Transactions for grid computing

by

Antoni Segura Puimedon

Helsinki November 10th, 2009

Advanced e-Business Transactions for B2B-Collaboration seminar

UNIVERSITY OF HELSINKI

Department of Computer Science

Table of Contents

1.- Introduction....4
2.1.- Grid Computing....6
2.2.- On-demand computing....7
2.3.- Peer-to-Peer....8
2.4.- e-Business....9
2.5.- Acid transactions....9
3.- The inadequacy of the traditional transaction model for Grid Computing....11
4.- Proposed solutions to the transaction model for Grid Computing....14
4.1.- Peer-to-Peer transaction processing....14
4.2.- GridTP....17
4.3.- Trust Management....18
5.- Conclusions....20
6.- References....22

Abstract

In the computing reality of this first decade of the XXI century, such as in about all other aspects of reality, the defining word is globalization, which in computational terms translates the traditional in-house supercomputer clusters of yesteryear into grids of computers scattered all over the world. In this literature review paper, the discussion is centered into the different approaches that have been proposed for providing a transactional process model for grid computing entities. Concretely, the argumentation will revolve around the convenience of adapting current transaction solutions[1], incorporating control principles of other technologies with the same distributed background Grid has[2] and devising new models[9] for a more adequate transaction-like control.

1.- Introduction

In the recent years, the reality of computing has shifted, and is still shifting indeed, from an internal model in which high demanding computations were handled by isolated or lowly networked supercomputers generally performing all the task, towards a model branded *Cloud Computing*[1][3] in which several companies, research groups or even individuals perform tasks over a more ethereal medium that consists of geographically and purposely scattered computers or supercomputers that belong to different organizations and perform different parts of the task.

It is in this new context of an Internet Cloud with increasingly distributed and specialized computing, specially those instances with high demands on resources, where *Grid Computing*[2] becomes of

1 "A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers."[4], extended in [6]

2 "A Grid [5] enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large scale resource-intensive problems in science, engineering, and commerce."[4]

essential relevance due to the inconvenience, and sometimes infeasibility, of managing and performing huge calculations or processes by single entities that are not interested in, or simply cannot afford, carrying them by their own means.

This preponderance of *Grid Computing* in the new age of the Internet has obviously attracted several authors to devise mechanisms to improve the infrastructure and intricacies that make it into a possible, effective, efficient and desirable way to perform the computational tasks. This efforts though have been traditionally focused almost in their entirety into technical aspects, either in software and hardware, and omitting or under-developing the transactional part of the Grid relationship between providers and consumers, regardless of its growing importance as more and more businesses interact in multiple party environment to achieve a single transaction with their customers.

This seminar paper aim is precisely to tackle the aforementioned under-explored area of Transactions in *Grid Computing* environments. The structure of the present document will be articulated around an initial revision of the key aspects of grid computing and the relevant transaction concepts that apply to it, a following section that will address the issues that surround the lack of a standardization on grid computing transactions and, finally a couple of sections directed towards the current proposals to avoid the casualties that could occur when performing the transactions and a concluding argumentation.

2.- Key concepts

In this second chapter of the paper, the author will review some of the notions and particularities that define *Grid Computing*, transactions and other aspects that are relevant to the current paper and how these concepts relate to each other, so they will be presented in a way that is rather oriented to their role in *Grid Computing* than as separate entities.

2.1.- Grid Computing

The term *Grid Computing*, as the reader certainly has imagined, comes from the analogy that one can establish between the emerging computing models and infrastructure to the traditional model of production and consumption of power in the electrical grid. As with the electricity that is produced in usually generally distant power plants with huge amounts of resources and specific hardware that enable a relationship with the customer in which the latter only has to plug his or her devices in the wall socket and seamlessly start consuming the goods offered by the electrical grid, the computing resources are also produces in *data centers* or *computing farms* in a much more efficient way, and are mainly delivered to the customer through the Internet or, as it is called under the new model, the Cloud.

The customer under the *Grid Computing* paradigm may or may not be the end user that one envisions when he adapts the model to the aforementioned analogy of the electrical grid that is plugging devices that range from vacuum cleaners to a Hi-Fi sets and ovens, but in the present paper, we will center the discussion on the business consumers that interact with the computing providers, which are likely also businesses, to satisfy their computing needs or to provide services to their own customers.

