Agent based Local Ad Hoc Grids

Tariq Abdullah, Koen Bertels

***Abstract*—Every organization owns or manages a LAN for its day-to-day businesses. Different studies shows that on the average only 5% of the total computational power of an organization is used and remaining 95% goes unused. On the other side there are enormous research projects that can't be performed because the research community or a single organization can't afford to purchase the required computational resources. Local ad hoc grids are formed with the aim to utilize the unutilized computational resources of a single organization or across different organizations. Participating computing nodes in local ad hoc grids need to be autonomous, intelligent, reactive and self-organizing. *Software agents* provide these characteristics. In this paper we describe *local Ad Hoc grid*s as an alternative to large scale grids and supercomputing. Different issues faced in developing an Agent-based local ad hoc grid will be discussed. At the end initial design and experimental setup of an agent-based local ad hoc grid will be discussed.

***Index Terms*—Ad Hoc Grids, Self Organization, Software Agents, Ad Hoc Grid Challenges**

The term grid computing was coined in mid 1990s to denote a new infrastructure of distributed computing for scientists and engineers in a more advanced scope. The name was inspired by the electrical power energy because of its pervasiveness, ease of use and reliability[6].

I. Why do we need Local Ad Hoc Grid?

The adoption of the grid by the commercial sector can be compared to the initial proliferation of the Internet in mid 1980s. Unlike single global Internet there exist a number of overlapping grid implementations supporting different requirements and scale. Existing grid implementations can be categorized into national production grids, community production grids, enterprise production grids, and volunteer production grids[6]based on the organizational boundaries. *National production grids* aggregate high-end computing, data and network resources across a nation to provide a unified distributed computing infrastructure like German D-Grid initiative announcement[10], CNGrid[9], NAREGI[16]). Membership, collaboration and access control is centralized and is regulated at national level. Application are also of national level importance. *Community Production Grids* are structurally similar to the national production grids. In these grids, aggregating resources are pools of resources across multiple geographic and administrative domains. These grids are formed to achieve a mutually beneficial scientific or commercial goal of interest to the community like GriPhyN[11], PPDG[19],SDSS, iVDGL[12], NEES[7].Membership in community grid is normally controlled by a some central administrative authority. *Enterprise production grids* comprises of the resources that are the part of the organization constituting the enterprise like Entropia[1]. These grids may include low-end computational resource that are the part of that enterprise. Access to the enterprise grid is often restricted to the members of that enterprise. *Volunteer Production Grids* allow Internet user to volunteer their unused computational resources, normally for a non-profit scientific task[8]. In volunteer production grids, membership is based on some trust model. We observe some common traits in above mentioned grid architectures after a careful analysis of these architectures.

- Support to mutually collaborative communities
  Irrespective of the scale and orientations, the participants of these grid architectures share a synchronized and non-conflicting objective.
- A centralized architecture and a regulated control for membership and access privileges
  These grid architectures have a dedicated administrative authority for policy enforcement, monitoring and access privileges)
- assumption of a stable well-defined collaboration
  These collaborations are based on some pre-defined usage polices rules, and access privileges.
- Mostly fixed goals, use policies, and membership rules during the life time of the grid
  It is interesting to mention that the scale of efforts involved in these grids, it is very rare that the goals, usage policies or membership rules will change during the life time of these grids.

There are another classes of applications which are resource intensive but, at the same time, very difficult to execute on the present day grid infrastructures, if not impossible at all. One such class of application is the situation where participants are offering different resources to collaborate on a common objective. A team of scientists may provide data analysis software, another team of scientists may pool visualization service and third group may provide data storage repository for the input of the analysis software. In this example every participant want to participate with his/her own usage policies and access rights to its resources for a certain limited amount of time, normally till the participant has some utility interest in the participation. Administrative overheads erupting from this type of experiments make it impractical for such transient communities to undergo a formal grid establishment process, probably one time collaboration in most of the cases.

t.abdullah@ewi.tudelft.nl, Computer Engineering Laboratory, Electrical Engineering Faculty, Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands, Phone: +31 15 27 81832, Fax: +31 15 27 84898

k.l.m.bertels@tudelft.nl, Computer Engineering Laboratory, Electrical Engineering Faculty, Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands, Phone: +31 15 27 81632, Fax: +31 15 27 84898

Another example can a grid market where grid resources are treated as commodity. Individuals or organizations participate in these type of grid market for trading their resources with potential buyers/consumers. In this scenario participants want to optimize their respective objective function (utility). *Every participant participate in the grid market with its own objective function, pricing rules and usage policy.* This type of grid market can't be implemented and monitored by single controlling authority. Moreover grid market have metamorphic structures and self-organize themselves over a period of time. Conventional grid infrastructures fail to support grid market as they rely on some predefined network and structure dependent services.

