

## Synergies between SOA and Grid Computing

ZAIGHAM MAHMOOD

School of Computing, University of Derby,  
Kedleston Road, Derby, DE22 1GB, UK  
z.mahmood@derby.ac.uk

### Abstract

*Service Oriented Architecture (SOA) is an architectural style for developing and integrating enterprise applications to enable an enterprise to deliver self-describing and platform independent business functionality. Grid Computing (GC) is a framework that allows pooling of physical resources to enable virtualisation of distributed computing, enterprise data and enterprise functionality. The two are synergetic in the sense that, whereas SOA can provide a strong basis for GC, technical framework based on GC provides the optimum foundation for SOA. This paper discusses the two paradigms and provides some useful information for the benefit of large enterprises that wish to embark on the development and implementation of SOA based on Grid Computing.*

**Keywords:** *Service Oriented Architecture, SOA, Web Services, Grid Computing, Enterprise Grid, Information Technology.*

### 1. Introduction

Service Oriented Architecture (SOA) is an emerging architectural style for developing and integrating enterprise applications. Knorr and Rist [1] define it as a broad, standalone and standards based framework in which services are built, deployed, managed and orchestrated in pursuit of an agile and resilient information technology (IT) infrastructure. This architecture aims to provide enterprise business solutions that can extend or change *on demand* as well as provide a mechanism for interfacing existing legacy applications regardless of their platforms or languages. It is being seen as a new approach to enterprise application integration (EAI). A report by Gartner has predicted that, by 2008, 60% of the enterprise will use SOA as their guiding principle when developing enterprise applications [2]. The Radicati Group also expects the SOA market to reach \$6.2 billion by 2008 [2]. Considering the implications, numerous toolkits, developed by well-known vendors, are beginning to appear in the market, to help the organisations to implement *service orientation*.

On the physical and technical side, Grid Computing (GC) is being seen as providing a sound foundation for SOA. Thus, for the successful implementation of SOA, Grid Computing is becoming a popular

technical infrastructure. The 451 Group expects the Grid technology to move into mainstream applications by 2010 [3].

In the current climate where businesses are required to be agile and flexible and IT managers are being asked to deliver improved functionality while leveraging existing IT investment, SOA and GC are being seen as complementary solutions that, together, will provide the following benefits:

- Increased business agility.
- Better alignment of IT with business needs.
- Enhanced reuse of existing applications, resources and other assets.
- Better scalability and graceful evolutionary development and deployment.
- More effective merging of enterprise applications and computing infrastructures in a heterogeneous technical environment.
- Decreased development and implementation costs.

### 2. SOA

Service Oriented Architecture (SOA) is becoming a popular approach for developing and integrating enterprise applications. It is an organisational and technical framework to enable an enterprise to deliver self-describing and platform independent business functionality providing a way of sharing business functions in a widespread and flexible manner [4]. In this architectural style, the business and technical processes are implemented as *services*. Each service represents a certain functionality that maps explicitly to a step in a business process [5]. In this context, a service is a software component that can be reused by another software component or accessed via a standard-based interface over the network. SOA extends the well-known *client-server* model – where service functionality is organised into loosely coupled components, normally implemented as *web services*.

An important aspect of service-orientation is the separation of service *interface* (the WHAT) from its implementation or *content* (the HOW). The interface provides service identification, whereas, the content provides business logic. In this sense, a service is an exposed piece of functionality, which is governed by the following basic and core principles [6]:

- Services are autonomous, self-contained, loosely coupled, reusable and stateless.
- Services share a formal interface, which is platform independent.
- Services abstract underlying logic that is invisible to outside world.
- Services are composable, allowing logic at different levels of granularity.

Although, there are a number of issues with SOA with respect to integration of services (because of loose coupling of services), interoperability of services (when services reside on heterogeneous hardware systems), security (due to lack of reliability of Internet protocols), governance (as the inherent issues are not well understood), standards (which are still evolving), development tools (which are in early stages of development), as mentioned by Mahmood [7], tremendous progress is being made by agencies such as OASIS [8] and W3C [9].