Grid Computing proposes a paradigm shift not only in infrastructure of production and delivery but also social-economic and in the market. The emergence of this new model, though, is not happening on the traditional time-scale of a few decades for the market and the government to adapt to a new infrastructure but in an internet-scale which speeds up the adoption and evolution of the market and the society by several orders of magnitude. Thus, it is crucial for enterprises, governments and all the other parties involved in the shift to *Grid Computing* to solve the key issues that will be hinted in the next few points and detailed in the next chapter.

2.2.- On-demand computing

On-demand computing, as opposed to the more traditional model where the users, whether they are businesses or human beings, buy a product following the retail procedure, consists on the availability of resources than can be bought an consumed for its instant use . This market paradigm was tailored to enable the service providers to meet the occasional peaks of the demand and adapt better to the eventual fluctuations of the demand achieving a higher level of dynamicity.

This dynamicity and flexibility is extremely invaluable in an environment that is constantly under steep trend changes and demand fluctuations such as the Internet is. Otherwise, companies would suffer the cost of not being available to generate resources at the rate needed to satisfy some periods of higher demand, or would be encumbered by the costs of huge data-centers and mainframes to be able to handle worst case situations , that would certainly be sitting idly an important part of the time.

Thanks to the emergence of *Grid Computing*, businesses are still presented with the same two situations as presented in the previous paragraph but with much more beneficial outcomes, id est, buy the resources in real time to computing providers at a sensible price that would virtually enlarge the capacity of the company to meet its demand, or make the extra computing resources generated by the investment in hardware to third-parties in an on-demand basis, respectively.

The data-centers of the companies, research entities or governments are starting and will certainly continue to shift towards a more ethereal or virtual nature as a result of the implantation of the on-demand model. No longer can one assume that the data-center of a certain company is in the company´s computing facilities, rather, one must assume it's virtual existence that may be scattered over several computing service providers.

The virtualization and scattering of the computer resources over the traditional borders of companies and/or service providers requires though the implantation of a new transactional model that allows a trust a and quality level sufficient for the enterprises to rely on it for the deployment of their business models.

2.3.- Peer-to-Peer

The popularity and influence of Peer-to-Peer or P2P systems has grown in the past few years until it has reached a point where this word has become part of the common knowledge of the general population, although sometimes is associated to lightly to other concepts that raise concerns like piracy and network clogging. In fact, it has become public knowledge that some ISP have been indiscriminately throttling this kind of communications to limit the impact of this widespread and successful technology.

Peer-to-Peer, though, is not merely used by file-sharing applications, but is a model which allows several nodes that are treating and computing data to communicate with each other without the additional step of having a supernode that supervises the major part of the operation of the others and on which the others rely and become effectively dependent.

Like *Grid Computing*, Peer-to-Peer[7] is also a method for organizing networks of nodes but the approach in which it bases its functionality is diametrically opposed to that of the former model. The former is oriented towards networks of moderate size that inter-operate with a moderate level of trustworthiness and sharing considerable amounts of resources per node such as it is done in supercomputing between different research groups. The latter centers itself around an environment with often ridiculously large amounts of nodes of varying nature and computational power that are far less reliable to operate until the end of the tasks than its Grid counterparts. To put an example in the same field, biological research, Folding@Home[3] takes a P2P approach and CSIC[4] has a Grid system for

3 Folding@Home is a distributed computing effort based on a P2P paradigm that harnesses the idle cycles of the CPU's and some well-equipped GPU's of donors to process data relevant to protein folding research. Further information at: http://folding.stanford.edu/

4 Centro Superior de Investigaciones Científicas (Superior Center of Scientific Research): It is the main government research entity of the kingdom of Spain.

several research groups that share resources to compute their different projects.

Peer-to-Peer has some advantages where *Grid Computing* is lacking and the other way around, so there have been some research on the likelihood of a convergence to some extent of their development[7] and even some hybrid approaches where the Grid transactions are handled in a Peer-to-Peer fashion[2]. The eventual convergence hypothesis is based upon the assumption (that time is increasingly making more likely to be correct) that Peer-to-Peer systems will leap into more complex use cases than the traditional file-sharing, thus raising the need for an increased control and a higher trust between peers; and that Grid systems will start scaling up in size and popularity (which in turn affects the consumers or users of the Grid, specially with the emergence of *Cloud Computing*) to levels that will require a more relaxed approach that avoids bottle-necking on the controlling nodes and other problems derived from the more central nature of Grids.