These classes of present day applications necessitates a new type of GRID that doesn't require centralized control authority, no predefined underlying network infrastructure, no predefined usage rules and access policies, no fixed resource discovery mechanism and no initial commitments for participation in grid from participants. *Current GRID infrastructures fail to accommodate the above mentioned classes of applications.* However we find the another, yet new, type notion of grid called *PC grid or Desktop grids or Local Ad Hoc grids.* Thes grids are suitable for the following categories of users.

- Application owners who need to improve performance and speed of their work without comprising on quality
- Scientists who need to deliver increased HPC capacity within budget and without increasing the infrastructure complexity.
- Organization who want to have maximum value of their hardware and network infrastructure.

II. What are Local Ad Hoc Grids?

Spontaneously formed grids to harness the unused resources of idle workstations inter/intra-organizational infrastructures to provide high performance computing nodes on demand can be called as an ad hoc grid. Some definitions from literature are as follows:

- "A distributed computing architecture offering structure-, technology-, and control- independent grid solution that supports sporadic and ad hoc use modalities of the gird"[3].
- "The opportunity to perform large computations at low-cost by using volatile desktop resources"[15].
- "Desktop grid computing is a form of distributed computing in which an organization (such as a business) uses its existing desktop PCs to handle its own long-running computational tasks ".

Increased complexity and heterogeneity, dynamic infrastructure and autonomous behaviour of participating nodes in local ad hoc grid requires it to be self managing. Software agents with autonomy, mobility, learning, reactivity, social-ability, intelligence and proactiveness are the promising candidates for local ad hoc grids. Every agent has a *unique identity* within a well-defined boundary and interfaces. Agent has a particular *design objectives* to achieve that can be represented implicitly or explicitly. Agent is *autonomous* when it has control over its internal state and on its behaviour. Agent needs to be *reactive* by timely responding to changes that occur in its environment in order to satisfy its design objectives. Agent also needs to adopt new goals and take initiative in order to satisfy its design objectives by being *proactive*. Agent can be *mobile*. Agent exhibit weak mobility when it can only migrate its data from one environment to some other environment and exhibit strong mobility when it can migrate process as well as data. Agent can exhibit *social-ability* by communicating with other agents. Agent can *learn* knowledge from its environment as well as from its past experiences. When agent applies its knowledge according to circumstances in order to fulfill its design objectives then it is called *intelligent agent*. Agent needs to be *interoperable* so that it can communicate and exist in different operating environments[20]. Due to the these characteristics of software agents can be used in developing Agent based Local Ad Hoc grids.

III. Challenges in the Development of ad hoc grids

- To deal with the increased complexity of the ad hoc grid due to absence of centralized control and extremely large number of participation.
- To consider the increased heterogeneity of participating nodes in hardware, software and network.
- To create a robust, reliable resource from above mentioned unreliable heterogeneous nodes.
- To cope up with the dynamic topology changes of the underlying network.
- To create a secure resource from un-secure participating nodes.
- Autonomous behavior of participating nodes to form more heterogeneous ad hoc grids.
- To match application requirements to the available resources in an effective manner

IV. Survey of Ad Hoc Grids

1) BOINC (Berkeley Open Infrastructure for Network Computing)
   BOINC [4] is a non-commercial open source middleware, under LGPL, originally developed to support SETI@home project. Its purpose is to make it possible to tap the enormous processing power of the computer around the globe. It is supported by Linux, Unix, different version of Windows, and Mac OS. Its structure consist of server and client layers. Server component can run on one or more Unix machines to to allow easy scalability of the projects of any size. Server components send jobs to client components. These client software communicate with each other to distribute, processes and return work units. It has over 430,000 hosts and provides 633 TFLOPS as of Sep-2007[18]. A number of different projects from Physics (Einstein@Home), Astronomy (LHC@home, SETI@home, Astroplus), Mathematics (ABC@Home, SZTAKI Desktop Grid), Earth Sciences (Climateprediction.net) and Biology (Cell computing, Malaria Control,