If implemented correctly, SOA promises the following benefits [10, 11]:

- Loosely coupled applications and location transparency.
- Seamless connectivity of applications and interoperability.
- Alignment of IT with business needs
- Enhanced reuse of existing assets and applications.
- Process-centric architecture.
- Parallel and independent development.
- Better scalability, ease of maintenance and graceful evolutionary changes.
- Reduced costs of application development and integration.
- Reduced vendor lock-ins i.e. reduced dependence on vendors for products and services.

SOA uses a *publish-find-bind-execute* paradigm as shown in Figure 1. The fundamental characteristics of SOA include:

- Service providers – providing *services* that execute business functions.
- Service consumers – using *services* published by service providers.
- Service registry – providing *service descriptions* for use by service consumers.

*Service Providers* build services and offer them via an intranet or Internet. They *register* services with *service brokers* and *publish* them in distributed *registries*. Each service has an interface, known as *contract* and functionality, which is kept separate from the interface. The *service consumers* search for

services (based on some criteria) and, when found, a dynamic *binding* is performed. In this case, the *service* provides the consumer with the *contract* details and an *endpoint* address. The *consumer* then *invokes* the service.

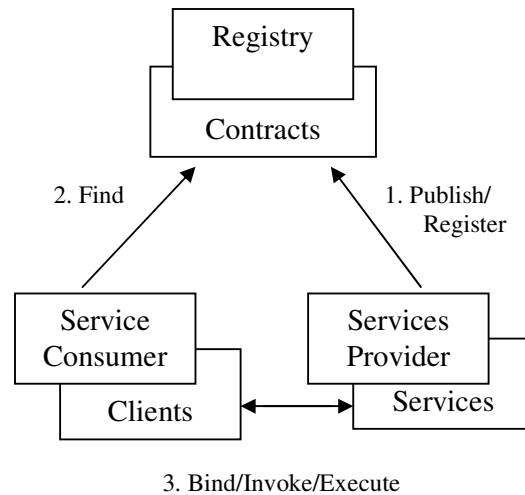


Figure 1: *Publish-Find-Bind-Execute* paradigm

Services, often implemented as Web Services (WS) are delivered using technologies such as eXtensible Markup Language (XML), Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP) and Universal Description Discovery and Integration (UDDI). Numerous tools have also been developed to help organisation to implement SOA. Some of these are: BEA SOA 360, Eclipse SOA Tools Platform, CapeClear SOA Tools, IBM WSDM, Stampede Web 2.0 Services. Mahmood [12] presents an overview of a number of such toolkits.

Implementation of *service orientation* requires unbundling of existing applications, designing self contained platform independent services, combining services to create composite services, implementing on an appropriate technical infrastructure, publishing (so that consumers can find them) and making them available, on demand, to consumers over the Internet or other networks. This also requires the understanding of a new approach to applications development as well as training in new technologies as mentioned above. For a successful transition to a SOA, one can view the SOA life cycle stages as being the following [13]:

- Development: of loosely coupled and reusable application components - exposed as services – to be used by business processes and other applications.
- Discovery: by organising a service directory, to act as ‘yellow pages’ based on an open standard e.g. UDDI.

- Integration: of services with applications and other services, including data transmission services, reliable message delivery mechanism etc.
- Orchestration: of services to *sequence* the required services to fulfil a particular business task using available standards e.g. Business Process Execution Language (BCPL).
- Deployment: of integrated and orchestrated services for the 24-hour-7-day availability using a robust network e.g. enterprise grid.
- Monitoring: of processes in real time and analysis and resolution of issues.
- Management: of the entire processes of development, execution and analysis with respect to process improvement.