2.4.- e-Business

In the *Grid Computing* context, the term e-Business encompasses all the interaction between Companies that not only are developing their economic life on a mainly automated electronic level, but also have embraced the new paradigm of the Grid to deploy their business strategies and provide their services and or products to their customers.

The adoption of the Grid, though, means that the typical communication among companies in an e-Business transaction model that happens at the external (data) level is hardly applicable and certainly not efficient or adequate to the highly distributed and intensive nature of the computation in the Grid.

2.5.- Acid transactions

When considering if *Grid Computing* is suitable to perform adequately e-Business transactions, one must refer to its ability to represent or implement a model of transaction that is acceptable and sufficient to connect the processes of all the companies involved in a business use case that is designed to reside and compute in the Grid.

It is with this requirement in mind that one turns to the legacy models of transactions for reference and see how they compare and or are applicable to the current problem at hand. ACID transactions are the most popular type of transaction for databases and consists on the faithful following principles:

- Atomicity
- Consistency
- Isolation
- Durability

The strict ACID transactions though, have also some requirements[8] as for the nature of the transactions to which they shall be applied, and these requirements are:

- The transaction is short lived.
- Coordinator has entire control power over participants.
- application systems are tightly coupled.

It is beyond the scope to argue why the strict ACID transactions are not suitable for the mean e-Business transaction, but in regard to the e-Business transaction that would be able to support the nature and semantics of Grid e-Business transactions, they have some short-comings that make them invalid for straight application.

Grid Computing is generally used for long-lived computations such as scientific simulations and other data intensive calculations. Moreover, it organizes in a model that makes the coordination of a transactional entity considerably challenging as the reader will see in the third chapter of the current paper.

3.- The inadequacy of the traditional transaction model for *Grid Computing*

The problem that is under review in the current paper is none other than the need to find a transaction model that is suitable and sensible to be applied to the Business interactions that happen across the Grid.

In the last paragraph of the second chapter, it was mentioned that the coordination of transactions was the main challenge to overcome when trying to apply transactions to Grid computations. That statement is supported by the reality of Grid task control usually conforms to the following points[8] and [2]:

– Grid transactions are composed of service calls that provide a much higher level of abstraction than the one defined by database transactions and these services that compose a transaction can be handled by different peers in the Grid. This enables a higher degree of parallelism that is beneficial for computing needs but it demands a stronger concurrency control.

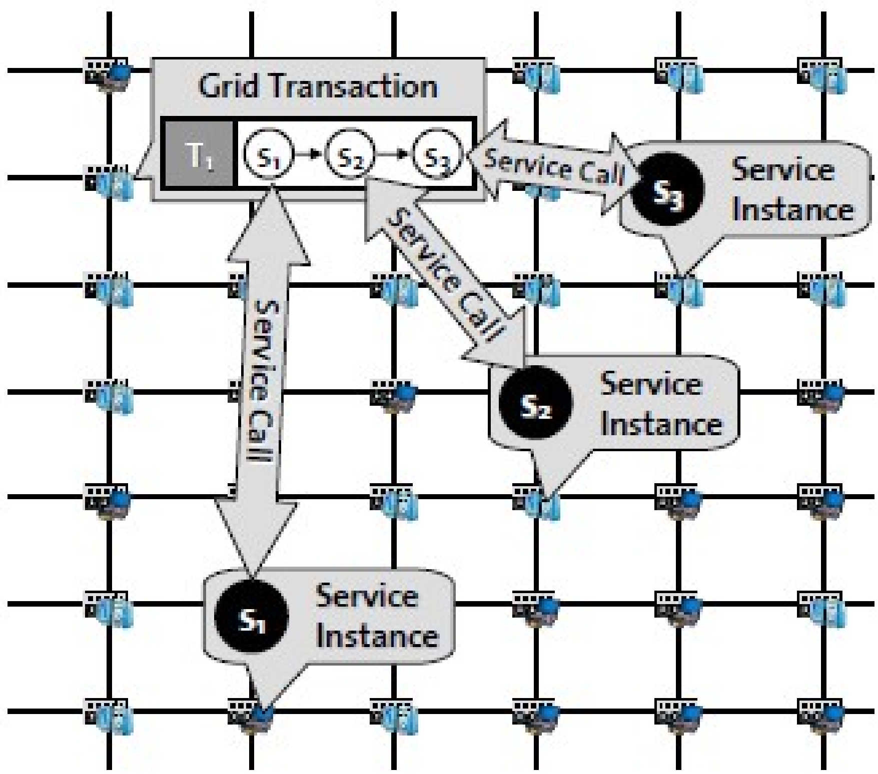

Figure 1: Depicts a transaction spread into service calls over the grid. From [2]

