



Twitter DM Client Desktop Remote Service Offline

Prabhjot Kaur, Rubal Chaudhary

Indo Global College, Chandigarh,
India

Abstract- Over the past decade, classic client side applications with Model-View-Controller (MVC) architecture haven't changed much but become more complex. In this paper, the problem is that when we need a data feed from another website we can't use Grid for that, as Functional support make it heavy to run which becomes Time Complex in order to run this on Local Service. Web Services only use XML for Response making hard for developer who doesn't know about XML. The base paper only used HTTP method but if we want to use the post form method then these are not supported in Web Service. Performance data suggests that this scheme of building application around messages is a practical architecture for the next generation Web application client.

Keywords- Grid Computing, Offline Data Structure, AssemblyDLL Structure, Twitter Services

I. INTRODUCTION

Web Services are becoming an increasingly important feature of Internet and Grid systems. They support a loosely coupled service oriented architecture that builds on previous distributed object architectures like CORBA, Java RMI, and COM to provide scalable interoperable systems. The broad applicability of this approach includes enterprise software, e-Science and e-Business. Correspondingly there are a growing number of powerful tools that are available for building, maintaining and accessing Web Service-based systems. Grid computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application. Grid computers also tend to be more heterogeneous and geographically [2] dispersed (thus not physically coupled) than cluster computers. Although a single grid can be dedicated to a particular application, commonly a grid is used for a variety of purposes. Grids are often constructed with general-purpose grid middleware software libraries.

Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly. One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers, sometimes up to many thousands. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale clusters.

The size of a grid may vary from small—confined to a network of computer workstations within a corporation, for example—to large, public collaborations across many companies and networks. "The notion of a confined grid may also be known as an intra-nodes cooperation whilst the notion of a larger, wider grid may thus refer to an inter-nodes cooperation". These tools include portals that allow user frontends to Web Services. This model for user interaction has new standards like portlets with WSRP (Web Services for Remote Portlets) [1] and the Java Specification Request JSR168 [2] supporting lightweight interfaces to the backend resources. This architecture shown in fig.1 implements the Model-View-Controller or MVC [3] [4] architecture with a clean message based interface partially specified by the portlet standard.

In section 2, we describe the other methods used in the literature. In section 3, methodology to solve the problem is proposed. Fourth section describes result.

II. LITERATURE SURVEY

Due to the difference of document object models that implemented by earlier versions of major browsers, web developers access HTML document using JavaScript had to write wrapper code to reconcile the incompatibility between Netscape and Internet Explorer. In 1998, W3C proposed Document Object Model (DOM) level 1 specification that defines "a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents." [8] DOM level 2 standard further specifies a generic event model [10].

The business model for Grids (at least that found in academia or government labs) is project-oriented in which the users or community represented by that proposal have certain number of service units (i.e. CPU hours) they can spend. For example, the TeraGrid operates in this fashion, and requires increasingly complex proposals be written for increasing number of computational power. The TeraGrid has more than a dozen Grid sites, all hosted at various institutions around

the country. What makes an institution want to join the TeraGrid?[3] When an institution joins the TeraGrid with a set of resources, it knows that others in the community can now use these resources across the country. It also acknowledges the fact that it gains access to a dozen other Grid sites. This same model has worked rather well around the globe, giving institutions incentives to join various Grids for access to additional resources for all the users from the corresponding institution. In order to support the creation of the so called “Virtual Organizations”—a logical entity within which distributed resources can be discovered [5] and shared as if they were from the same organization, Grids define and provide a set of standard protocols, middleware, toolkits, and services built on top of these protocols [6]. Interoperability and security are the primary concerns for the Grid infrastructure as resources may come from different administrative domains, which have both global and local resource usage policies, different hardware and software configurations and platforms, and vary in availability and capacity.

Software as a Service (SaaS) [2] delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model. Salesforce is an industry leader in providing online CRM (Customer Relationship Management) Services. Live Mesh from Microsoft allows files and folders to be shared and synchronized across multiple devices. Although Clouds provide services at three different levels (IaaS, PaaS, and SaaS), standards for interfaces to these different levels still remain to be defined. This leads to interoperability problems between today’s Clouds, and there is little business incentives for Cloud providers to invest additional resources in defining and implementing new interfaces. As Clouds mature, and more sophisticated applications and services emerge that require the use of multiple Clouds, there will be growing incentives to adopt standard interfaces that facilitate interoperability in order to capture emerging and growing markets in a saturated Cloud market.

