Distributed computing platforms like clouds and web standards: what could be the solution in an open environment?

SERENA PASTORE
INAF - Astronomical Observatory of Padova
Vicolo Osservatorio 5 – 35122 Padova
ITALY
serena.pastore@oapd.inaf.it

Abstract: - Complex distributed computing platforms, like cluster systems, grids or clouds, are today’s choice to deploy modern applications. With the proliferation of different devices, most of them in the mobile area and the wide support of web social networks, a wider range of applications are executed through a browser, the software components that are currently present in all of the users’ devices. Moreover, since the shift of application development from native applications to web applications, modern applications now require complex features in term of supported platforms and development standards to guarantee the applications’ interoperability. Distributed architectures such as cloud computing and web standards seem to be the candidates, but there are some issues related to their real applicability, especially when considering an open environment and the languages adopted. Working in a research institute, our focus is on an open platform and adherence to standards since only common standards could guarantee interoperability solutions and real culture diffusion. This paper discusses our approach in searching for suitable platforms and standards to develop applications for several scientific fields: from astronomical applications that are based on web user interfaces querying on distributed databases and producing output in different formats, to communication and outreach fields where the web is the natural platform to disseminate information.

Key-Words: - cloud computing; grid computing; web applications; web standards technologies; markup languages

1 Introduction
Distributed computing platforms [1] are the natural choice in providing Internet services to guarantee service quality (i.e., redundancy, load balancing, geographical distribution, and etcetera), even if these require complex management of hardware and software infrastructures. In the last years, both grid computing and cloud computing [2] have emerged as the right platforms to solve some of the problems that current applications have. The research and educational environment experimented on grid computing through the establishment of European Grid infrastructure (EGI) [3] that has been developed and tested within the EGEE project [4] that officially ended in 2010 and now is supported through the EGI.eu Foundation. The foundation aimed at collecting several national grid initiatives (i.e., NGIs) and guaranteeing the long-term availability of a generic e-infrastructure. In the astronomical environment, there are reports of some successes in the use of such a platform for the execution of some kinds of applications (i.e., batch applications requiring a large amount of computing and storage capabilities); but in the web application area, there are reports of failures in the platform’s adoption. Applications that for example required web services in order to interact with distributed databases manifested some issues that required an improvement in the entire infrastructure through the introduction of a new element that is able to model database resource [5]. Probably, this latter problem was due principally to the software features of the middleware adopted (i.e., the gLite [6] toolkit). This middleware was developed around job scheduler mechanisms, and was initially poorly customized for web applications since it lacks specific services as the discovery mechanism able to search and find the best grid resources for a web application [7]. Moreover, the need for restrictive requests in terms of authorization policies, limit the usability of shared resources whose control remains on the owner of the site. Actually, there is an effort, within the European Middleware Initiative (EMI) [8] in order to create an upgraded middleware known as Unified Middleware Distribution (UMD) as the fusion of the used grid middleware gLite, the ARC [9] and the UNICORE [10]. On the other hand, most current applications are developed as web applications, thanks to the advanced features of web browsers that are becoming the ideal execution environment with their simplified development curve and management. In this area, with the adoption of web services technology and the wide diffusion of the service-oriented vision [11], another distributed approach is used through the introduction of the cloud paradigm. Thanks to virtualization
technologies [12], the resource concept is extended to include an entire virtual machine, with all software stacks installable by the user. However, the different distributed paradigms do not seem to be in competition; they seem to be complementary platforms customized according to the type of application to be deployed on them. In Italy, the national grid initiative converges through the Italian Grid Infrastructure (IGI) [13], and one of the topics under study is the possible effective integration of the two platforms.

The web technologies field shows different trends. Applications are becoming more rich and interactive, but the standards supporting such features seems to create obstacles in applications’ development, as the case on the future of HTML languages has shown [14]. This paper describes our preliminary study, meant to evaluate the state of the art standards and implementations in two areas related to web applications deployment. Our aim is to create a sort of “best practice” in choosing platforms and applications, both for web-based astronomical applications (i.e., those that interact with database) and for communication and science dissemination goals. The area is that of an open environment, and thus implementations should as much as possible create an open community.