- Coordinating Grid services often takes a long time due to business latency or/and user interaction.
- Users may not be able to lock necessary resources because of the autonomy of Grid service.
- Communication is unreliable and transaction suffers from missing messages.
- Under the traditional atomicity model, operations within a transaction that failed would lead to roll-backs of many computational steps turning the recovery of failures in an almost unbearable procedure.
- Grid services are loosely coupled.

To address the reality of distributed transactions , there have been some approaches by several parties that standardize and model them but they are not aimed at solving the particularities of *Grid Computing* transactions, even though those are in essence a kind of distributed transactions.

Some of the traditional solutions to distributed computing are Distributed Transaction Processing (DTP), Object Transaction Service (OTS), WS-Coordination and WS-Transaction, BEA and HP-WST. All this solutions present some issues when they are to be applied on the specific distributed computing case of the Grid.

DTP is a widely used model to handle distributed transactions and it defines several roles and interfaces to achieve this purpose. OTS is a standard defined by the Object Management Group (OMG) to articulate distributed transactions on the Common Object Request Broker Architecture (CORBA). Both methodologies, though, come in as flawed when applied without any extension to the *Grid Computing* model for their inability or inadequacy to support long-lived transactions.

WS-Coordination and WS-Transaction conform a Web-services transaction framework proposed by three different parties, IBM, Microsoft and BEA. This framework is able to represent multiple coordination protocols and classifies transactions into atomic transactions and business activity, although the latter is not detailed extensively enough for it to be deployed in *Grid Computing* production systems.

The Organization for the Advancement of Structured Information Standards (OASIS) is the proponent of BTP and specifies a whole set of messages to be exchanged between the coordinator of the transaction and the other participants. BTP has received some extensions to reflect a model more Grid-Like such as HP's HP-WST, which is aimed at applying BTP to Web-services Transactions. A further adaptation of the latter to the Grid, though, would likely be too encumbering to the controlling nodes by the complex message structure and management workflow. Moreover, the fact that BTP does not provide a set of mechanisms for addressing recovery in a flexible way harm the chances of it becoming a solid base to *Grid Computing* Transactions.

As the reader can see, all these proposals cover to some extent the needs for Grid transactions but a Company willing to shift its strategy from a conventional model built around the use of its own resources, and eventually using the resources provided by very close partnership agreements, to a model more in consonance with the new trends in the closing years of the first decade of the XXI century, such as Cloud Computing built on top of the Grid, would hardly find the current state as a definite solution that provides a unique solution based upon a standardization and robustness solid enough to generate the amount of trust that such a strategy shift requires.

Trust, in fact, is probably the word that can define better the ingredient to move the private market, in mass, to a new technology that constitutes a change in the paradigm. It is obvious that there are always companies willing to assume the risks of early adoption when there are several key aspects to polish, refine and sometimes even define by intensive ad-hoc coupling and designing of the needed functionality that they require to operate. These early adopters are likely to bring innovation to the field, but are usually restricted to the biggest players who can afford to set considerable budgets to market creation and protocol creation efforts or small start-ups that rarely carry momentum enough to succeed and even less frequently to impulse the other parties in the market. To sum up the issues around transactions for *Grid Computing*, there must be a standard flexible enough for the market to regard *Grid Computing* as a reliable, cost-effective and trustworthy way to carry their internal operations and share their resources to maximize their benefit, that is what counts in the end.

4.- Proposed solutions to the transaction model for *Grid Computing*

In this chapter the author will present some of the proposals that have been made in the past few years to address the problem of transactions in *Grid Computing*. It is important to note, as pointed out in the end of the previous chapter, that companies have been operating with ad-hoc solutions for their business models on the Grid for a few years now, but the general feeling of the market is that for the model to really turn into an infrastructure as worthy and as implanted as the electrical grid, the road system and other social and market changing infrastructures, a set of solid and generally applicable protocols or frameworks must be reached.

