer Systems 29 (2013) 460–471

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

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

Performance evaluation of measurement data acquisition mechanisms in a distributed computing environment integrating remote laboratory instrumentation

Luca Berruti [b,c], Franco Davoli [a,b,*], Sandro Zappatore [a,b]

[a] DITEN, University of Genoa, Via Opera Pia 13, I-16145 Genoa, Italy

[b] National Inter-University Consortium for Telecommunications (CNIT), University of Genoa Research Unit, Via Opera Pia 13, I-16145 Genoa, Italy

[c] SIR S.r.l., Via XX Settembre 3/6, I-16121 Genoa, Italy

ARTICLE INFO

Article history:
Received 25 April 2011
Received in revised form
12 July 2012
Accepted 20 July 2012
Available online 1 August 2012

Keywords:
Grid-based architecture
Tele-Laboratory platform
Performance evaluation

ABSTRACT

The usage of laboratory and measurement instrumentation of any kind, from large complex equipment to networks of sensors that collectively appear as a distributed measurement device, has become of relevant importance in all branches of experimental sciences. Owing to the increasing networking capacity and access ubiquity, this bulk of instrumentation is ever more frequently accessed remotely by users who want to perform experiments, collect and process experimental data, analyze and interpret results. With reference to a remote instrumentation architecture deeply rooted in distributed computing paradigms such as grids and clouds, we evaluate the performance of mechanisms for the collection of data generated by instruments, in order to assess the capabilities of remote instrumentation services. In the presence of instruments generating measurements at high rate, which must be delivered to a multiplicity of users, publish/subscribe dispatching (push) mechanisms are shown to outperform pull-based ones.

1. Introduction

Managing experiments on remote laboratories over the Internet has been a topic of investigation for many years since the inception of the network. Apparently, one of the main areas where this topic has raised interest is that of scientific education. Ref. [1] reports an extensive survey on this aspect up to 2006 and, in fact, quite a few more recent developments (e.g., [2–7]) put emphasis on the educational side. The specific laboratory and instrumentation environments covered are wide, ranging from chemistry [8] to robotics [9], medical imaging [10] and telecommunications [11], among others.

More generally, the benefits of accessing, configuring, and managing remote physical instruments and whole laboratories, as well as launching and controlling the execution of experiments, analyzing and visualizing their results, extend far beyond the educational sector. The complex of activities generally recognized as *eScience* [12] requires highly intensive computation on huge data sets in a large-scale networked distributed environment, and the capability of real-time interaction and collaboration among many scientists working on a specific problem or data set worldwide. eScience embraces such diverse disciplines as particle and high-energy physics [13], astronomy and astrophysics [14], geo-science [15], biology [16], and medicine [17], just to mention a few. The very large data sets it deals with are usually produced by specific instrumentation—the pieces of equipment physically generating the data that pertain to a certain experiment. The high interest in accessing and managing remote instrumentation and scientific laboratories in general is witnessed by a number of international research efforts undertaken in this field, both in the US [18] and Europe [19–21].

There are also perspective market opportunities for new services, fostered by the flexibility of access to instrumentation. One may think, for instance, of the case of a Small–Medium Enterprise (SME), in need of a very specific, sophisticated and costly instrument, in order to comply with the requirements of a certain customer or to be able to complete the development of a product, without having the guarantee of a short-term utilization of the same piece of equipment for other products. The existence of a service provider, able to offer networked temporary access to the necessary instrumentation, would certainly constitute an appealing opportunity.

The access to remote instrumentation has a long history, and there are widely diffused and consolidated commercial products (notably, LabVIEW® by National Instruments [22]); however, the generalized usage of instruments over the network has spawned a number of efforts toward open software architectures and models.

* Correspondence to: DITEN- University of Genoa, Via Opera Pia 13, I-16145 Genoa, Italy. Tel.: +39 0103532732; fax: +39 0103532154.
E-mail address: franco.davoli@unige.it (F. Davoli).

0167-739X/$ – see front matter
doi:10.1016/j.future.2012.07.007

Most of the first platforms that emerged for this purpose exploited Java components, sometimes including native C code and embedding LabVIEW at the server-side (e.g., [23,24], among others). The advent of web services, and then of the Service Oriented Architecture (SOA), has fostered the integration of remote instrumentation within distributed open source computing platforms such as the Grid, through the development of specific middleware elements (see, e.g., [25–27]).

There are essentially two different communication requirements in such middleware-based remote instrumentation environments: (i) the exchange of management and control messages between instruments and their users (for configuration, status checking, parameter setting, etc.); (ii) the retrieval of measurement data from the instrumentation and their distribution to a potentially large user population.

