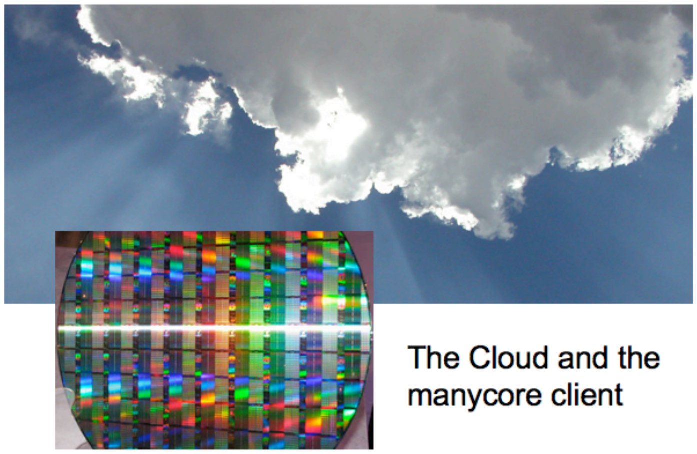
The Grid, Cloud Computing and our Manycore Future

Dennis Gannon
School of Informatics
Indiana University

The Grid concept originated with a brilliant vision. Larry Smarr was the first to articulate the idea of computing as a utility that operated like the electrical power grid. You could “plug in” anywhere and information and computational services were instantly at your disposal. When we use the electrical power grid, we don’t concern ourselves with which power generator provides us the electricity. We just use it. The computational and data Grid should operate the same way. For information services, the Web already operates this way. We don’t care which of Google’s compute farms processed our queries or where the data I down-loaded was stored.

But what about computational services? The Grid research community has been trying to build the standards to make Smarr’s vision a reality for nine years, but this year we have been confronted with some ideas that are altering the landscape. In fact, 2007 may be remembered as the year the future for Grids became very “cloudy”. The industry giants (Amazon, Google, IBM, Microsoft, Yahoo) have been building massive data centers at a rate that is completely outpacing the growth of the national supercomputing centers. These compute and data clouds are transforming our vision of what a Grid can be. Google led the way by showing how we can a use a large cluster of machines to do massively parallel data analysis tasks by uploading analysis codes into a ``MapReduce’’ software framework. Amazon demonstrated how a computing cloud could upload a virtual machine image to run arbitrary user applications as web services. Yahoo has made their computing cloud available to students to do further research on the MapReduce model. Microsoft has announced a billion dollar investment in data and compute clouds.



The Cloud and the manycore client

So what does manycore computing have to do with this? If we envision desktop machines with 32 or more cores by the end of the decade and a hundred soon to follow, there are profound implications for our digital life. For example, the Linked Environments for Atmospheric Discovery (LEAD) project has built service-oriented architecture (http://www.leadproject.org) to support research in severe storm prediction. This SOA requires a rack of a dozen quad-core machines to support the user portal, data, metadata and data provenance management, workflow orchestration, fault tolerance, and a pub/sub event system. This SOA is used to orchestrate and record the output from large-scale weather data analysis and simulations running on the TeraGrid. In the near future we will be able to run this entire SOA on a single manycore desktop machine where it will run more efficiently and reliably. That same desktop can reach out to the cloud to do the big simulation computing. Furthermore, our interaction with the system will become far richer because we no longer need to do it through a web browser. The entire middle layer of Grid services becomes part of our personal productivity environment.

The potential synergy between my manycore desktop/laptop/phone and the cloud go far beyond the current generation of Grid middleware. We anticipate having dozens of agents running on our personal devices interacting with the cloud and monitoring the computations and data feeds on topics of interest to us. The explosion of networked sensor data and video feeds together with advanced data mining and image recognition tools all running continuously on a manycore client can drive rule-based systems that know how to alert us when something of interest happens. For example, knowing when a friend came to the front door while we were away or predicting an impending health crisis of a remote relative.

We use the computer to create and explore. Our explorations consist of database queries, web searches, interactive sessions with tools like Matlab, Mathematica, and experimental workflows that invoke multiple tools and remote services. Explorations can lead to dead-ends or to new knowledge. With enough computing power, one of my agents can track and tag my explorations, pruning out the dead-ends and recording a provenance of each discovery. If I use a graph, table or image in a document that resulted from such an exploration, the system can tag it with its provenance record. Furthermore, the provenance record is "re-executable". If new data arrives that updates data that was used to make a discovery, the system can detect this and automatically redo the discovery process and notify me with the results. The system can build a network of discovery trails and artifacts that constitute a web of my knowledge. As I begin a new exploration, the system can remember related explorations and, like a good apprentice, pre-fetch potentially useful data or do preliminary analysis.

Introducing the data center cloud into the picture adds new dimensions to the process of problem solving. Small teams of collaborators or large communities that share a common interest will be able to create Virtual Organizations (VOs) using cloud services to set up shared data and application spaces. The process should be as easy as setting up a conference call. The agents on our multicore client automatically upload and share that part of our personal knowledge web that we wish to contribute to the community. Just as the Web has become the repository of the writings of a billion people, the cloud can be the enabler of the collective problem solving skills for communities of all sizes and types. For example, with my new manycore workstation I may be trying to solve a problem by sifting through my own knowledge web. For example, how can I can I predict the function of an enzyme from sequence similarity? I do this by running sets of analysis workflows using the data and cloud services. However, my agents, in consultation with the cloud services, recognize that my colleague Sun Kim and Yaoqi Zhou have looked at similar problems exploring structural similarity. What emerges is a proposed solution using both methods. How does this happen? Daniel Wegner has written about transactive memory in social cognition. The idea is that we may not know all the details to solve a puzzle, but it may be that the missing pieces are part of the collective knowledge of our community. Of course, exploiting this fact is easy if we can always remember who knows what. The challenge occurs when somebody with a key to my puzzle does not know they have the key. Unfortunately, this is the common case. When do you know that you have a special insight into a problem that somebody else is trying to solve? Our manycore client agents will have to constantly mine the content of the cloud matching our problem solving patterns and behavior against that of others. Together the cloud and our network of manycore agents will cooperate in ways that are today not possible.

Industry is deploying massive data center clouds because it is a cost-effective way to manage complexity and provide scalable, on-demand services. With the explosive growth in the computing power of our manycore clients, coupled with the impact manycore will have on the architecture of the data centers themselves, the capability of this collective resource will be remarkable.