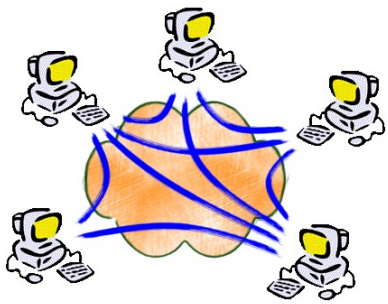
Robust and Effective Resource Management in Distributed Desktop Grids

Jik-Soo Kim
Department of Computer Science
University of Maryland

Desktop Grid and P2P System

From: www.eminerals.org/minigrid/condor.html

From: www.cs.virginia.edu/~mngroup/hypercast/



***Centralized* Server-Client Architecture**
Complex scientific applications

***Decentralized* Peer Networks**
File Sharing

***Convergence* of Grid and P2P**

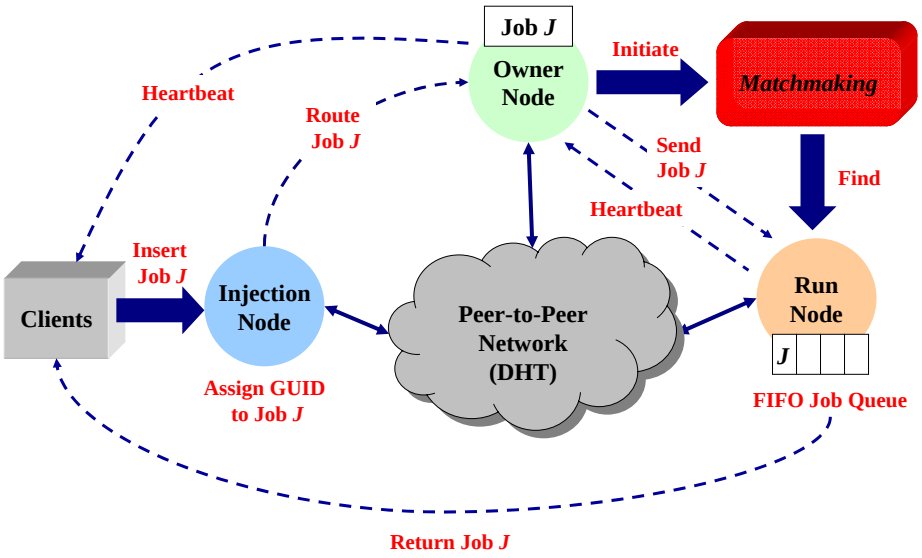
Hard Problems / Issues

- Job Submission
  - Submit a job into the *decentralized* P2P system
- Matchmaking
  - Find a resource that can *meet* the minimum resource requirements of a job
- Load balance
  - Distribute the load (jobs) across the nodes in the system
- Resilience to failure
  - Overall system must be resilient to the failures

Basic Assumptions

- Underlying *Distributed Hash Table* (DHT)
  - Object location and routing in a P2P network
  - *Reformulate* the problem of matchmaking to one of routing
- *Job* in the system
  - Data and associated profile
  - All jobs are *independent*
- Optimization criterion
  - Minimize time to complete all jobs (combination of throughput and response time)

