IT Service Provisioning by Passing Hints in Module Interfaces

Nitesh Naik[1], John P Veigas[2], K Chandrasekaran[3]
Department of computer science and Engineering
National Institute of Technology Karnataka
Suratkal, India
{[1]naiknitesh22gec05,[2]veigasjp, [3]kchnitk}@gmail.com

***Abstract*— Communication is a crucial aspect in the domain of IT service provisioning, where data is being transmitted from the sending entity to the receiving entity by the interaction of applications via an interface. When the data is transferred, large amount of time is consumed for its computation by the service at the receiving side. When two applications interact via an interface the communication can be made effective by enhancing the IT service provisioning system with inclusion of web services by some faster means. In this paper we discuss how IT service provisioning can be handled by making use of hints using web services that can be passed by the modules within the application via interfaces from a client to the service, which can avoid expensive computation by the service. The hint is passed in its request and if correct, a service can avoid the computation and communication overhead leading to better system performance. If the message or hint is incorrect it leads to erroneous computations. Passing hints between the modules within an application should take place in an effective, non-erroneous way by imparting quality and less computation by the service involved between two communicating entities contributing towards the promises of services provided by cloud computing.**

Keywords- Communication, Hints, Interface, Modules, Web services

I. INTRODUCTION

In any processing environment there exist multiple applications to carry out a particular operation. An application consists of different modules and modules contain different procedures. Consider an application A1 consisting of modules M1, M2, M3 each with procedure P1, P2, P3.If the procedures within the modules wants to communicate this can be accomplished by communicating procedures within the modules. This scenario is possible if the modules want to communicate within the application itself. But when two different applications wants to communicate .i.e modules consisting of procedures in application1 wants to interact with procedures of modules of application2, when such scenario occurs it leads to inter-application communication.

When two applications wants to communicate that is one is asking for a service and other is providing the service. This task can be accomplished by making use of IT service provisioning. By the use of IT service provisioning for the above scenario and handling the drawbacks of IT service provisioning will be discussed throughout this paper which can be efficiently handled by the use of passing hints between the modules of the applications involved in communication via an interface.

II. IT SERVICE PROVISIONING

IT service provisioning is known as Information Technology service provisioning. The terms IT service and provisioning are important concepts for understanding the promises of cloud computing. For example, an IT service could be an e-mail account, remote PC backup, data set storage, course management system, help desk, high-performance computing (HPC) job, multi-core programming code optimization consultation, licensed software distribution, or other technology that enables education and research. An IT service also has elements of contracting, legal and policy compliance, evolution and improvement, measurement regarding cost and effectiveness, and user support as essential components. Provisioning defines the means through which these components of an IT service are made available. An academic department may have IT staff that buy hardware, manage software systems, and provide assistance. Alternatively, IT services may be provisioned through a shared-service unit of a school or a campus, through a consortium, or through a contract with a commercial vendor[1]. Cloud computing proposes new means to provision some familiar and some new types of IT services..

A. How IT service provisioning takes place

The need for better control over IT budgets and IT investments, aligning business and IT, and increased compliancy, requires organizations to design and implement some form of IT governance. As a part of research, METRI has developed an IT Governance Framework (Figure-1) to support IT governance. Their offerings are structured alongside this framework.

Figure -1. Framework of IT Governance depicting IT service provisioning as one phase developed by METRI

Figure-1 depicts the IT governance designed by METRI[2] which emphasizes on IT service provisioning as one of the phase of the IT Governance. IT Governance is needed for better control over IT budgets and IT investments, aligning business and IT, and to handle increased compliancy in an organization. This framework supports its clients in realizing

978-0-7695-4723-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ICACC.2012.39

Authorized licensed use limited to: NATIONAL INSTITUTE OF TECHNOLOGY SURATHKAL. Downloaded on April 09,2021 at 03:58:54 UTC from IEEE Xplore. Restrictions apply.

and optimizing effective IT governance. We will be dealing with IT service provisioning using web services as main aspect of our research.

B. Perspective set by IT service provisioning

The main perspective or aim of IT service provisioning is "Customer focus requires a transparent Service Catalogue" Nowadays, business units demand transparency from IT in what they deliver and the associated costs. To satisfy this demand, IT department needs a good service catalogue which is customer focused. The catalogue should not be filled with technical descriptions, but with clear service descriptions which are understandable for the customers, i.e. written in the "customer's language". It enables them not only to determine what present services are delivered, but also to describe them with a customer focus. This also helps in pricing the services by providing not only the customer focused service catalogue but also the inclusion of underlying technical description to the customer. The IT service provisioning should be completely transparent to the customer using the required services.