2 Astrophysics and the web

The Internet and the web are two fundamental aspects of actual science, both from the research point of view and in the communication and dissemination areas. Most astronomical applications need great storage and computation capacities and, thus such applications make use of distributed platforms that are mostly based in an Internet network in order to exploit some paradigms as grid. The web is the ideal platform to disseminate information about astrophysics, its projects and its discovery; it is also the ideal software for e-collaboration, website management and social network. We shall focus on two aspects like distributed paradigms and web technology in order to define the best practices on platforms and methods that can be used in web application development and deployment, considering that the web is going for another evolution and the battle between open source and proprietary software is escalating. Two specific aspects should be considered: the commercial trend of cloud computing and standards issues in web technologies that limit interoperability between solutions.

2.1 Promises of cloud computing and considerations about open cloud implementation

Starting from our experience in grid computing and deployment of web applications, we begin by analyzing cloud computing that seems to solve some of the issues introduced by the adoption of the grid. The core concept of cloud computing starts with services provided via the web. As the NIST [15] definition states “cloud is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable resources that can be rapidly provisioned or released with minimal management effort or service provider interaction”. The model manifests three delivery models, as the Fig. 1 shows: SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service). This means that there are different deployment solutions related to the type of offered service. This is outlined by the presence of various cloud software implementations, both commercial and open source. They range from offering software, available as APIs to frameworks to develop and deploy applications in various programming languages (i.e., Google App engine solutions [16] or Microsoft Azure [17]) and into a complete machine. In the latter case, issues related to ownership are solved and thus in such a virtual environment it is possible to install the software stack needed for a specific execution.

![Fig 1: Different types of cloud implementation](image)

According to the literature, the cloud is a mixture of Service Level Agreement (SLA), web services, and virtualization technologies: it seems to be an evolution of already used technologies that follow a different business model. Cloud computing’s best implementation in the commercial environment rests on it capability to answer user demands for computer resources. The most famous and used solutions are the Amazon Ec2 solutions (AWS) [18], even if most of IT and software enterprises (e.g., Oracle [19], HP [20], and etcetera) offer their specific cloud models and platforms. For the IaaS solution, the discriminator factor is the virtualization technique that ranges from open source solutions like Xen or KVM, to Vmware and Microsoft Virtual Machine Manager (VMM) based on Hyper-V technology [12]. A comparative study between grids and clouds has been done within the EGEE project [21]; they study looked at EGEE grid implementations and the Amazon Web Service (AWS) for cloud. This is an important study, the main results of which are reflected in the Fig. 2. The main differences between these two
platforms based on different environments where these two distributed platforms are used, is as follows: grids are typically used for job execution, while clouds are used to support long-terms services.

<table>
<thead>
<tr>
<th>Target Group</th>
<th>EGEE Grid</th>
<th>Amazon Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>Scientific community</td>
<td>Business</td>
</tr>
<tr>
<td></td>
<td>short (batch style) processing</td>
<td>long (data intensive, distributed)</td>
</tr>
<tr>
<td>SLA</td>
<td>local (between the EGEE project and</td>
<td>global (between Amazon and users)</td>
</tr>
<tr>
<td></td>
<td>resource provider)</td>
<td></td>
</tr>
<tr>
<td>Resource-side middleware</td>
<td>Open Source (Apache 2.0)</td>
<td>Proprietary</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Heavy</td>
<td>Light</td>
</tr>
<tr>
<td>Ease of Deployment</td>
<td>Heavy</td>
<td>Unknown</td>
</tr>
<tr>
<td>Resource Management</td>
<td>probable similar</td>
<td></td>
</tr>
<tr>
<td>Funding Model</td>
<td>Publicly funded</td>
<td>Commercial</td>
</tr>
</tbody>
</table>

Fig. 2: A comparison between features of a grid computing (i.e., EGEE grid) and a cloud computing implementation (i.e., Amazon cloud)

Probably, the best solution could be their integration; such, however, could be possible only if open source software implementation were made available to encourage standardization. Unfortunately, this is not yet the case and there are only few open source implementations of a cloud. Within this panorama, Eucalyptus [22] provides an open source version of a web services based implementation of a cloud computing infrastructure. The architectural components run on top of existing resources (i.e., Linux distribution and hypervisor agnostic) and maintain a compatibility with commercial solutions such as AWS. Such a solution could be adopted in our environment, since it satisfies our goals. Users’ of Eucalyptus are also part of a community, called the Eucalyptus Community Cloud (ECC) [23]. This is an environment where they can test drive and experiment this software.