System Architecture



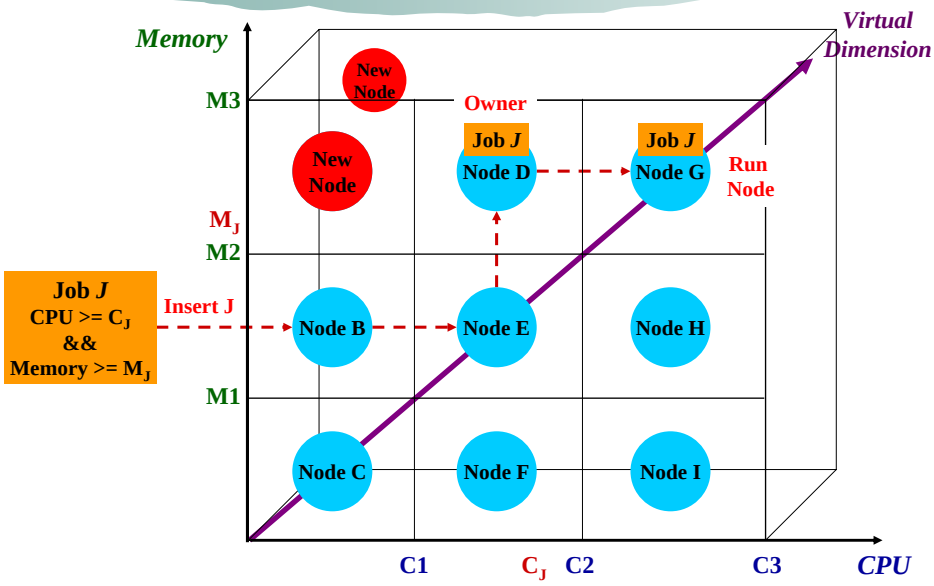
Goals of Matchmaking Algorithms

- *Expressiveness*
  - Allow users to specify any type of minimum resource requirements
- *Load balance*
  - Distribute load across multiple candidates
- *Parsimony*
  - Resources should not be wasted
- *Completeness*
  - A valid assignment of a job to a node must be found if such an assignment exists
- *Low overhead*
  - Routing must not add significant overhead

Modified Content-Addressable Network

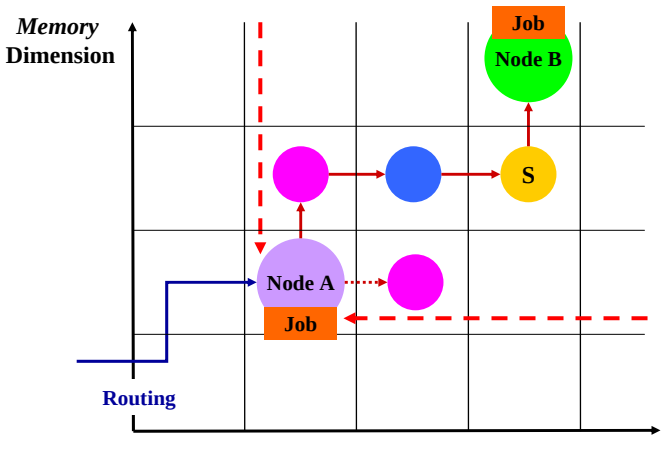
- Basic CAN
  - Logical *d*-dimensional space
    - zone, neighbors, greedy forwarding
- Formulate the matchmaking problem as a routing problem in CAN space
  - Treat each *resource type* as a distinct CAN dimension
  - Map nodes and jobs into the CAN space
    - Resource capabilities and requirements, respectively
  - Search for *the closest node whose coordinates in all dimensions meet or exceed the job's requirements*

