Notes from the "Peer to Peer and Grid: Synergies and Opportunities" Workshop

GGF9, October 5, 2003

Chicago, Illinois

Organizers: Andrew Chien (UCSD), Karan Bhatia (UCSD and SDSC), Dejan Milojicic (Hewlett-Packard), Adriana Iamnitchi (U Chicago), Ian Foster (U Chicago)

Courtesy of George Clapp (Telcordia)

I. P2P and Grid Computing; Panel 1.
   A. Karan: introduction
   B. Brian Cooper, GA Tech
      1. Data management; grids is about computation over data. Ability to specify a query and get a result; computed efficiently. Be unaware of computation of the cloud. Data management people need to be aware. Users don't need to sweat the small stuff; can focus on domain specific application.
      2. Example 1: PIER

         Distr data; declarative queries. Query planning to determine which resources collaborate to give you the data you need. System works to pull together the data. Challenges: heterogeneity. Opportunity: leverage resources.
      3. Example 2: Piazza

         data integration; data has different schema: diff fields, relationships, constraints. Schema aren't always the same. If no intention to collaborate; won't interoperate. How to make mappings to collaborate productively? Hard when there are lots of sources.
      4. Example 3: SIL: Search Index?

         Another problem: distribution and heterogeneity don't scale; there is the problem of data resource allocation and management. Strategically place indices, caches.

         Selfish decision yield more efficient results. Individual resources monitor their load and monitor and make and break links. If many autonomous nodes, centralized mgt system can't assign load.

         Denial of service attacks? Yes, many people are looking at that: security, trust, denial of service. There is the question of mechanism design: how to incent nodes to take on load.
      5. Example 4: Chord, CAN, Tapestry, Pastry: data-centric routing

         Efficient application routing: how to get data to the people who need it? "Publish and subscribe" system. Register interest. DHT: Distributed Hash Tables. Efficient routing of data to people interested in it.
      6. Summary

         System offers more scale, self-organizing data communities, and high performance data processing; hidden success.

         Centralized approach won't work for reasons of scale and organizational practices.

         What state is the community in? Research prototypes available for downloading and trial. Industrial prototypes. EII marketplace. IBM's information integrator.

         Point well taken; take those systems and scale up.

         Scale? Put some numbers? Kazaa: very large scale system: 4M simultaneous users; petabytes of data. Semantics are very very relaxed. Keyword searches but not much else. Holy Grail: 4M users with SQL queries.

         Use in industrial setting? Make it so easy to be a part of the infrastructure; take existing infrastructure and leverage it. Industry is concerned about how their resources are being used; security. Take existing data set and integrate into p2p grid.

Correctness of response: local relational model. Relaxed consistency; most popular of most recent… Unsolved challenge: meet everyone's need.

Replicability: not covered by this model. not production quality; smart people are developing them.

C. Alex Mallet, Development Manager, Peer Networking Group, MS
   1. p2p investment

      Shipped SDK in 7/03: adv networking pack for Windows XP. Common functionality; key capabilities. APIs for the capabilities. Serverless DNS. Security model not based on centralized authority.

      Found that many people are behind NATs. IPv6 NAT traversal (Teredo)
   2. General principles

      MS is a big target
   3. Key capabilities
      a. Name registration and resolution

         Publish end points on home PCs. Serverless.

         Multiple endpoints on a machine (multiple services). People don't have static IP addresses. Dynamic endpoints.

         (1). Easy way to map friendly names to unfriendly names (mechanically produced)

         (2). Other initiatives in this problem: classic problem.

            (a). I3C: LSIV

            (b). GGF:

            (c). Handle.net: CNRI
      b. Data Distribution:

         (1). Continuum between efficient and robust; happy medium. Rooted tree sometimes.

         (2). Multiple transports

         (3). Disconnected case
      c. Persistent Peer Groups

         Rejoin when back online. Membership restrictions. Similar to the "Groove" model. Similar, though Groove doesn't have the notion of insecure groups.
      d. Security

         Signing and validation; people sending data have authorization.