4.1.- Peer-to-Peer transaction processing

As stated in the second chapter of this paper, Peer-to-Peer is a computational model that aims for an objective similar to that of the *Grid Computing* but taking a different approach in its base. This, however, as introduced by Foster and Iamnitchi in [7] can lead to an eventual convergence of the models. It is in this line of convergence, that Türker, Haller, Schuler and Schek propose a hybrid system where the computation itself that motivates the e-Business transaction between companies is handled by the Grid paradigm, but the transaction, in its control, is sustained by a Peer-to-Peer mechanism.

The model proposed by [7] makes some assumptions about the grid that should be noted:

– The Grid is composed by peers that are able to communicate not only with the usual control nodes that hand them the tasks, but with all the other nodes that integrate the Grid.

- Single peers provide a set of services that can be invoked using the peer's service interface. The services are executed as local database transactions.
- Services are not replicated on different peers, thus, conflicts cannot exist between services of different peers. (To eliminate this assumption and allow replication virtual peers are proposed).
- Conflicts to other services are set at the registration of the service.
- A grid transaction is a multilevel transaction that spans over several smaller service transactions. Thus, a grid transaction can be regarded as a compound service with transactional guarantees.
- Each peer provides a compensation service for each of its services which if the semantics allow it can be a bogus service.
- Each peer contains a log of the invocation of local services.
- Each transaction manages its own serialization graph with all the conflicts the transaction is directly involved in.
- Each peer provides a transaction execution environment that allows other peers to invoke its services in a P2P relationship.

With these assumptions stated, the model defines a strategy for providing semantic concurrency control, compensation and recovery for the grid transactions. As the name of the section suggests, instead of taking the easy route and creating a coordinating node that would assume the control over the grid transaction performed by the peers, which would hardly be optimal, although it is common, in big and complex enough grids due to the bottle-necking that would ensue the concentration on control on a specific peer, the authors propose a model in which every peer performs its own control and communicates with the others to ensure serializable executions, semantic atomicity and in general a reliable e-Business transaction.

Without a central coordinator, each peer must take upon itself the shared task that the whole transaction is carried out correctly and thus the peers are modeled as depicted in the figure 2.

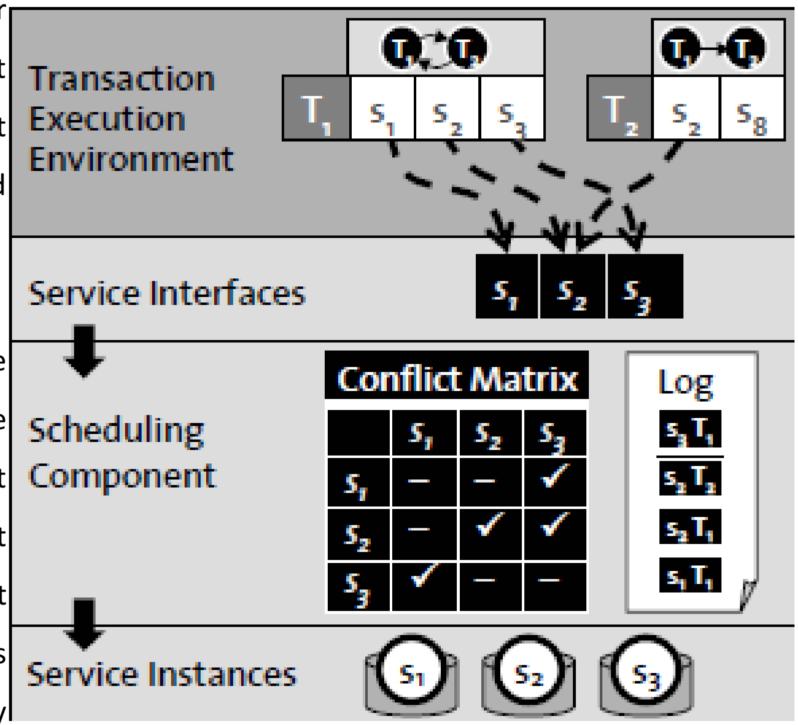

Figure 2: A grid peer on the P2P transactional model for Grid Computing. From [2]

To produce globally serializable executions, transactions will not be allowed to commit if they are dependent on an active transaction. To ensure that the peer knows on which transactions it depends, a direct conflict matrix is defined that will give the necessary information to apply the commit rule. To populate this information, each peer communicates directly with the other peers at service invocation time and fills in the local log and, at end, returns the the the conflict information together with the result of the service invocation.