With reference to the specific architecture designed and implemented by the DORII project [21,27], this paper aims at evaluating and comparing the performance of two communication paradigms addressing the second requirement mentioned above. The results of the comparison can serve the purpose of dimensioning and engineering the data collection/distribution part of remote instrumentation systems, by choosing the most appropriate mechanisms to observe the outcomes of remotely performed experiments, with respect to their scalability with the number of users, the observed system's dynamics, the nature of the access network, and the retrieval system's complexity. After describing the motivations that led to the implementation of the DORII platform and its main architectural elements in Sections 2 and 3, respectively, we concentrate on the performance evaluation of the data transfer modes from the instruments towards a multiplicity of users' clients. The message exchange mechanisms we consider are based on pull and on publish/subscribe (push) paradigms. They play an important role in relation to different applications and to the frequency of measurement updates. Section 4 describes the performance evaluation environment, and Section 5 reports the experimental results. Section 6 contains the conclusions.

2. Measurement application environments

Numerous user communities are involved in research activities that require high computational capacity to process data generated by experimental equipment. In general, there are different ways in which the data are treated. Among others, we mention explicitly here a number of aspects pertaining to problems and requirements.

(a) *Processing*: Data may be processed (or pre-processed) locally at the acquisition instrument (e.g. a spectrum analyzer actually performing the analysis in the frequency domain of the acquired signal, to produce the power spectrum), or remotely at some computing elements that receive raw data (e.g. a prediction model of a physical phenomenon running on the basis of data acquired on the field, as in meteorology, oceanography, or seismic engineering), or both. As concerns processing time constraints, the data may be either gathered and processed in real time, or they may be gathered, temporarily stored, and then processed without particular real-time constraints.

(b) *Location of instruments*: The acquisition devices/instruments may be co-located (i.e. they are part of a single real laboratory with a specific physical location), or they may be geographically distributed and coordinated to perform a cooperative measurement (e.g., in electronic Very Large Baseline Interferometry—e-VLBI, a number of radio telescopes observe the same portion of the sky, and their observation data are correlated to obtain a high resolution interferometric map).

(c) *Numerousness of users and devices*: The number of users that are interested in the experimental output data and observe them (or the results of their processing) at the same time may be variable from a single user to a large population (e.g. in the latter case a tutor may control the experiment, while remote participants in a distance learning session visualize the results). The number of measuring devices that compose the system may range from one or several units (e.g., a single or multiple antenna system capturing one or more wireless signals) to a myriad of small sensors that observe a distributed phenomenon (e.g. a wireless sensor network for environmental monitoring).

(d) *Heterogeneity of platforms*: The internal architecture, software, and user interfaces of the measurement systems may be proprietary or based on open standards. In general multiple types of devices with different features need to be supported.

In order to better highlight the scenario, a specific use case is discussed in Appendix A.

Though the list of features reported above is certainly not exhaustive, it suggests a relevant aspect that is common to most application environments of experimental science: data are produced (either spontaneously or upon user-generated stimuli) by a physical phenomenon that is being observed, monitored, or controlled; the data are collected in some places, processed at the same or in another location, and the results are to be presented to and interpreted by the user(s) carrying out the experiment. In general, the experimental environment may be itself a distributed system. Therefore, it is appealing to view the whole experiment monitoring and measurement process in the framework of distributed computing systems, where a very large base of knowledge and practical implementations already exists. In this respect, whereas the data processing and storage aspects lend themselves almost naturally, the same is not completely true for the measurement and laboratory instrumentation, which actually produces the data. This instrumentation should be exposed as a manageable resource, in order to become a full-fledged member of a well-integrated distributed organization, such as provided by the paradigm of Grid computing or, more generally, of a Service Oriented Architecture (SOA) [28,29].

Some of the remote instrumentation projects already mentioned in the Introduction (notably, CIMA [25], GRIDCC [19], RINGrid [20] and DORII [21], among others) attempted to address the requirements derived from such a wide spectrum of features that pertain to different application environments. The general philosophy underlying these approaches is characterized by a number of common basic viewpoints:

- The construction of suitable abstractions of the remote instrumentation, in order to make it visible as a manageable resource;
- The intention to present the user standard interfaces, which allow browsing the "distributed laboratory space", choose different pieces of equipment, configure their interconnection, orchestrate experiment executions, collect, process and analyze the results—all by providing also built-in security services;
- The implementation of standard capabilities to perform whatever functionality may be required by the application, irrespectively of the communication network that provides the data transport and, to a certain extent, of the user community conducting the experimental activity.

The Remote Instrumentation Services (RIS) architecture stemming from these efforts has been outlined in [20]. In the next section we describe the RIS platform that was developed in the DORII project [21,27], which we will refer to as the Grid-based Tele-Laboratory Platform. As already stated, our interest is in evaluating communication mechanisms for data collection from the