The convergence of P2P and Grid Computing
Foster and Iamnitchi, IPTPS 2003

Alan Sussman

CMSC 818S

March 13, 2007

Notes

- Project schedule by Thursday
  - groups?
  - ideas?
- Postpone Ninja paper until after break (and after I get back from IPDPS)

Death, taxes, etc.

- Grid and P2P computing target the same general problem:
  - organization of resource sharing within virtual communities
- And take the same general approach
  - create overlay structures that exist on top of underlying organizational structures
- "Grid computing addresses infrastructure but not yet failure, whereas P2P addresses failure but not yet infrastructure"

Grids vs. P2P

- Grids – sharing environments
  - persistent, standards-based services to share resources within distributed communities
  - resources owned by physical organizations, shared via locally defined policies – what, who, and when – a virtual organization (VO)
- P2P – applications that take advantage of resources at the edges of the Internet
  - unstable connectivity, network addresses, etc., so no reliance on central servers
  - create overlay networks in implementations

Target communities & incentives

- Grids motivated by needs of scientific users needing access to remote resources (datasets and/or computers)
  - users willing to devote effort to create and operate infrastructure
  - some degree of trust and accountability
  - VOs can’t impose solutions on local owners
- P2P more grass-roots – file sharing and client-server cycle sharing
  - large scale in number and diversity of users
  - need to detect and deal with resource misuse

Resources

- Grid resources can be more powerful, diverse and better connected than P2P resources
  - so need explicit administration according to well-defined policies
  - QoS important, but increases integration cost
  - leads to needing information services, to determine resource properties
  - Availability tends to be higher, and more uniform
- Grids can integrate desktop resources too
  - but such resources are aggregated into administrative domains (e.g., Condor) to integrate into Grid
- Most P2P resources less powerful, home or desktop computers

Applications

- Grid apps very diverse across communities
  - remote access to supercomputers
  - desktop grid apps
  - virtual laboratories – remote instrument access
- P2P systems often vertically integrated
  - file sharing
  - cycle sharing
  - BOINC is an exception
  - diversity comes from different design goals – scalability, anonymity, availability, etc.
- Grid apps often data intensive (high network/disk throughput rates), but P2P apps aren't (even file sharing?)
  - maybe not true for P2P streaming video, if that ever happens

Scale and failure

- Scientific Grids modest scale in number of participants (institutions, pooled computers, simultaneous users)
  - so scalability not a big issue until recently, since centralized solutions sufficed
- P2P communities can consist of millions of nodes (for file sharing and cycle sharing)
  - so need robust self management of large numbers of nodes
- But total amount of activity in P2P systems not quite as big as it seems
  - 1-2TB per day in file sharing systems at end of 2001
  - 3-4TB per day in D0 experiment Grid example

Services and infrastructure

- Grid community has created persistent, multipurpose infrastruture services
  - e.g., authentication, authorization, discovery, resource access, data movement, etc.
  - overlays on resources and services maintained by participating institutions, but still enforce local policies
- P2P systems have focused on integrating simple resources (e.g., files/cycles on individual computers)
  - protocols provide specific, vertically integrated services
  - each P2P app has its own protocols and interfaces to its services

Summary

- Both Grid and P2P have the same vision
  - a worldwide virtual computer where access to resources and services can be obtained when and where needed
- Need to combine strengths of both to
  - address failure using scalable, self-configuring protocols
  - provide persistent and multipurpose infrastructure, supported in an organized, distributed way to achieve robustness, performanced, and trust