         Make sure people could plug in their own security measures.
      e. Management and Deployment

         Real world experience. Need mgt and debugging knobs. Hard: few insights into the system. Non-sophisticated users. Install and "do the right thing."
      f. Programming Model: still trying to figure out. If you can't explain it, you certainly can't test it. Wound up with a best effort.

         Question: applications? 3 Degrees: shared play list of music. Corel: collaboration of note takers on tablets.
      g. Miscellaneous
   4. Key capabilities?
   5. Questions: not built on .net? No. Like Pastry: name resolution is similar but not built on Pastry. Reliable delivery: provide none. SRM: scalable reliable multicast. Provide folks with our version of key capabilities. Not trying to sell it!

D. Geoffrey Fox, Indiana U
   1. Focus on messaging
   2. Narada Brokering: p2p/grid model; entities talking together. Deliver messages reliably and securely.

3. JXTA plays a prominent role. Interoperation between peer groups. JMS: Java Message Service. NWS: Network Weather Service.
4. Cellular Grids: "Gridlet" interconnected by messaging. Generalized p2p with each node a grid. JXTA example. Each JXTA Peer Group is a gridlet. Mediation agent: routing, I/F mapping,

   Autonomic: save messages so "transaction" is replayable.
5. Only 30 minute notice of need to give a presentation.

E. Adriana Iamnitchi, U Chicago, What are the Capabilities of P2P that Grids Should Exploit?
   1. Looking at deployed systems
   2. Certainty: convergence of p2p and grid computing
   3. Comparison of the communities
   4. Comparison of the resources
   5. Statement of similarity in Grid 20 years ago (multiple parallel efforts) with P2P today. Difference is that the user base is not mature. Greece: can point out the app's with large computation requirements; p2p has not.
   6. Question: p2p called grids and grids called p2p? Hype engine!
   7. Convergent environment
      a. Normal path is for Grids to grow in scale
      b. What applications have that many people who are interested in them?
      c. Interesting parameters to *connect* grids and p2p rather than separate them.

F. Jim Browne, U Texas
   1. Distinguishing characteristics of grid and p2p? OGSA? OGSI? Sufficiently fuzzy as to not provide a definition.
   2. Neither have clear goals.
   3. Suggest fundamental differences: way to put them together. Work towards an integration? Unsure? Incorporation: standards for grids; p2p; clean i/f between to leverage strength of each. Claim that there are fundamental differences. Both are instances of distributed systems. Specification of assumptions that underlie grid and p2p; not easy. Know a little about distributed systems.
   4. Grids: goal: construct virtual organizations? P2P: Virtual System. Common Knowledge base of participants? Grids: central state; p2p: protocols. Grids implicitly specify central control; not grids. Member of audience: underlying Grid services are long running. Centralized vs. decentralized control. Scalability: fully decentralized.
   5. Commonality: low level implementation mechanism is ultimately a sophisticated procedure invocation with composition by control flow and direction of output.
   6. What we'd like to get in grids: adaptive optimization and fault-tolerance. Have these in p2p. Metrics and evaluation. Discussed in a quantitative fashion with crisp definitions.

G. Discussion:
   1. DHT: distributed hash tables. Grid infrastructure; p2p functionality. Incorporate p2p functionality in grid infrastructure.
   2. What type of grid are you trying to build? Scaling may not be that critical for some applications?
   3. Changing: production grids and those to be: definitely a notion of having more users and decentralization; limits to current technology. P2P has advantages to address those problems. People here recognize that their grid systems must deal with these problems.
   4. Consider the distinction between consumers and providers; predominantly one role or another? predominantly both? Scientific data sets and music files. Hard time understanding, if having a hard time doing computation, why should I share cycles? Tragedy of the commons. Radically asymmetric users: 10% of users consuming large portion of resources.
   5. Understand what is needed to build the system? Learn from other solutions.
   6. P2P begins with decentralization and no centralized control.

7. Not imaginative chaos. There are means of control. P2P are not anarchic. Control doesn't look like a business organization.
8. People who are building and deploying P2P are not purists about decentralization. Directories make things easier; super nodes. Purists aren't going to get far.
9. Difference between grids and p2p: p2p is not just distr control; it's distr data. Where to find this chunk of data. Grids: where to find a service. Data generated by service can be cached and moved around the network.
10. Big area of data grids; look a lot like p2p.
11. Need both: some resources which are by nature centralized, e.g., particle accelerator. Desemination.