C. Delimitations of IT service Provisioning

As we are already aware of the main aim of IT service provisioning and its contributions in the computing field. However there are certain things we need to be cautious about. There are some down sides in IT service provisioning, but users should not depend too heavily on these services due to following delimitations. Here we have discussed some of them:

1. *Security & Privacy*

The biggest concerns in IT service provisioning is security and privacy. Users might not be comfortable handing over their data to a third party. This is an even greater concern when it comes to companies that wish to keep their sensitive information on servers. While most service vendors would ensure that their servers are kept free from viral infection and malware, it is still a concern considering the fact that a number of users from around the world are accessing the server.

Privacy is another issue with the servers providing service. Ensuring that a client's data is not accessed by any unauthorized user is of great importance for any service. To make their servers more secure, service vendors have developed password protected accounts, security servers through which all data being transferred must pass and use data encryption techniques. After all, the success of a service depends on its reputation, and any sign of a security breach would result in a loss of clients and business.

2. *Dependency (loss of control):*
   - Quality problems with ITSP. .No influence on maintenance levels and fix frequency when using services from a Service Provider.
   - No or little insight in service provider contingency procedures. Especially backup, restore and disaster recovery.
   - No easy migration to another service provider.
   - Measurement of resource usage and end user activities lies in the hands of the service provider.
   - Tied to the financial health of another Company.

3. *Cost.*

Cost can be higher. While in the long run, hosting is a lot cheaper than traditional technologies, the fact that it's currently new and has to be researched and improved actually makes it more expensive. Data centers have to buy or develop the software that'll run the service, rewire the machines and fix unforeseen problems which are uncertain. This makes the primary service offers more expensive. Like in all other industries, the first customers pay a higher price and have to deal with more issues than those who switch later for their given purpose.

D. Proposal: Improving the demerits of IT service provisioning by making use of hints

Passing data is a very challenging task between the two entities in IT service provisioning. Passing data between a client and a service plays an important role for maintaining effective computation and communication time involved in this operation. The main aim of passing data involves that the syntax and semantics of the data are interpreted properly by the service provided by the provider. When this communication takes place the service experiences large amount of delay in its computation. To improve this we can make use of Hints within the data when the message is passed in the domain of service provisioning. If the hint is correct it will lead to the decrease in computation time by making use of the hint by the service.

1. What are hints?

The definition of a hint [3] suggests that a variant in which information is passed that is guaranteed to be correct and hence requires no checking. A hint is information which is also known as a tip. Passing hints has become an important measure when we are dealing with client and service based exchange. The service takes ample amount of time for its computation, so this overhead can be reduced by making efficient use of correct hints that will improve the performance of the system in terms of speed and memory access time. By making use of a correct hint the time taken by the service doing computation can be utilized to carry out some other useful operation contributing towards the enhancement in the services of cloud computing.

2. Passing Hints for Informed Prefetching between module interfaces

Hints are used within the data which is passed to the service in its request, we make use of access pattern hints that should avoid long access latencies, incorporating hints in data helps us to inform the application about future accesses herby reducing useful computation time. The extraction and collection of such accurate and timely hints should be feasible as it contributes to reduction in overhead of overall computation time by the service involved.

Authorized licensed use limited to: NATIONAL INSTITUTE OF TECHNOLOGY SURATHKAL. Downloaded on April 09,2021 at 03:58:54 UTC from IEEE Xplore. Restrictions apply.

3. *Exploiting Hints*

Hints are stored in the cache memory of the service involved. Hints enable informed prefetching to overlap those accesses for which caches are ineffective with useful computation. Hints are basically use in exploiting memory access time. Hints specify the set of files needed for the program to work effectively that can minimize latency penalties by pre-loading the cache of the service provider.
As we said all the hints are stored in the cache memory of the service involved, the cache will be searched for the resources and if the resources are not available to the entire cache then We can make use of access order and access pattern hints that will allow to read ahead the non - sequential access and early release of precious memory resources. Files that are accessed only once can be pre-fetched by making effective use of access pattern hints .For example if we consider the use of files that provide the service of bulletin boards that will contain the hint to read and fetch the next message of interest to be displayed within the bulletin board. [4]