2.1.1 Applications in the web 2.0 era
Software development followed a series of changes over time, as Fig. 3 shows.

These changes, involved paradigm models and approaches in writing code. In the last years, we have seen a shift towards the web as the ideal platform to execute applications: web languages are gained a great popularity through standard development efforts in order to establish such a platform. We could think of such a deployment and execution environment as a web-oriented platform (Fig. 4) that is comprised of a core element: that is, the data offered as a web resource in a wider meaning. Through the web, such data is distributed in different ways that go beyond its simple delivery as hypertext.

Fig. 4: Components of a web-oriented architecture

Developing an application today is a matter of deploying services in the Internet. Although some software remain locally installed in users’ machines, frequently used applications are executed through a web browser and thus are developed with the use of web technologies. Clients are becoming more and more complex and able to execute complex web applications that enhance the user’s experiences (Fig. 5).

This is due to the wide adoption of the ECMAScript [24] language specification (better known as Javascript) that is an indispensable part of making web applications since it allows the web to become a dynamic environment. Web applications are comprised of user interface code and back-end data. The user interface code runs in markup and client-side languages, while the back-end code, which can be proprietary, can run in
whichever language. In publishing audio or video, there are the plug-in technologies such as Flash or Silverlight [25] that are not open but are actually used in web multimedia thanks to their advanced features. The HTML5 specification [26] is an enhancement in hypertext markup language that aim to solve issues found in a previous version of HTML and addresses the need for web applications. The issues related to web standards are addressed by the W3C (World Wide Web Consortium) [27], the most important organization as regards standards and best practices in the web. This organization seems to abandon HTML in flavor of its redesign in XML-like language [28]. XHTML version 2.0, unlike previous specifications, is based solely on XML and even if it is strictly related to all XML technologies, it does not provide compatible with existing content or previous markup language versions. This was a great problem for web developers that continue to use previous versions of the language that are more simple to use. Also for this reason, a group called the Web Hypertext Application Technology Working Group (WHATWG) [29], initially founded by individuals from most commercial enterprises (i.e., Apple [30], Mozilla Foundation [31] and Opera Software [32]), works actively at HTML development in the optics of web applications, and thus of APIs and rich interfaces. HTML5 sounds like new version of the previous markup languages specifications (i.e., HTML 4.0, XHTML 1), plus the enhancement of the document object model (i.e., DOM); nevertheless, it is more a transitional technology than a revolutionary one and is meant to address deficiencies in HTML4. This working group and the W3C are now working together to create this next generation language (Fig. 6), after the W3C decided to review the adoption of the HTML5 specification. However, at the present, the two versions manifest minor differences, even if their overall goals are equivalent.

Fig. 6: Current frequently used versions of the hypertext markup languages

Evolution of such a language seems to be forced by main software enterprises (i.e., Microsoft, Apple, Google and Adobe) due mainly to the adoption of their new devices (e.g., Apple with iPad or iPhone), or plug-in technologies (e.g., Adobe with Flash platform technologies [33] or Microsoft with Silverlight [34]) or their great free web services (e.g., Google apps [35] and the Google operating system aka Google Chrome OS [36] that is considered a web operating system based on the open source project Chromium OS [37]). Special attention is also put on mobile web applications, which are very hard to program especially due to the different languages of the many different devices. HTML5 could be the unifying language. Actually, there is a debate between native versus web applications particularly on mobile devices. The debate is between the development of installable applications based on a native programming language and applications that live in the web browser. Installable applications allow for the building of superior user interfaces and for the access of hardware capabilities (such as the accelerometer). However, for advanced hardware integration, more W3C working groups [27] (i.e., WebApps, Device APIs and Policy, and Geolocation) hope to develop a Javascript API that reduces the advantage of native applications.