II. P2P and Grid Computing: Panel 2
   A. Andrew Grimshaw; founder of Avaki
      1. P2P: included small devices such as Palms, phones. Grid: started with supercomputers and working their way down.
      2. OGSI is a spec; OGSA is a collection of interesting ideas.
      3. Policy description and negotiation
      4. P2P hasn't done "naming and binding" well. XML is something everyone can agree upon.
      5. Byzantine security models in the wild and woolly Internet. Peer devices may be arbitrarily malicious. Certification levels; "web of trust."
      6. "Simple and lightweight;" construct from that something highly functional. Neuron analogy.
   B. Sergio Mendiola, Oracle
      1. P2P and Grids: Synergies and Opportunities.
      2. Knob from centralized to decentralized.
      3. Scalability: assumption of Web Services scalability
      4. Things are in the OGSA in nascent form. Just a vehicle for other existing technologies?
      5. No requirement for decentralization, so self-healing.
      6. Word "scale" and "scaleability" only appear once in the entire OGSA document." Stated by a co-author of the document. It is an issue.
   C. ?, Bloomington, University of Indiana
      1. Suns JXTA: another model to build P2P
      2. Grid community has scale envy when it looks at P2P. A few nodes are super peers. "Power-law networks." Successful P2P evolves to this state. Where does control lie. Clumped in an interesting way.
      3. Chien: contradiction to fixed degree, homogeneous.
      4. Limited applications: file sharing.
      5. Lot of talk about Web Services and P2P but no evidence of action.
      6. Question: do grid standards like OGSI/OGSA help the cause for P2P.
         a. No
         b. Well, don't hurt
         c. Perhaps
         d. Yes, but when thought of in a different way.
      7. What can be leveraged? OGSA: ubiquitous distributed services to allow us to think of OGSI and resources as a simple computer. Chien: distr operating system.
      8. Rich layer of services to enable a richer set of P2P applications.
      9. Not convinced that Grids are scalable. Built on Web Services that are designed to scale.
      10. Grid: Virtualizing resource management. P2P doesn't do that at all. Grid looking at SLAs carefully.
      11. Tests of XML signatures in SOAP. Would hate to do too many chains of signature validation.
      12. Think that most Grid s/w is too heavyweight. Static IP isn't going to work.
   D. Andrew Chien, UCSD

1. List of focus areas for OGSA. Very telling of mindset. Management, control, ownership mentality. New capability, perhaps higher quality and manageability.
2. Focus on things thought to be expensive.
3. Not clear that this is the right list for P2P! Focus may not be right. Synergy may not be so obvious.
4. P2P applications: two clusters: people with large scale deployed systems: cycle scavenging, etc. Up and running for years. "Make big complex things that you are stuck with, work together productively." Second half of P2P world: academic. Not pursuing 100sK users but DHT and systems built atop it. Not confused with folks running the large scale systems. Characterized by assumptions different than grid assumptions.
5. Moderate control infrastructure, and that's being kind. Can't predict and control. Marked contrast to "monitoring, accounting, SLAs, etc."
6. P2P: functioning systems without control; early in the evolution with a few specific applications success stories.
7. Question: hit on a key difference; corporate IP is critically important. Corporations won't risk their IP. Agree.
8. Grids: one size fits all. Infrastructures that are widely shared; not homogeneous. Globus has been successful because it started with the toolkit.
9. PlanetLab analogy.
10. Converging, parallel, diverging? Nature of applications?
11. They will have to talk together, but not in a single infrastructure!