### 3. Grid Computing

SOA is an attractive paradigm for the development and integration of enterprise applications. However, if the computing infrastructure used to deploy business and application *services* is not properly organised then the benefits offered by the SOA will not be realised. In this context, Grid Computing (GC) provides an appropriate technical infrastructure, often known as *Computing Architecture* (CA). Here, GC refers to what is commonly known as *Enterprise Grid*, rather than *Scientific Grid*, *Global Grid* or *Utility Computing*. Although, the concept of GC is still evolving, it is the next logical step in distributed computing [14]. It allows uniting pools of physical resources (servers, storage media, computer systems) and enables virtualisation of distributed computing, enterprise data and application resources to create a single system image, allowing seamless access to the enterprise IT capabilities [15] - also providing resilience, better security and more effective management. Grid provides a flexible and cost effective infrastructure that fosters innovation and collaboration.

A grid is a collection of machines, sometimes referred to as *nodes*, *resources*, *members*, *donors*, and *hosts*. Some of these may be assigned to specialist tasks while others may be available for general use. Grids can be built in all sizes, ranging from just a few machines in a department forming a *cluster* of machines (called *intragrids*) to a large group of machines of different types, belonging to different organisations and dispersed over a geographically much wider area (called *intergrids*) [16]. The way the machines are linked together is important as the configuration of a grid can greatly affect the performance, reliability and security of an organization's computing infrastructure. Platforms on a grid will often have different architectures, operating systems, devices and capacities. In the simplest form, a grid would consist of just a few

machines. More advanced grid systems would include a number of services that would provide a number of useful functions e.g.:

- A *job scheduler* - to automatically find and allocate the most appropriate resource depending on task requirements.
- A *management node* - to report the status of nodes on the system so that tasks can be assigned and re-assigned reducing idle time.

Fig. 2 represents a simple GC model. The *Grid Coordinator* acts as a *job scheduler* (combined with a *management node*), which is linked to the *Service Registry* (an element of SOA). The *Services* reside on the *Grid Nodes*.

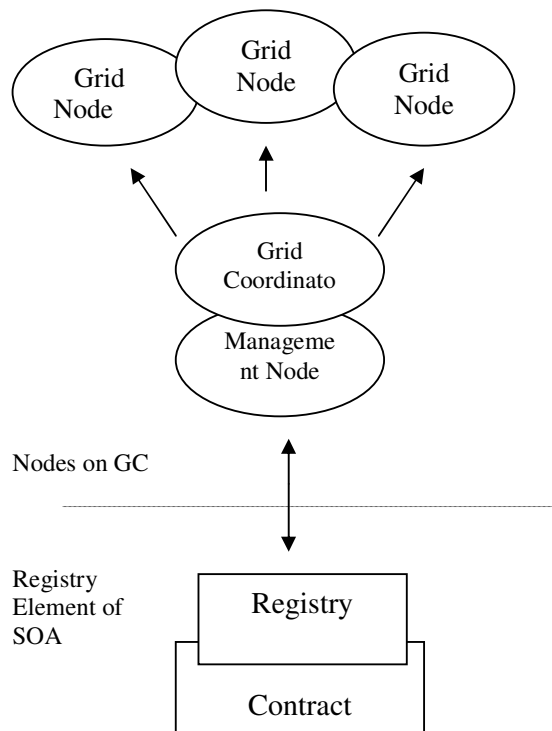


Fig 2: A Simple GC model

An effective Grid infrastructure to support SOA strategies promises the following benefits [17]:

- Application performance – in terms of speed and response time.
- Resilience and reliability – in the event of system error and application failure.
- Flexibility and API independence – in terms of supporting a variety of clients that can be quickly virtualised, including Java, .NET, SOAP, C++, etc.
- Usage-based accounting and control – in terms of global management of services,

- including policy-driven service, resource assignment and workload distribution rules.
- Dynamic provisioning – because of easily adaptable load balancing mechanisms.
  - Rapid development and deployment – because of inherent nature of service orientation and virtualisation.
  - Costs reduction – as a result of the above advantages.

However, there are a number of inherent difficulties and issues, as listed below:

- Security – with respect to the sensitive data belonging to other organisations.
- Policies – with respect to the deployment, use and management of the infrastructure.
- Dedicated machinery for some applications - to increase the quality of service.
- Dedicated communications to connect certain specialist nodes.
- Planning – may become much more problematic on a global scale.
- Tools for debugging and measuring – as they are in early stages of development.
- Grid configuration – as this can affect the performance, reliability and security.
- Standards – as they are still evolving.
- Flexibility of grid applications – not easy when a wide range of platforms and environments are used.

