

Grid RPC meets Data Grid: Network Enabled Services for Data Farming on the Grid

Satoshi Matsuoka
Tokyo Institute of Technology
Tokyo, Japan
matsu@is.titech.ac.jp

The Computational Grid[1] is a promising platform for running large-scale scientific applications. It provides a base software infrastructure that allows for the development of *middleware* aimed at deploying applications on Grid resources. The question is, how do you program it---in this regard, Network-Enabled Server (NES) paradigm, which enables Grid-based RPC, or *GridRPC* for short is a good candidate as a viable Grid middleware that offers a simple yet powerful programming paradigm for programming on the Grid. Several systems that facilitate whole or parts of the paradigm are already in existence, such as Neos[7], Netsolve[3], Nimrod/G[4], Ninf[2], and RCS[6], and we feel that pursuit of a common design in GridRPC, as had been done for MPI for message passing, will bring benefits of standardized programming model to the Grid world. This talk will introduce the NES/Grid RPC features, discuss early user experiences, and touch upon the Grid Data Farm project, based on Grid RPC, which involving processing Petabytes of collider accelerator data streaming over the Euro-Japanese link with thousands-node scale cluster possibly spread over several Japanese institutions.

Compared to traditional RPC systems, such as CORBA, designed for applications that facilitate non-scientific applications, GridRPC systems offer features and capabilities that make it easy to program medium- to coarse-grained, task parallel applications that involve hundreds to thousands or more high-performance nodes, either concentrated as a tightly coupled cluster, or a set of them spread over a wide-area network. Such applications will often require handling of shipping megabytes of multi-dimensional array data in a user-transparent and efficient way, as well as requiring the support of RPC calls that range anywhere from 100s of milliseconds up to several days or even weeks. There are other necessary features of Grid RPC systems such as dynamic resource discovery, dynamic load balancing, fault tolerance, security (multi-site authentication, delegation of authentication, adapting to multiple security policies, etc.), easy-to-use client/server management, firewall and private address considerations, remote large file and I/O support etc. These features are essentially what is needed for the Grid RPC systems to execute well on the Grid---features either

missing or incomplete in traditional 'closed world' RPC systems---and in fact are what are provided by lower level Grid substrates such as Condor[10], Globus[8], and Legion[9]. As such GridRPC systems either provide these features themselves, or builds upon the features provided by such substrates.

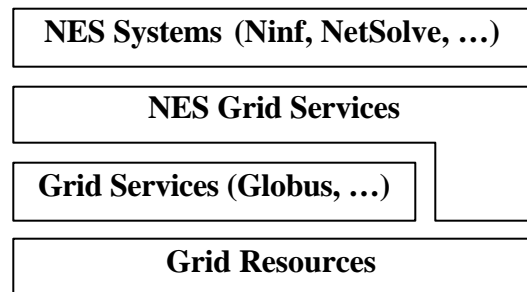


Figure 1. Software hierarchy

In a sense, NES/GridRPC systems abstract away much of the Grid infrastructure and the associated complexities, allowing the users to program in a style he is accustomed to in order to exploit task-parallelism, i.e., asynchronous parallel procedure invocation where arguments and return values are passed by value or reference depending on his preference. Our studies as well as user experiences have shown that this paradigm is amenable to many large-scale applications and especially to scientific simulations. The difference here is that 1) because of the 'open world' Grid assumptions the underlying GridRPC system must be much more robust, and 2) the scalability of the applications, in terms of the execution time, the parallelism, and the amount of data involved, must scale from just one node with a simple LAN RPC to thousand-node task parallel execution involving weeks and Terabytes even to Petabytes of data. In the talk I will aim to demonstrate such benefits, both from the test cases as well as experiences from various application studies.

One challenging application and middleware project we are tackling now with NES/Grid RPC is the *Grid Data Farm* project. The project is collaboration among KEK (High-Energy Physics Lab), ETL/TACC (Electrotechnical

Laboratory / Tsukuba Advanced Computing Center), the University of Tokyo, and Tokyo Institute of Technology.