The protocol bases its functionality around two parts. The first one is carried out at the peer level which consists on updating the local log at creation and returning results as described and spreading commit information among the dependent transactions for them to update their conflict status. The second part is handled at transactional level and consists on the execution phase, on which the transaction invokes services on peers according to its specification and waits for the peers to return them the results; the validation phase, where it checks if it is allowed to commit by checking the local serialization graph and waits if there is some active transaction on which it is dependent; and the commit phase, where it informs the peers on which it has invoked services of its commit, and those return the conflicts information that will be used to update the local serialization graph with the dependent transactions that have to be informed about the commit.

The proposal also proposes a method for distributed cycle detection that could make two or more transactions to wait for each other to commit if they are executed on different peers. To accomplish this it proposes a timeout solution and one that exchanges serialization graphs among transactions.

4.2.- GridTP

Under this name, Qi, Xie, Zhang and You propose in [1] a new architecture for Grid Transaction Processing based on the OGSA[5] platform and the X/Open DTP model. Unlike the previous architecture, GridTP does not constitute a major overhaul of the Grid architecture, and thus, takes a more evolutionary approach that maintains a similar programming model and it is independent of the transaction protocols such as those presented in the third chapter allowing it to be deployed in more conservative environments.

GridTP is built by merging OGSA supports[10](through [1]), via standard interfaces and conventions, the creation, termination, management, and invocation of stateful, transient services as named, managed entities with dynamic, managed lifetime; and the X/Open DTP[6] model that lacks solutions for naming , security and administration. Thus, coupling both architectures in a new protocol could provide a successful model for Transactions for *Grid Computing*.

5 Open Grid Services Architecture: An architecture for grid computing developed by the Global Grid Forum based around Web-services technologies such as WSDL and SOAP, but not completely tied to any of them.

6 For further information on X/Open DTP references [1] and [11]

GridTP is structured in three layers as depicted in Figure 3. The Common Resource Model layer consists of resources (every database is regarded as a resource) and the Transaction Manager (TM) which conducts the task of coordinating the transactions. Thus, grid databases are now treated as manageable resources.