3 Suggested solution

The solution should be as much as possible an open-source model both in software implementation and in the distribution and dissemination of information. The creation of a research community aimed at redistributing technical knowledge is also a vision of our study.

Our case studies regard applications that are related on two main fields: those based on web user interface and interaction with databases that allow users to extract specific data and provide multimedia files, and those that enable communication and outreach in astrophysics and require rich interfaces. In these two areas, the interaction with users require the creation of dynamic applications that in many cases could take advantage of on-demand resources, but only for specific period (i.e., related for example to astronomical events). Our study is focused on providing the right platform able to satisfy all these constraints; it could become the ideal platform to develop and deploy these kinds of applications.
Our attention is both on distributed platforms and on methods to develop applications that could be adopted in an open environment, such as that of research institute. The grid is initially born and developed as an open environment, even if it is subjected to security constraints related to owner resources; the cloud, on the other hand, is born primarily as a commercial environment with the aim of pay and use. However, there are several incentives for the creation of an open cloud community, in some sense parallel but interoperable with the commercial one. Example would be, the establishment of the Open Cloud Consortium [38] that supports the development of standards for cloud computing and frameworks for the interoperation between clouds.

Moreover, an interesting initiative is the Open Cloud Manifesto [31] dedicated to the belief that the cloud should be open. In the lists of supporters of such a vision [39], included in the customer list is the European Space Agency (ESA) for the astronomical data processing of the Gaia mission [40]. The agency decided to prototype a cloud-based system to analyze their data. Supporting such a model for specific applications related to provide network services could be a real opportunity. Unfortunately, in an open environment, one of the few possible choices seems to be the Eucalyptus software that provides an interaction with commercial solutions. The other important aspect is about applications and the transition from stand-alone to web apps. Even if specific applications take advantage of being stand-alone to maintain a great performance, most applications could be enhanced if executed through a browser, both for the possibility of execution in most devices and the lack of installation, and thus the need for specific interdependency. In writing any kind of flexible code and specifically for web applications, the two key concepts are portability (and thus the ability to run components or systems written for one environment into another) and interoperability (and thus the ability to write a code that works with multiple providers regardless of the differences between them). HTML 5 promises to help in writing applications that work across-platforms and thus our study suggests to work in this direction and thus in using HTML5, CSS [41] and Javascript for programming applications especially as sophisticated user interfaces that collect input data from a database and create output as video collection. This is in spite of the continued browsers war with major players that develop some technology of their own and thus HTML 5 has little support in modern browsers, just like all the various standards in the past. Moreover a great issue is in the mobile area that, more than in desktop solutions, manifests niche market in operating systems, as applications like the iPhone and Blackberry show.

4 Conclusion

Cloud computing and HTML5 are two buzzwords in the IT environment that in some way demonstrate that we are now really going into an always connected environment. Network services and distributed applications could take advantage of on-demand resource requests without reconfiguring the hardware and software infrastructure, while application could include multimedia and interaction capabilities without plug-in technologies, and through a technology that cloud really be cross platform. For these main reasons, we suggest the adoption of these technologies that, though born in a commercial environment, could find an implementation as an open platform. The proposed solution is the Eucalyptus as an open implementation of the IaaS cloud infrastructure that is able to offer a web application deployment environment on demand but in the form of a virtual machine resource and thus with the full flexibility. On the other hand, we continue to work on making the application developer follow web standards, even if that could require complex solutions and need more attention in considering specific turnaround to support the different browsers. However, we think that an open community means more than transparency, and the problem with any new technology is how to avoid vendor lock-in.

References:
[27] World Wide Web Consortium (W3C), http://www.w3.org
[34] Microsoft Silverlight, http://www.microsoft.com/silverlight
[38] Open cloud manifesto, http://www.opencloudmanifesto.org
[41] Dan Cederholm, CSS3 for Web Designers, A book Apart, 2010