- Multi-Tier data sharing and efficient access
- Program sharing and management

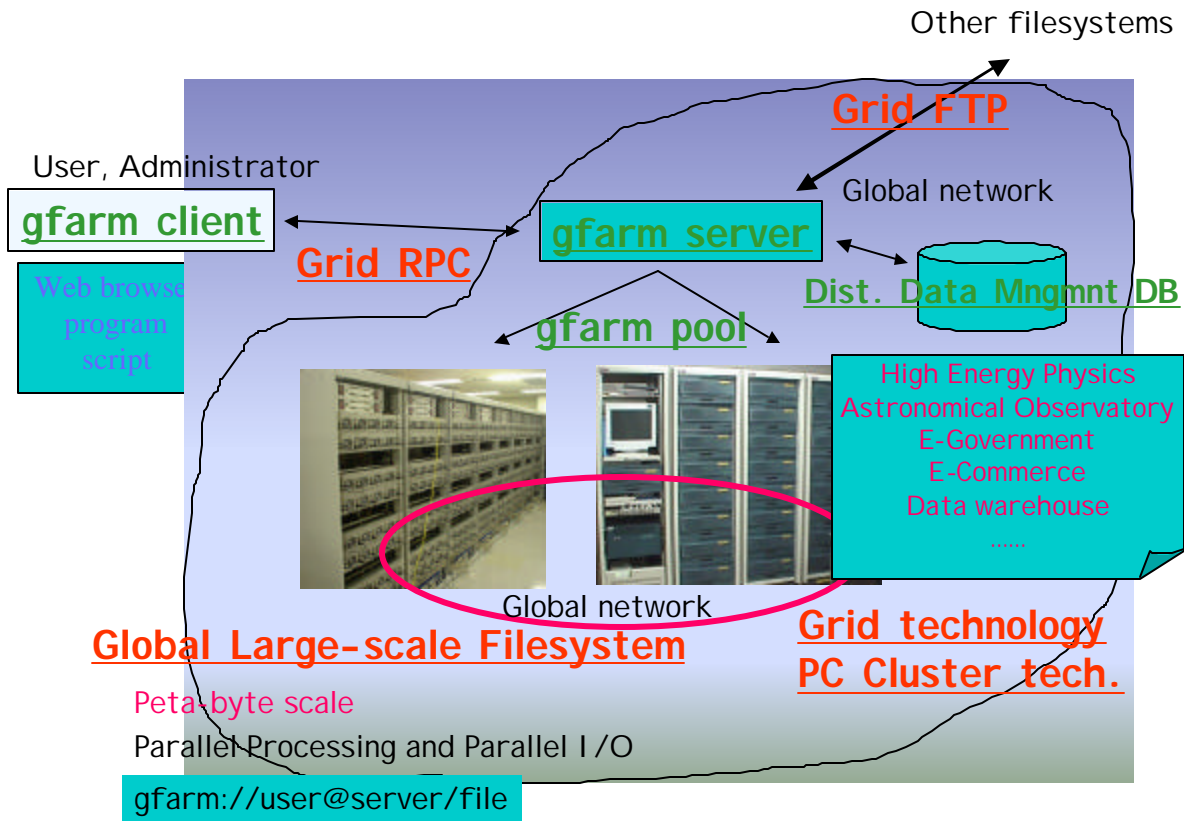


Figure.2: Overview of the Grid DataFarm

The challenge is a so-called 'DataGrid' challenge, and will involve construction of a data processing framework that will handle 100Terabyte to Petabyte scale data emanated by the ATLAS sensor of the Large Hedron Collider Accelerator that will be built at CERN by 2005. The underling hardware will be a thousands node scale PC cluster, each node facilitating a near-Terabyte of storage, and incoming data of approximately continuous 100Mbps bandwidth from CERN will be systematically stored and will be subject to intensive processing. The Grid Data Farm itself is a middleware that will be built with GridRPC as the major component, and will facilitate the following features for collider data processing as well as serving as a framework for other types of data-intensive scientific applications:

- World-wide group-oriented authentication and access control
- Thousands-node, wide-area resource management and scheduling

- System monitoring and administration
- Fault tolerance / dynamic re-configuration/ Automated data regeneration or re-computation
- Global file system for Petabyte scale data
- Parallel I/O and parallel processing for fast file service