Matchmaking and Load Balancing



Improving CAN-based Algorithm

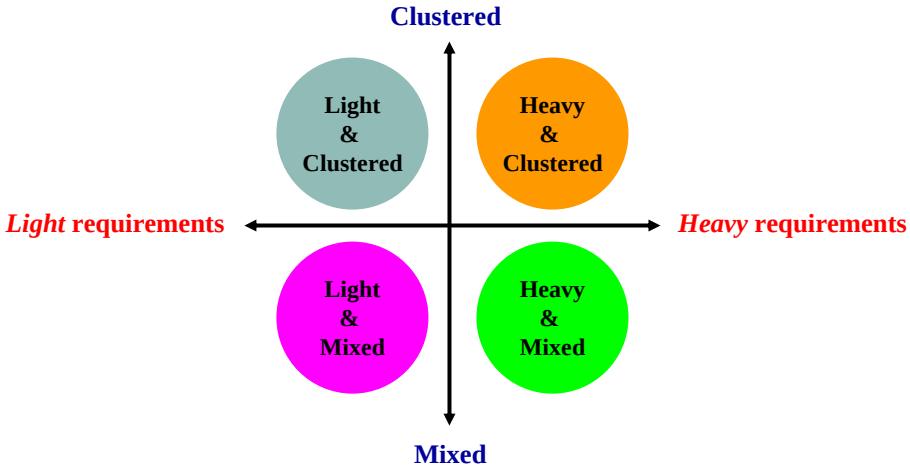
- *Dynamic Aggregated Load Information*



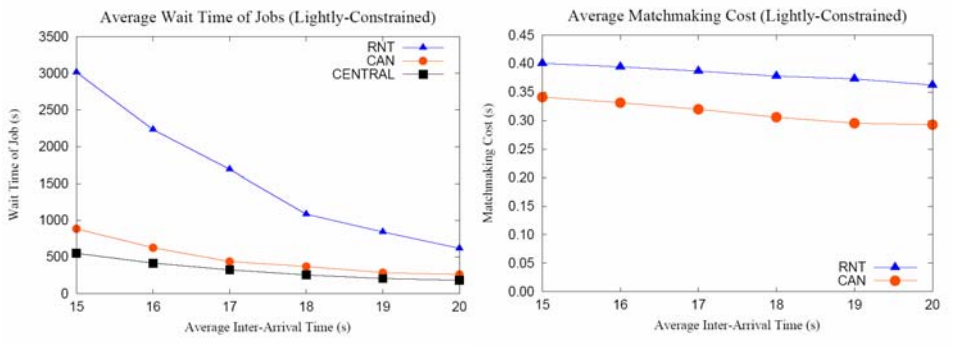
- Aggregate Load Information
- Choose the least loaded direction
- Push a job into under-loaded region
- Stop pushing
- Choose the best run node

Experiments through Simulations

- Workload Setup



Comparative Analysis



Current Status

- Summary of results from simulations
  - Grid 2006 and HPDC 2007
  - CAN algorithm balances load almost as well as centralized algorithm with low matchmaking overhead (few messages)
- Ongoing work
  - Handling a new type of resource dimensions
    - e.g. OS, system architecture
  - First CAN-based peer implementation in testing, and soon deployed to astronomy collaborators
    - Basic CAN services and authentication mechanisms
    - Job management and GUI client interface
    - Load balancing and other new mechanisms still being implemented

The Project Team

- Faculty members
  - Alan Sussman, Pete Keleher, Bobby Bhattacharjee, Derek Richardson (Astronomy), Dennis Wellnitz (Astronomy)
- Matchmaking algorithms and simulations
  - Jik-Soo Kim
- Prototype implementation
  - Michael Marsh, Beomseok Nam, Jaehwan Lee, San Ratanasanya
- Project funding from NASA and NSF
  - to develop algorithms, build and deploy the system

The End

Desktop Grids

- Client-server systems for running large numbers of (similar) jobs
  - server hands out work to clients connecting over the Internet (or intranet)
  - issues include security, reliability, and relatively limited scope of problems (massively parallel)
  - volunteer efforts include SETI@home (BOINC), Folding@home, prime number searches, World Community Grid (IBM)
  - companies include United Devices, Entropia, Parabon
    - business model is for use on corporate intranets, but not very successful

Workload Assumptions

- *Must* accommodate heterogeneous clusters of nodes running heterogeneous batches of jobs
- *Clustering* in nodes (resource capabilities) and jobs (requirements)
  - A small number of equivalent classes of nodes
  - Parameter sweeps, e.g., N-body or weather simulations

Nodes \ Jobs	Clustered	Mixed
Clustered		Condor
Mixed	BOINC/ SETI@Home	

Modified Content-Addressable Network

- *Virtual* Dimension
  - Clustering of nodes and jobs
    - Resource capabilities and Requirements
    - Distribution of ownership of a zone and Load imbalance
  - Supplement the *real* dimensions
    - Corresponding to node capabilities
  - Coordinates for nodes and jobs for the virtual dimension generated *uniformly at random*