4. *Obtaining Hints and its applications to overcome service provisioning*

The availability of accurate and timely hints contributes towards success of smart prefetching. In this we will expose the applications of hints. In simple terms a programmer notifies the I/O system, through a hint, of a sequential access pattern. The programmer can use a hint based mechanism to determine what to read-ahead, so that data can be pre-fetched in a programmer defined random access pattern. An important beneficiary of this approach will be the large scientific programs that execute alternating row and column access patterns on huge matrix data files [5]. At least one of these access patterns will not be sequential in the file's linear storage, but is easily and obviously specified by a programmer. that will allow the programmer to explicitly and asynchronously pre-fetch sequential and non-sequential data , fewer programs than we might expect actually do this [5] because it expands virtual memory requirements in systems where memory is a critical resource, as it is not always portable, and as it requires more than a few lines of code. In contrast, Programmer can use short and simple hints for system-level pre-fetching that can easily compensate for reluctance on the part of programmers.

By providing an interface to the programmer to pass hints, we can extend optimizing compiler techniques to also issue hints. By concealing file names and access patterns in data structures, many programs that access files simply, file name comes as arguments, access calls are made from previous offsets or loop indices[5]. Hints can be extracted automatically by making use of such straightforward styles. Hints can be considered as a process' or thread's parent code. For example, a software compilation tool such as "make" has extensive knowledge of the tasks that are going to be executed in the immediate (and sometimes not so immediate) future. "make" has the capability to internally recognize the functions it spawns, it can give hints about the source, object, and library files that will be needed for further processing.

Another example is a user shell. A shell is needed to perform a filename completion or wildcard expansion, it could pass these filenames as hints about what will soon be accessed. The user shell could recognize certain program names on the command line it has constructed for execution in the design of more accurate hints. By recognizing the programs file arguments can be named in hints before or in parallel with programs invocation. To recognize and issue hints for a specific program the user shell program specific information could be made available with file of hints registered during program installation or a specially formatted string in the first page of programs binary which can be invoked in parallel also [6].

Finally, and perhaps most importantly, we must find a good mechanism to take hints from high levels of abstraction and deliver them to lower levels. Some hints may be best exploited by database code, some best processed by cache management code, some targeted at device drivers, and some useful at many or all lower levels. We need interfaces and abstractions for distributing this type of performance optimization information without disturbing the benefits of layering and modularity in modern applications and operating systems. From a practical point of view, establishing the right interface is essential if tools as invasive of other parts of the system as are ours, are to survive in today's standards driven environments. We will be discussing how correct hints can be applied in service provisioning for carrying out specific and critical computations by the service involved leading towards fast and efficient processing of system data.

III. OUR STUDY AND ANALYSIS

Organizations can have multiple provisioning systems that exchange information about the modification of user records. In addition, there can be applications that interact with multiple provisioning systems. Connectors can enable the interaction between two provisioning systems or between an application and a provisioning system so that each application synchronizes with the other. However, configuring a custom connector for each combination of these systems leads to a lot of overhead.

Different organizations have multiple provisioning systems that exchange information about the modification of user records. There are applications which interact with multiple provisioning systems. Connectors can enable the interaction between two provisioning systems or between an application and a provisioning system so that each application synchronizes with the other. However, configuring a custom connector for each combination of these systems leads to a lot of overhead. For this purpose we propose the notion of hints which can be implemented in the request which is being sent to the service provider during the communication. Hints can be passed in the request for reducing the communication overhead.

SPML Web services provide a standard means of interoperability between different software applications, running on a variety of platforms and/or frameworks [7].

Authorized licensed use limited to: NATIONAL INSTITUTE OF TECHNOLOGY SURATHKAL. Downloaded on April 09,2021 at 03:58:54 UTC from IEEE Xplore. Restrictions apply.

The SPML Web Service sends and receives SPML requests in the form of SOAP messages. The SPML model consists of the following entities that participate in an end-to-end provisioning scenario. As an addition in this request we can pass hints from the SPML client to the SPML web service which can reduce the overhead when the computation is carried out by the service. By passing the hint in the request the communication can be improved and service can be enhanced. Following are the components of the SPML web service [7].

The implementation of the SPML protocol allows for the reliable exchange of provisioning requests and a model on which you can build a more complex application-level provisioning functionality. SPML is the language of exchanging the management requests used by provisioning systems to manage and control an identity. Since SPML is a standard the above idea can be implemented directly.