Name		Objective	Participants	CPU cycles	Type	Comments
1	BOINC	substrate for global computing projects	430,000	633 TFLOPS	Research	centralized scheduler for participants
2	XtremWeb	same as BOINC			Research	centralized set-up of servers and workers
3	Entropia	use unused cycles of underlying network			Commercial	Clusters and mainframes can be connected into the Desktop grid
4	PC GRID	Commercial solution	Same as Entropia		Commercial	Centralized control model
5	DeskGrid	Same as Entropia	Same as Entropia		Commercial	read the website for details
6	SZTAKI	Promote PRC model in Hungary	25982	1.5 TFLOPS (peak value)	Research	Based on BOINC
7	EnFusion	based on economic concepts				commercial variant of Nimrod-G

Table I
AD HOC GRID PROJECTS

Predictor@home, proteins@home, SIMAP, TANPAKU, World Community Grid) are using BOINC.

2) DeskGrid
It is a distributed processing framework to use idle time of the Enterprise desktop PC's. DeskGrid is ideal for problems that require extensive computing resources and can be broken into pieces. It includes a generic submitter and wrapper program for submitting user jobs to the grid. A client is activated by its screen saver.

3) DG-ADAJ (Desktop Grid-Adaptive Distributed Application in Java), CC-ADAJ
It[2] is a middleware platform, facilitating the SSI (Single System Image), to enable the execution of heterogeneous applications with irregular and unpredictable execution control in Desktop Grids. DG-ADAJ is designed and implemented above the JavaParty (allows easy port of multi-threaded Java programs to distributed environments) and Java RMI platform. The DG-ADAJ may consist of many desktop grid centers that can work in parallel under the control of desktop grid host server. It provides a distributed execution platform with an observation and automatic load balancing mechanisms. The observation mechanism helps to predict the communication tendencies between these objects during run time[15].

4) Entropia
A[1] desktop grid comprising of heterogeneous computing resources, heterogeneous OS, heterogeneous network environments, complex security/trust relationship and large numbers of resources. Its purpose is to utilize unused cycle of Internet based or enterprise based underlying network. It includes extensive support for central management, failure management and robust execution. This project was founded in 1997 to support mathematical research. Its resource management techniques are targeted to make the systems manageable, usable and highly productive. These techniques exploit database, Internet/network and high performance computing technologies on a scale of hundreds of thousands of computers.

5) PC GRID
It is a commercial solution for ad hoc grid from United Devices (UD). Its implementations vary from small departmental solutions to global enterprise implementations. It is based on UD's Grid MP technology. Its main features are unparalleled scalability, security, data analysis for infrastructure visibility, ease of use, ease of management and a robust framework for enabling the application[13]. Job scheduler, resource manager and grid agent are its main components. Scheduler optimizes scheduling based on best-fit resource availability. Resource manager manages application and data. Whereas grid agents advertise availability and capacity, execute work and return results. All these components work unobtrusively without effecting the work of end users. It provides faster application processing and job throughput and increased ROI on existing hardware. It also in improved planning through capacity, utilization and job analytics.

6) SZTAKI
It[14] is a desktop grid utilizing the Public Resource Computing (PRC) concept in Hungary. It uses BOINC for executing computational task and for storing data sets. It supports Linux, Windows and Mac OS. Client nodes need only to download BOINC software to be a part of SZTAKI grid. User can control when, how much and how long his/her machine will take part into the desktop grid. Jobs are assigned to the clients according to the resources volunteered by the client itself. As of October 1, 2007, there were 19776 users with 25982 hosts. Its peak performance has been 1.5TFlops/s[17]. It is being used in different projects like drug discovery, data mining, machine learning and information theory.

7) XtremeWeb
It[5] is a free, open source software platform to explore the scientific issues and applications of desktop grid, global computing and P2P distributed systems. It can use idle CPU time of pools of resources connected through Internet or through LAN. It is composed of client, server and worker components. Worker contact the server to get the job. Server sends the job and data to the worker. When worker completes the job it sends the results to the Result Controller. It can also be used to build centralized Peer-to-Peer systems with centralized control such as audio file exchange projects.