#### 4. SOA and Grid Computing

When implementing SOA, technical framework based on GC provides the optimum foundation. Likewise, SOA can provide a strong basis for GC. Thus, SOA and computing infrastructure virtualisation techniques such as a Grid are highly synergistic. From the SOA perspective, grids offer an exceptional model for the distribution of information and resources, which is a key feature of SOA. From the grid perspective, SOA offers alternative but flexible methods for adjusting the architecture of grid solutions and making them more transparent and supported on a wider range of platforms and environments. The two solutions (SOA and GC) share the same objectives, although approach is different. Since, SOA improves software mobility, it is better suited for dynamic environments such as Grid infrastructure [18].

Although, there are issues as mentioned above, various standards bodies and other agencies such as Open Group Services Infrastructure (OGSI), Global Grid Foundation (GGF), Enterprise Grid Alliance (EGA), World Wide Web Consortium (W3C) and Organization for the Advancement of Structured Information Standards (OASIS) [8, 9, 19] are making efforts in this area to define the

required standards. Numerous vendors have also developed toolkits e.g.:

- The OGSA (Open Grid Services Architecture) provides a mechanism to describe grid services. This architecture defines grid service behaviours, service description mechanisms, and protocol binding information by using Web services as the technology enabler. The core technologies used (XML, WSDL, SOAP, UDDI) are the same as those in SOA.
- The IBM Globus toolkit, provided as part of the IBM Grid Toolbox is an open source software providing resource management, information services and data management for building grid tools, services and applications [20].
- The IBM Grow also provides easy to deploy and integrated solutions to develop enterprise grids [21].

The implementation requires a different development approach – a shift from writing software to assembling and integrating services. As suggested by Brown [22], merging Grid applications with SOA principles requires the following:

- Migrating applications to a service model in terms of individual operations.
- Dividing grid applications into smaller, discrete components.
- Ensuring that nodes and controller are capable of independent working – not relying on database servers or other resources.

Similarly, merging SOA applications with Grid principles requires the following [22]:

- Elimination of areas of application where they rely on a single host or environment to operate.
- Incorporating statistical and monitoring information into SOA services.

Services, the basic building blocks of SOA, implemented as web services, are dynamic entities. They need to be started and stopped on demand. When new services come online, the middleware in GC can easily recognise them and deploy them.

#### 5. Conclusions

SOA is a framework that allows building, deployment and integration of services and functions for an enterprise's IT resources, applications and business process flows [21]. It promises to reduce complexity, increase flexibility and provide enhanced streamlining of business processes. In this

approach, business and customer functions are defined as loosely coupled and platform independent *services*, which are *published in directories* and usually implemented as *Web Services*.

To realise the benefits offered by a SOA, enterprises need to establish a technical infrastructure that is flexible, responsive, scaleable and dynamic in nature. This paper suggests that Grid Computing (GC), in the form of an *Enterprise Grid* provides such a dynamic infrastructure. Here, a *management node*, as part of the grid, ensures that unutilised resources are kept to a minimum and that individual projects can be easily prioritised and resourced over the grid.

The two solutions (SOA and GC) use the same technologies (XML, WSDL, SOAP and UDDI) and share the same objectives, although approach is different – SOA virtualises services and processes, whereas GC virtualises hardware resources.

This paper discusses the synergies between SOA and GC and aims to provide some useful information for the benefit of large enterprises that wish to embark on the development and implementation SOA based on Grid Computing to gain further efficiency and an edge over their competitors.