III. PROBLEM FORMULATION

An integrated development environment to establish web service in grid computing is studied based on the types and method related to SOAP, UDDI, WSDL Importing or running a web service on a local host or live URI based. What is needed to do with the web service is to import it from a Network adapter or a Dedicated IP and then import it into your solution. The Problem is that when we need a data feed from another website we can’t use Grid for that, as Functional support make it heavy to run which becomes Time Complex in order to run this on Local Service.

Web Services only use XML for Response making hard for developer who doesn’t know about XML. The base paper only used HTTP method but if we want to use the post form method then these are not supported in Web Service. We need to get its license if to use. That makes a problem of open source Development also not compatible with MVC frame-work.

Objectives:-

- To develop a web service in a web API framework which handles the Business logic on HTTP method and also Supports GET Post implementation of Methods.
- Provide support for all Open Source Technology Development by using this Service Architecture.
- To support a JSON, AJAX calling rather than to use XML for every response and request handle by HTTP module.
- Making compatibility for every MVC framework architecture supported in various language.

IV. METHODOLOGY

The present research work is proposed to be completed in three stages which have to be preceded in parallel fashion, as described below

Web Services

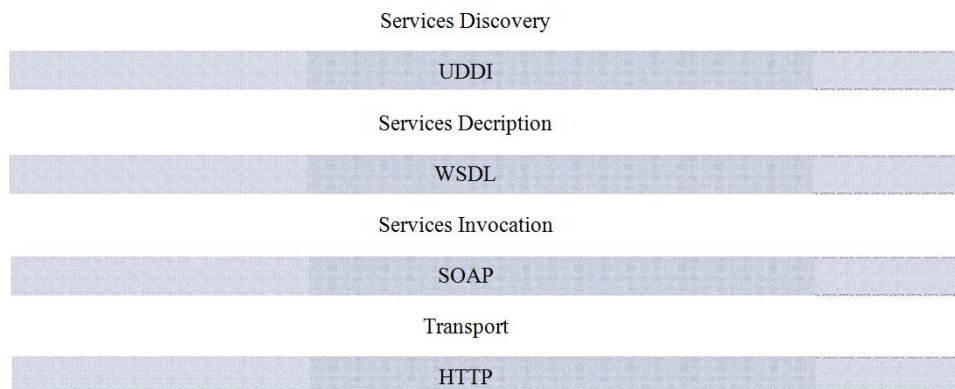


Fig 1: Block Diagram

V. RESULTS

The Application of Grid Computing is developed which is reliable. Figure 1 shows Twitter DM Client Graphical User Interface

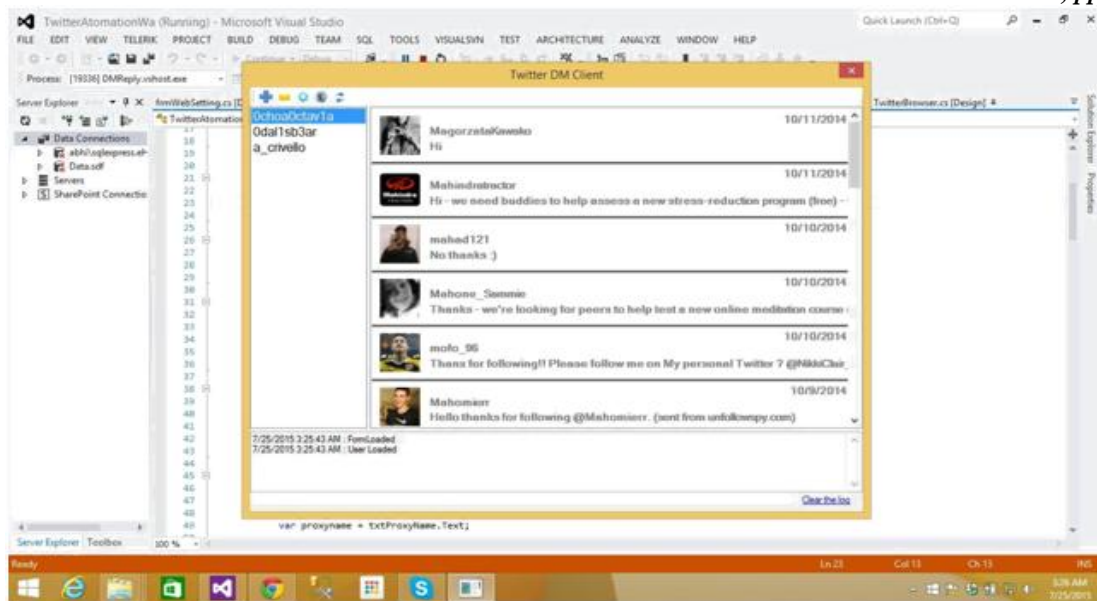
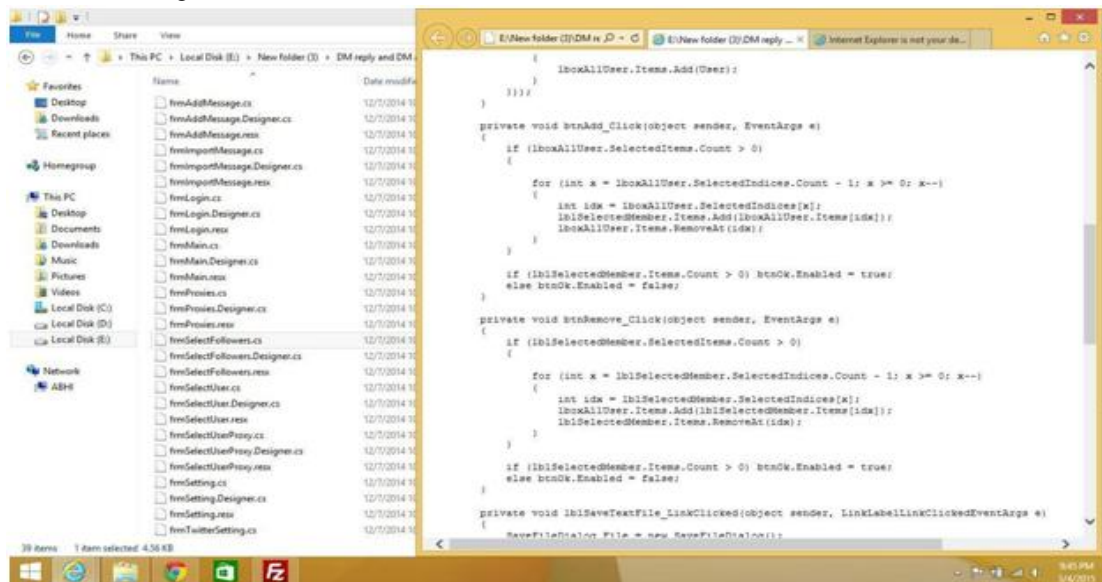