E. Karan Bhatia, SDSC
1. Grid app's: NSF TeraGrid. Teragrid is not P2P. Homogeneous clusters of machine; huge networking component. 4-5 sites. Very administratively centralized structure.
2. NPACI Grid: more heterogeneous: equipment and institutions. PRAGMA (sp?) Grid: pacific rim. GeonGrid: geo sciences community: resources are data. Has to be decentralized. Best-effort. Very autonomous; themes of P2P. Distributed and autonomously managed. 20 different institutions; scaling to 100. Grid systems are looking for some subset to be more like P2P.
3. Requirements:
   a. NATs and DHCP (laptops)
   b. User interface? Not part of the grid? Data is located on my laptop; want the results on my laptop. Want to be part of the grid to run my app.
   c. No: just a user.
   d. Laptop has to have a service end point.
   e. Just a certificate
4. Security
5. Value proposition isn't clear; understand the value prop for my systems, though.

F. Discussion
1. OGSA discussions focusing on Byzantine settings? Not in the security groups; punted to the security group. They are not looking at Byzantine: highly structured, Certificate based.
2. Where to take the authorization group that we just started. Requirements for policy language. Conceptually, running far ahead. Delegate rights, but no way to implement; no common language to express. Standardize on a policy language. Need in grid and P2P. Understand statements about trust, rights, and delegation of rights.
3. Need conventions about the semantics.
4. XML-based.
5. Concept of Agreements: machine is reliable, degree of security. Defining the profiles of machines. Not just security; wide range.
6. Discussion is premature. Taking products, stovepipe solutions. OGSA standards. Mismatch. None of OGSA are in place. What is the GGF trying to solve? Help to P2P: where to embed standards. P2P between machines, not services. Other modes to establish environment.

7. Andrew: agree that it's early. Problem: would be too late if we waited until that is true. Raise people awareness that there are underlying forces that will lead to their being less useful.
8. Analogy: goal for the ARPANET: communication in the presence of large number of faults. Contrasted to the telephone company. Grids look like telephone companies. P2P look like the Internet. They will get together. Need now to identify some structure in the P2P community to get them to work with Grid. Some folks in corporate world using P2P and Grid; see if they wouldn't be interested.
9. Alan Weissburger (sp?): question wrong: grid standards applicable? No standards. Documents imply that Grid Services are Web Services. Web Services Distributed Management WG. Are Web Services standards applicable to P2P.
10. Searched the web looking for answers to that question: P2P using Web Services.
11. Web Services are too top heavy for P2P.
12. Disagree: boils down to RPC; means using XML over SOAP; all the tooling that is coming out, e.g., .net. WDSL itself is a useful standard. SOAP isn't bound to its lower level comm – but it is. Web Services isn't necessarily the wrong thing for P2P, but it isn't there.
13. Alan: propose that P2P and Grid are totally incompatible because Grid is build on Web Services.
14. Web Services aren't that widely adopted yet.
15. Been looking at implementing more general computations using P2P. Web Services, if not so cumbersome, would be perfect. One thing that could be done: define middleweight and lightweight of RPC as a standard for P2P comm.
16. GSR doesn't have to be a binding.
17. Also want to send code to execute. Very easy thing to do. Not rocket science.
18. Andrew: preserve a toolkit approach. Subset the architecture: take this piece. You want one real small piece.
19. Get some success on a small piece. NSF sponsored project: P2P and Web Services computation.
20. Grid service handle and resolution to grid services references; very general. Scale where referencing; GSH for every doc in the world. Very interesting push to a different type of scale. P2P community with a notion of handle and handle resolution. PlanetLab. Conflicting security model.
21. Couple of things missing in handle resolution. Resolution itself. Depends on name space. Plans on the side: on line global registries. Work to be done to address that issue. Scalable name resolution. How would it help the P2P if that were available.
22. Name resolution and binding is something that everyone needs.
23. Andrew: benefit of interoperating with those guys. Incentives for interoperation before adoption.
24. Incentive would be there if part of a toolkit and part of a global name.
25. Problem of Web Services being top heavy may disappear over time. Plan for the probable future.
26. Andrew: listed foci of OGSA that are perceived to be expensive. Accounting and resource management really important in huge hard drives and bandwidths.
27. Accounting isn't just because things are valuable but for auditing. Looking ahead 10 years is a good idea, but some things are needed now: naming, security.
28. End-to-end connectivity isn't guaranteed: IPv6 isn't being adopted, NATs. Scale and decentralization will be more of a problem.
29. P2P will be forced into Web Services; will be non-standard.
30. As you scale up, you have to assume fault tolerance. Tremendous relief, don't have to compute MTTF.
31. Andrew: when things fail, have to assume that they will fail in arbitrary and Byzantine ways.
32. Literature assumed fail-stop. Unix doesn't fail-stop. Can't afford the overhead of Byzantine model. 80% of our code is exception handling.