IV. WORKDONE AND RESULTS

We are not making use of SPML as it is a standard for service provisioning applications as described; instead we are overcoming it by using web services for passing hints. To demonstrate, we created a simple currency converter web service using the ASP.NET 4.0. In our implementations we are making use of Microsoft provisioning system in which we make use of passing hint mechanism between modular interfaces by using this simple currency converter with ASP.net. In our work the service provisioning is only allowed if the user is subscribed to that particular service otherwise it will not allow the system to use the web service. In this implementation authentication details are passed as hints in the SOAP header to the web service provider, that will check the authenticity of the user and if the user is authentic then only it will allow the web service to execute the operation and return the required result. If the user is not authenticated then access will be denied to the user as shown in figure-2. Here the hints can be used to check the authenticity of the user apart from parameters of the web service.

Figure 2- shows the consumer application of web service for authentic user with the application of hints

The hints are passed in the header of the SOAP Envelope which is shown in figure-3. The client consuming the web service is sending the hints along with the web service parameters in the SOAP header. The web service receives the hint and performs the processing.

```
POST /CurrencyService.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <soap12:Header>
    <AuthSoapHdr xmlns="http:///MyExample.org/">
      <myStrUsername>string</myStrUsername>
      <myStrPassword>string</myStrPassword>
      <Created>dateTime</Created>
      <Duration>int</Duration>
    </AuthSoapHdr>
  </soap12:Header>
  <soap12:Body>
    <GetExchangePrice xmlns="http://MyExample.org/">
      <Amount>float</Amount>
      <FromCurrencyName>string</FromCurrencyName>
      <ToCurrencyName>string</ToCurrencyName>
    </GetExchangePrice>
  </soap12:Body>
</soap12:Envelope>
```

Figure 3- SOAP header with hints

V .CONCLUSION

The drawbacks encountered in IT service provisioning for providing a particular type of service can be improved by making use of hints. To make the transfer of data more richer, we can incorporate hints in the request between the interface of the modules of that given application .If correct hints are passed the service takes less amount of time in computation hereby contributing towards less overhead on the service leading to better system performance. Hints can be embedded in the request of the SPML interface between the SPML client and SPML web service provisioning application which is a standard developed and also by making use of web services as demonstrated above that will make the web service more effective in performing its operation for the desired request from the client. IT service provisioning can be enhanced by using this approach providing better and efficient exchange between the client and the service involved by providing better service to the customers.

REFERENCES

[1] Provisioning Above-Campus IT Services: Supply and Demand http://www.educause.edu/EDUCAUSE+Review/EDUCAUSEReviewMagazineVolume44/ProvisioningAboveCampusITServi/185223.

[2] IT Service Provisioning, [online]. Available: www.metrigroup.com/governance/it-service-provisioning/.

[3] George Varghese, Network *Algorithmics An Interdisciplinary approach for designing fast networked devices*. Morgan Kaufmann publishers, 1st Edition, 2004

[4] Miller, E., "Input/Output Behavior of Supercomputing Applications," University of California Technical Report UCB/CSD 91/616, January 1991, Master's Thesis.

[5] R. Hugo Patterson, Garth A. Gibson, Eka Ginting, Daniel Stodolsky, Informed Prefetching and Caching, [online]. Available: http://www.cs.columbia.edu/~nieh/teaching/ e6118_s00/ papers/p79-patterson.pdf.

[6] Garth A. Gibson†, R. Hugo Patterson‡, M. Satyanarayanan. Disk Reads with DRAM Latency School of Computer Science Dept. of Electrical and Computer Engineering Carnegie Mellon University5000 Forbes Avenue Pittsburgh, PA 15213-3890 in Third Workshop on Workstation Operating Systems, Key Biscayne, FL, April, 1992, pp. 126-131.

[7] SPML Web Service [online]. Available: http://docs.oracle.com /cd/E10391_01/doc.910/e10366/spml.htm#insertedID1

[8] Darko Markulin, Kristina Musa,Proposal for specification of complex processes for telecommunication service provisioning in existing IT environment, IEEE conference MIPRO 2010, May 24-28, 2010, Opatija,Croatia,340-343

Authorized licensed use limited to: NATIONAL INSTITUTE OF TECHNOLOGY SURATHKAL. Downloaded on April 09,2021 at 03:58:54 UTC from IEEE Xplore. Restrictions apply.