Grid and P2P: compare and contrast

Awalin Shopan

- Same general **problem**: the organization of large set of resource sharing within virtual communities.
- Same general approach to **solution**: the creation of overlay structures that coexist underlying organizational structures.
- "**Grid** computing addresses **infrastructure** but not yet failure, whereas **P2P addresses failure** but not yet infrastructure".

Target communities

GRID:

– professional communities needing to access remote resources.

– Trust, accountability, and opportunities for sanctions in response to inappropriate behavior

P2P:

– grass-roots, mass-culture file-sharing.

– Diverse and anonymous individuals with little incentive to act cooperatively.

Resources

Grid:

– powerful, more diverse, and better connected

– cluster, storage system, database, or scientific instruments

– administered in an organized fashion according to some well-defined policy

– availability tends to be higher and more uniform

P2P:

– intermittent participation

– highly variable behavior

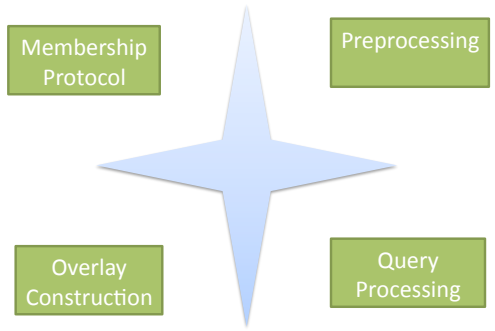
Application

- Grid: more integration, data intensive, numeric simulation, optimization problems
- P2P: specialized resource-sharing problems

- Goal: convergence of Grid and P2P into one
  - Grid lacks supports for intermittent resource participation
  - P2P lacks attribute-based search
- Concerns for Grid systems:
  - Scalability
  - Independence from global control

P2P approach to Resource Location in Grid

Architectural Components of the Proposed Solution Space



Emulated Grid

- Extensible
- Nodes forming an overlay network
- Node=process
- Communication via TCP
- Information at each node:
  - Resource: info about a set of resources
  - Membership: about other nodes in the overlay network

Environment variables

1. Resource information distribution & density
2. Resource information dynamism
3. Request popularity distributions
4. Peer participation

Implemented:
- Resource information distribution
- Request popularity distributions

Assumption:
- Static resource attribute (ignoring 2)
- Constant peer participation (ignoring 4)
- No failures

Experimental Setup

Resource distributions
- Balanced
- geometric

Request distributions
- One week's requests for computers in CONDOR
- Synthetic request with uniform distribution

Components Setup

- Membership protocol:P2P like join
- Overlay function: unlimited neighbors
- Preprocessing: N/A
- Request processing: based on forwarding
  - Random walk
  - Learning based
  - Best neighbor
  - Learning-based + best neighbors

Unstructured, static network in the absence of preprocessing

Result

- Cost={response time, storage }
- Learning based consistently performs well
- Learning based + best neighbors: unpredictable, expensive
- Random walk: lowest cost, least efficient

- Best neighbor strategy performs better in uniform user distribution load, rather than unbalanced.
- In case of variable usage pattern: best strategy is to use Learning based forwarding.

Thank You