33. Looking 10 years ahead: top edge goes up the scale; bottom moves down the scale. Greater heterogeneity. Web browsing on cell phones. Don't push the argument to heavyweight things. History says it will. Increased scale and heterogeneity. Push down to smaller devices and more ubiquitous.
34. PDAs now are like the machines we used to do. Big crypto on those is really expensive. Crypto will migrate to devices.
35. Grids have tended to focus on middle to high end. P2P will be focused on the range.
36. Andrew: note that moving down is harder than moving up.
37. Smart dust: half megabyte memory; MHz processor.
38. PDAs won't be part of the grid; they offer no grid service.
39. Imagine that the grid are all the devices here in the room (laptops). No, because you can disconnect now and go home. Suppose a new phone with camera and memory: does offer services: access control device, same security model with phones as with anything out there. From security perspective, since devices are data sources.
40. Looked at app's and use cases of grids and p2p. Two categories: large scale, scientific. Mass consumer file sharing. Insufficient treatment of business applications. How businesses are affecting this debate?
41. In industry today, businesses are using grid. Integration or access to data between or within organizations. Share info in real time. Contractors who aren't necessarily trusted. Life sciences: huge amount.
42. OGSA: motivations are enterprise level applications. Very large concern. If not there, they'll walk. 80% of OGSA working people are industrial.
43. No mid-level app's for P2P is not clear that businesses wanted to invest resources in building infrastructure. Hope will help by the emergence of toolkits, MS sol'n. Can't speak to the grid stuff.
44. If not using standards like CORBA, why use standards like Web Services? "They'll see the light!"
45. Don't think they will use web services. P2P trust themselves and not others. Will go for lowest common denominator that they trust, sockets.. Avoid admins and thought police.
46. Andrew: zero incremental cost.
47. P2P larger than pirates. They don't want to get caught. Not the determining factor for legitimate P2P.
48. Enterprises want to keep their own domain.
49. If build a system that is secure: anonymity, trust. Need a security infrastructure. Not too many sol'ns there.
50. Difference between grid and p2p is the amount of money they have to spend. Any good P2P enterprise applications?
51. Collaboration technology has a P2P flavor. Pick up Web Services component. Starting to do that.
52. Common standard.
53. When US commercial people talk with potential customers about Web Services and P2P? Our customers couldn't care less. Sol'n to a problem. Some care about Web Services and OGSA, but I swear they don't know what they're talking about (OGSA).
54. Andrew Chien: interoperability; long term tech strategy; interoperability and manageability. Manageability and integration with existing security infrastructure.
55. Sun person: JXTA group has applied to IETF. JNGI.

G. One Minute Statements
1. Andrew: life is complex; real systems are complex. Myriad ways to fail. Construct abstractions; OGSA allows us to build layers systems and hide complexity.
2. To convince P2P users to use Web Services will be hard; toolkit is a good approach.
3. Grid community – before going to P2P comm. to tell them what they should do – is let their hair down and rock and roll with them; learn what they're problems are. Then start to build P2P, then rest of the world will care.

4. Andrew C: fielded by diff communities. Grids have emerged as enterprise intranet app. P2P as internet: global resource synergy. PlanetLab analogy. None knows what the outcome will be. Example in networking. Internet synergy created IP-based networking.
5. Advantage in both ways. Sol'ns to specific problems can help. Don't have to solve the problem. Get your stuff out faster.

H. Question: propose actionable items?
1. Grid research oversight committee: strongly suggest that this group put together to write GGF info docu based on observations from this workshop.
2. Send notes to Cees
3. Andrew C: solicit ideas and proposals.
4. P2P WG in GGF. Members to participate to seed some ideas.
5. Do want to I/F with these groups; question of structure. Requirements, e.g., scalability.