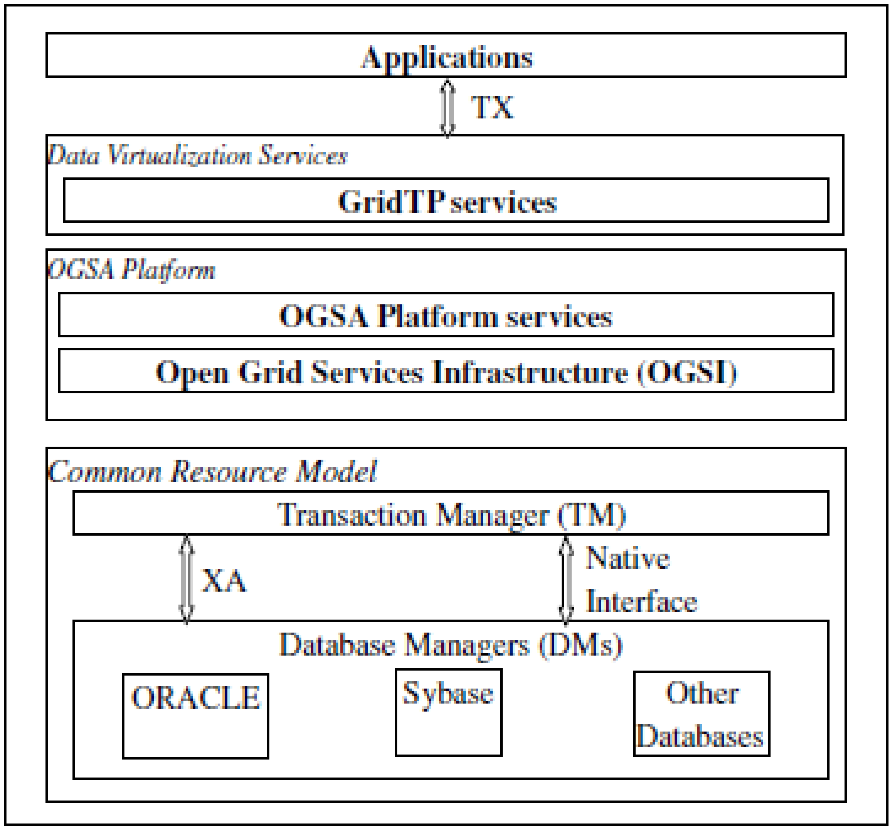

Figure 3: Architecture of GridTP services. From [1]

In the OGSA platform everything is regarded as a Grid service, and thanks to the interfacing OGSI and the transaction manager, databases can also be represented as Grid services, allowing them to support "stateful service supporting reliable and secure invocation (when required), lifetime management, notification, policy management, credential management, and virtualization."[1].

In the Data virtualization Layer is where the GridTP services are constructed based on the underlying OGSA platform. Once the GridTP services are produced, they are made available through transactions (TX in Figure 3) under the WSDL standard.

The handling of transactions, then, is performed at two different levels. The first level is global transaction control by the means of the TX interface between the application that uses the Grid as its execution environment and the GridTP. The second level is structured by local and more traditional transactions between GridTP and the Transaction Manager and through the latter to the Database Managers that implement database transactions.

4.3.- Trust Management

As mentioned in the beginning of the last paragraph of the third chapter, trust is vital for a mass adoption of the *Grid Computing* model by the business world. It is in this precise direction that Azzedin and Maheswaran have researched and published a paper that proposes a solution that builds the trust upon identity and behavior. Behavior trust is, in fact, an aspect that has been considerably overlooked when tackling the problem of trust but that can be of vital importance for e-Businesses to determine which providers are more reliable in a dynamic way allowing them to automatically move their virtual data-centers towards more trustworthy spaces.

To generate the necessary amount of trust that overcomes the natural reticence to share resources with external entities that can be located literally on the other side of the globe, first one must define the trust that is entitled to the other party in the relationship on the event of a successful identification[7].

The identification process, however, does not provide the means to tell if the software and or hardware upon which the resources are provided or consumed is trustworthy. To achieve this purpose, Azzedin and Maheswaran propose a behavioral trust that is composed by weighing the

Trust Level (TL)	Description
A	very low trust level
B	low trust level
C	medium trust level
D	high trust level
E	very high trust level
F	extremely high trust level

Table 1: Depiction of the qualitative values that can be assigned to entities.

7 The identification process is usually based on common technologies such as encryption, data hiding, digital signatures, authentication protocols, etc. From [9]

trustworthiness of the entity, that is calculated by direct interaction with it, and the reputation of the parties that they define as "*The reputation of an entity is an expectation of its behavior based on other entities' observations or information about the entity's past behavior within a specific context at a given time*.". Under this definition, it is necessary to implement some algorithms that give an indication of the Trust Level (TL) that can range from *very trustworthy* to *very untrustworthy.*

This interaction with an entity, whether it is direct or to other parties that are trustworthy to us, is based upon the updating of trust on transaction completion, more precisely, on the outcome of the transaction that is established between e-Businesses.

Trust in a *Grid Computing* e-Business environment must reflect the original properties of the concept in the Business world, and thus the proposed model makes trust a decaying value, id est, it is lowered automatically as the time advances from the last interaction. Furthermore, the parties of the e-Business transaction may form alliances and partnerships that should affect the amount of trust that is perceived from them.

Calculating all the trust and reputation values at a general level could impact the performance of the Grid by an obvious bottleneck on the trust controller, but [9] proposes a model in which the Grid is split into several *Grid Domains, where the* computation is handled locally making the whole approach far more distributed. This sharing of the trust computation farther improves the trust calculations that are not updated soon but typically are performed on large amounts of data.

Some refinements are also proposed such as inheritance of trust where the model provides a mechanism to reflect the fact that if a new entity joins a domain, whether it be, the client or the provider one, the trust associated to the domain is inherited albeit not with the same weight as the elder members of the domain. Another improvement is the protection against newcomers that consists of an enforced higher security interaction when the TL is starting to be generated and, consequently, does not give the other parties much confidence of its reliability or even its honesty.