Table-I represents some Desktop/PC/Ad Hoc Grid research projects or its implementations.

V. Conclusions and Future Work

In this paper we described the scenarios where conventional grids are not suitable and local ad hoc grids are required. Challenges faced in development and deployment of local ad hoc grids were presented. Why and how software agents are useful for Ad Hoc grids was discussed later in the paper. An overview of the existing ad hoc grid projects and ad hoc grid implementations was given. These include BOINC, DeskGrid, DG-ADJ, Entropia, PC-Grid, SZTAKI and XtremeWeb. In future we plan to develop middleware required for the implementation of Agent based local Ad Hoc grids in GRAPPA project. This implementation will attempt to tackle the challenges faced in the development of local Ad Hoc grids with particular focus on the run-time configuration and use of reconfigurable hardware for the specialized applications.

References

[1] Chien A., B. Calder, S. Elbert, and K. Bhatia. Entropia: Architecture and performance of an enterprise desktop grid system. *Journal of Parallel Distributed Computing*, 63(5):597–610, May 2003.

[2] Iyad Alshabani, Richard Olejnik, Bernard Toursel, Marek Tudruj, and Eryk Laskowski. A framework for desktop grid applications: Ccadaj. In *ISPDC '06: Proceedings of the Proceedings of The Fifth International Symposium on Parallel and Distributed Computing*, pages 208–214, Washington, DC, USA, 2006. IEEE Computer Society.

[3] Kaizar Amin, Gregor von Laszewski, and Armin R. Mikler. Toward an architecture for ad hoc grids. In *Proceedings of the 12th International Conference on Advanced Computing and Communications,ADCOM 2004*, Ahmedabad Gujarat, India, December 15-18 2004.

[4] David P. Anderson. Boinc: A system for public-resource computing and storage. In *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, pages 4– 10, Washington, DC, USA, November 2004. IEEE Computer Society.

[5] Franck Cappello, Samir Djilali, Gilles Fedak, Thomas Herault, Frédéric Magniette, Vincent Néri, and Oleg Lodygensky. Computing on large-scale distributed systems: Xtrem web architecture, programming models, security, tests and convergence with grid. *Future Gener. Comput. Syst.*, 21(3):417–437, 2005.

[6] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[7] http://access.ncsa.uiuc.edu/Stories/NEESgrid. Neesgrid home page, last visited on may-2007.

[8] http://setiathome.berkeley.edu/. Seti@home project home page, last visited on may-2007.

[9] http://www.csis.hku.hk/ clwang/grid/CNGrid.html. China national grid (cngrid), last visited in may-2007.

[10] http://www.d grid.de/. German national grid initiative, last visited in may-2007.

[11] http://www.griphyn.org/. Gridphyn home page, last visited on may-2007.

[12] http://www.ivdgl.org/. Ivdgl project home page, last visited on may-2007.

[13] http://www.ud.com. Home page of united devices for pc grid.

[14] Petter Kacsuk, Norbert Podhorszki, and Tamas Kiss. Scalable desktop grid system. In *VECPAR 2006: 7th International meeting on High Performance Computing for Computational Science Rio de Janeiro*, pages 1–13, 2006.

[15] Derrick Kondo, Michela Taufer, Charles L. Brooks, Henri Casanova, and Andrew A. Chien. Characterizing and evaluating desktop grids: an empirical study. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, April 2004.

[16] S. Matsuoka, S. Shimojo, M. Aoyagi, S. Sekiguchi, H. Usami, and K. Miura. Japanese computational grid research project: Naregi. In *Proceedings of the IEEE*, volume 93(3), pages 522–533, march 2005.

[17] SZTAKI Grid Home page. http://www.lpds.sztaki.hu/desktopgrid/.

[18] BOINC Statistics. http://boincstats.com/.

[19] PPDG team. The particle physics data grid (ppdg): From fabric to physics, final report, July 2006.

[20] M. Wooldridge. Agent based software engineering. *IEE Proceedings on Software Engineering*, 144(1):26–37, February 1997.