Finally, for providing bulk of computational resources not just for the Grid DataFarm but as general compute resources of NES/GridRPC systems as well as wide-area usage of high-performance systems, our group is vigorously pursuing the construction of COTS clusters, intended to be used as a Grid nodes for future Grid infrastructures including the ApGrid (Asia-Pacific Grid, <http://www.apgrid.org>). Our lab currently has 6 clusters for variety of software development as well as simulation runs, ranging from the 256 processor large-scale/high-density/'very commodity' Presto II cluster, several 100-node scale clusters, to highly reconfigurable 'Plug&Play' cluster. Currently, our lab alone features over 400 processors with aggregated peak of 500 GigaFlops, while

our collaborator at ETL/TACC has over a TeraFlop of cluster computing power. Details of our Grid and Clusters research can be found at sites <http://ninf.etl.go.jp>, and <http://matsu-www.is.titech.ac.jp>.



Figure 3: 256-processor high-density Presto II cluster (left)

References

- [1] Foster, I.; Kesselman, C. 1998. *The Grid, Blueprint for a New computing Infrastructure*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA.
- [2] Sekiguchi, S.; Sato, M.; Nakada, H.; Matsuoka, S.; Nagashima, U. 1996. "Ninf: Network-Based Information Library for Globally High Performance Computing." In *Proceedings of Parallel Object-Oriented Methods and Applications* (POOMA, Santa Fe, February). 39-48.
- [3] Casanova, H. and Dongarra, J. 1997. "NetSolve: A Network Server for Solving Computational Science Problems". *The International Journal of Supercomputer Applications and High Performance Computing*, Vol. 11, No. 3, 212-223. Also in *Proceedings of Supercomputing 1996*.
- [4] Abramson, D., Giddy, J., Kotler, L. 2000. "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?" In *Proceeding of the International Parallel and Distributed Processing Symposium* (IPDPS, Cancun, Mexico). 520-528.
- [5] Kapadia, N.H.; Forter, J.A.B.; Brodley, C.E. 1999. "Predictive Application-Performance Modeling in a Computational Grid Environment." In *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing* (HPDC8, Redondo Beach, August).
- [6] Arbenz, P.; Gander, W.; Oettli, M. 1997. "The Remote Computational System." *Parallel Computing*, Vol. 23, No. 10, 1421-1428
- [7] Czyzyk, J.; Mesnier, M.; More, J. 1996. "NEOS: The Network-Enabled Optimization System." Technical Report MCS-P615-1096. Mathematics and Computer Science Division, Argonne National Laboratory, IL, USA.
- [8] Foster, I.; Kesselman, K. 1997. "Globus: A Metacomputing Infrastructure Toolkit." *International Journal of Supercomputer Applications*, Vol. 11, No. 2, 115-128.
- [9] Grimshaw, A.; Ferrari, A.; Knabe, F.C.; Humphrey, M. 1999. "Wide-Area Computing: Resource Sharing on a Large Scale." *IEEE Computer*, Vol. 32, No. 5, 29-37.
- [10] Litzkow, M.; Livny, M.; Mutka, M. 1988. "Condor - A Hunter of Idle Workstations." In *Proceedings of the 8th International Conference of Distributed Computing Systems*. 104-111.

Satoshi Matsuoka received his B.C., M.S., and Ph. D. from the Information Science Department of the University of Tokyo in 1986, 1988, and 1993, respectively. He became an Assistant Professor at the Information Engineering Department at the University of Tokyo in 1993, and after moving to the Associate Professorship position at the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, in 1996, he is now a Professor at the Computing Center of Tokyo Institute of Technology, starting from April, 2001. His current interests are in combining object-oriented, high-performance software technologies, and high-performance global-scale computing, as well as highly-available clusters. His current research projects include the Ninf project, a high-performance global computing system, and the OpenJIT project, which applies reflection technology to a Java Just-In-Time compiler for parallel platform-specific customization and optimization, and various commodity clustering technologies. He is a member of ACM and IEEE Computer Society. He has received the Sakai award for research excellence from the Information Processing Society of Japan in 1999.