5.- Conclusions

Transactions for *Grid Computing* are a commonly overlooked as field of study. This is probably because more resources have been allocated to the core functionality of the Grid itself as a computational model, id est, aspects such like fair time sharing, task control, topology, etc and transactional aspects have been addressed in an ad-hoc way. With the advent of the new computing that will be brought upon us by a more than likely shift in the computational model from retail and standalone user powered applications to the Cloud, though, the relevance of e-Business transactions in Grids will become increasingly relevant and, as such, it is important to start funding research like in the lines of those presented in the paper that propose models that could become more or less standard to apply to the most common cases.

The first two solutions presented in the previous chapter tackle basically the same problem, which is the nonexistence of a transactional model that is suitable for computations carried on a Grid. While the Peer-to-Peer transaction processing aims for an ambitious merger of two emergent technologies (Grid and P2P) that started from very different approaches to a similar problem to construct a solution that follows a philosophy that is almost optimally in consonance with the long-running and highly-distributed spirit of *Grid Computing*; *GridTP* aims for a more incremental construction on a solution based on integrating existent and proofed technologies that are complementary to some extent and adding a layer on top to provide the amount of functionality, security and reliability necessary for e-Business transactions on the Grid.

The third solution, provides a very interesting perspective to transactions on *Grid Computing* by making them represent a key concept that is often taken for granted in transactions such as trust. This focus is much more relevant to *Grid Computing* due to the virtual and shared nature of the resources that are and will be consumed by the companies that collaborate in this new model. The amount of fidelity with which trust is modeled into the transaction is worth of mentioning, as it can be used to automatize the choice of clients and providers with an accuracy similar to that of a traditional company that shifts customer strategy depending on the results of previous interactions with them.

To sum up, the opinion of the author is that ideally the transactions for grid computing should grow in the distributed and highly independent, although respecting the transaction dependency semantics, direction proposed by Peer-to-Peer transaction processing, but adding the advantages of the trust management proposal. The second solution, seems also adequate to affront the needs of grid transactions, and indeed could be likely developed over a few iterations to constitute a successful aggregation of fire-proofed technologies that could stack quite seamlessly, but in the opinion of the author the approach is lacking a higher amount of compromise with specific underlying solutions that could simplify it's representation and costs of development.

6.- References

[1] Qi, Z. and X. Xie, B. Zhang and J.You,(2004). *Integrating x/open dtp into grid services for grid transaction processing*. In FTDCS, pages 128-134. IEEE Computer Society.

[2] Türker, C. and K.Haller, C. Schuler, and H.J. Schek, (2005). *How can we support grid transactions? Towards peer-to-peer transaction processing*. In CIDR, pages 174-185.

[3] A. Weiss, (Dec 2007). Computing in the Clouds. In *netWorker*,11(4) pages 16-25.

[4] R. Buyya, C. S. Yeo, and S. Venugopal, (HPCC 2008), *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*. 10th IEEE International Conference on High Performance Computing and Communications, Sept. 25-27, 2008, Dalian, China.

[5] I. Foster, and C. Kesselman (eds), (1999). *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, San Francisco, USA.

[6] *Twenty-One Experts Define Cloud Computing*, (January 24th, 2009) http://cloudcomputing.sys-con.com/node/612375

[7] I. Foster, and A. Iamnitchi, (2003).*On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing*. Peer-to-Peer Systems II, pages 118-128.

[8] F. Tang, and M. Li, and J. Cao, and Q. Deng, (December 2003). *Coordinating Business Transaction for Grid Service*. In *Proceedings of the Second International Workshop on Grid and Cooperative Computing*, Lecture Notes in Computer Science 3032, Shanghai, pages 108–114.

[9] F. Azzedin, and M. Maheswaran, (May 2002). *Evolving and Managing Trust in Grid Computing Systems*. In IEEE Canadian Conference on Electrical & Computer Engineering (CCECE '02).

[10] I. Foster, and C. Kesselman, and J.M. Nick, et al, (January 2002). *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. http://www.globus.org/ogsa

[11] X/Open Company Ltd , (February 1996). *X/Open CAE Specification. X/Open Distributed Transaction Processing: Reference Model* (ISBN: 1-85912-170-5, G504).