## 6. References

- [1] Knorr E. and Rist O., 10 Steps to SOA, *Info World - San Mateo*, Vol. 27, Issue 45, pp 23-35, Nov. 2005
- [2] Foster I., Grid's Place in the SOA, *Computer World*, Available at: [www.techworld.com/opsys/features/index.cfm?FeatureID=1029](http://www.techworld.com/opsys/features/index.cfm?FeatureID=1029)
- [3] 451 Group, Predictions about Grid Computing Evolution, Available at: <http://www.computerworlduk.com/technology/servers-data-centre/grid-utility/in-depth/index.cfm?articleid=704&pn=2>
- [4] Cartwright I. and Doernenburg E., Time to jump on the SOA bandwagon, *IT Now*, British Computer Society, May 2006
- [5] Groves D., Successfully planning for SOA, *BEA Systems Worldwide*, 11 Sept. 2005
- [6] Erl T., Core principles for service-oriented architectures, August 2005. Available at: [www.Looselycoupled.com/opinion/2005/erl-core-infr0815.html](http://www.Looselycoupled.com/opinion/2005/erl-core-infr0815.html)
- [7] Mahmood Z., Service Oriented Architecture: Potential Benefits and Challenges, *Proc. of 11th Int. WSEAS Multi-Conf on Computers*, Vol 4, pp 496-500, Crete, Greece, July 2007
- [8] OASIS, Available at: <http://www.oasis-open.org/who/>
- [9] W3C, Available at: [www.w3.org](http://www.w3.org)
- [10] Sonic Software Solutions, Service oriented architecture, Available at: [www.sonicsoftware.com/solutions/service\\_oriented\\_architecture/index.ssp](http://www.sonicsoftware.com/solutions/service_oriented_architecture/index.ssp)
- [11] Johnson B., The benefits of service oriented architecture, *Objectsharp Consulting*, Available at: <http://objectsharp.com/cs/blogs/bruce/pages/235.aspx>
- [12] Mahmood Z., Frameworks and Tools for Building Enterprise Information Architectures, *Proc. of IBIMA Int. Conference*, Bonn, Germany, June 2006
- [13] Mahmood Z., Service Oriented Architecture: A New Paradigm for Enterprise Application Integration, *Proc. 11th Int. WSEAS Multi-Conf on Computers*, Vol 4, pp 490-495, Crete, Greece, July 2007
- [14] IBM, Introduction to Grid Computing, Available at: [www.ibm.com/developerworks/grid/newto/](http://www.ibm.com/developerworks/grid/newto/)
- [15] IBM, What is Grid Computing, Available at: [www-03.ibm.com/grid/about\\_grid/What\\_is.shtml](http://www-03.ibm.com/grid/about_grid/What_is.shtml)
- [16] Berstis, Viktors, Fundamentals of Grid Computing, *Redbooks Papers, IBM*, 2002
- [17] Vizzini K., Grid and the Migration to SOA, Available at: <http://www.gridtoday.com/grid/396739.html>
- [18] Castro F. and Jain P., SOA and Grid: A Successful Marriage of Two Paradigms, Available at: [www.gridtoday.com/grid/1650257.html](http://www.gridtoday.com/grid/1650257.html)
- [19] Global Grid Forum, Available at: <http://www.ggf.org/>
- [20] IMB, Introduction to Grid Computing with Globus, *IBM.com/redbooks*, 2004, Available at: <http://www.redbooks.ibm.com/redbooks/SG246895/wwhelp/wwimpl/java/html/wwhelp.htm>

[21] IBM, Grid Computing: Past, Present and Future, *IBM*, June 2006

[22] Brown M. C., Build Grid Applications Based on SOA, 2005,  
Available at: <http://mcslp.com/gridpdfs/gr-soa.pdf>

Copyright © 2009 by the International Business Information Management Association (IBIMA). All rights reserved. Authors retain copyright for their manuscripts and provide this journal with a publication permission agreement as a part of IBIMA copyright agreement. IBIMA may not necessarily agree with the content of the manuscript. The content and proofreading of this manuscript as well as and any errors are the sole responsibility of its author(s). No part or all of this work should be copied or reproduced in digital, hard, or any other format for commercial use without written permission. To purchase reprints of this article please e-mail: [admin@ibima.org](mailto:admin@ibima.org).