Fig 2: GUI of Twitter DM Client

Figure 2 shows the coding of the twitter DM client



VI. CONCLUSION

The application of Twitter DM Client using Grid Computing is developed and is working as a desktop application. The application uses Web services to access the twitter DM client.

REFERENCES

- [1] S. Ahuja, N. Carriero, and D. Gelernter. "Linda and Friends", IEEE Computer 19 (8), 1986, pp. 26-34.
- [2] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke. "Data Management and Transfer in High Performance Computational Grid Environments", Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.
- [3] Amazon Elastic Compute Cloud (Amazon EC2), <http://aws.amazon.com/ec2>, 2008.
- [4] Amazon Simple Storage Service (Amazon S3), <http://aws.amazon.com/s3>, 2008.
- [5] B. Bode, D.M. Halstead, R. Kendall, Z. Lei, W. Hall, D. Jackson. "The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters", Usenix, 4th Annual Linux Showcase & Conference, 2000.
- [6] J. Brodtkin. "Gartner: sevencloud-computing security risks", <http://www.networkworld.com/news/2008/070208-Cloud.html>, 2008.
- [7] R. Buyya, D. Abramson, J. Giddy. "Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid", IEEE Int. Conf. on High Performance Computing in Asia-Pacific Region (HPC, ASIA) 2000.
- [8] R. Buyya, K. Bubendorfer. "Market Oriented Grid and Utility Computing", Wiley Press, New York, USA, 2008.

- [9] A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi, “Storing and Querying Scientific Workflow Provenance Metadata Using an RDBMS”, in Proc. of the 2nd International Workshop on Scientific Workflows and Business Workflow Standards in eScience, in conjunction with e-Science, pp.611-618,Bangalore, India December,2007.
- [10] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke. “The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets”, Jnl. of Network and Computer Applications,,23:187-200, 2001.
- [11] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke. “From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution,” March 5, 2004.
- [12] S. Davidson, S. Boulakia, A. Eyal, B. Ludäscher, T. McPhillips, S. Bowers, M. Anand, J. Freire, “Provenance in Scientific Workflow Systems”, IEEE Data Eng. Bull. 30(4): 44-50 (2007).
- [13] J. Dean, S. Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters”, Sixth Symposium on Operating System Design and Implementation(OSDI04),2004.
- [14] C. Dumitrescu, I. Raicu, I. Foster. “The Design, Usage, and Performance of GRUBER: A Grid uSLA-based Brokering Infrastructure”, Intl.Journal of Grid Computing,2007.
- [15] Eucalyptus, <http://eucalyptus.cs.ucsb.edu/>, 2008.