



CMSC 818S Grid Computing

A Peer-to-Peer Approach to Resource Discovery in Grid Environments

Adriana Iamnitchi

Ian Foster

Daniel C. Nurmi

Binh Viet Nguyen

Content

- Introduction
- Resource Discovery in Large Scale Resource Sharing Environments
- Grid vs. Peer-to-Peer Environments
- Discovering Generic Resources in Unstructured Networks.
- Resource Discovery in an Emulated Grid
- Summary

Introduction

- In large sets of shared resources, a basic problem is locating resources in the absence of a naming scheme: a resource is often described as a set of desired attributes
- Resource discovery problem in a resource sharing environment that combines the complexity of computational Grids with the scale and dynamism of P2P communities

Convergence of Grid and P2P

- Current Grid: large variety of services provided to scientific communities of hundreds of users.
- P2P: boast a larger communities (millions) but offer limited, specialized services
- The convergence of computational grids and P2P.

Resource sharing in Grid

- Scale in the order of thousands of pooled computers and hundreds of simultaneous users.
- Some level of centralization
- Complex operations
- Stabilities in resource participation
- Some level of homogeneity in usage behavior
- Homogeneity in resources
- Existence of technical support personnel

Resource sharing in P2P

- Large scale: in the order of hundreds of thousands of simultaneous computers/users
- Less centralization
- Specialized functionality, often in the absence of trust enforcing mechanisms
- Unpredictable resource participation
- Uneven user behavior
- Highly heterogeneous resources
- Lack of technical expertise and administrative authority

Convergence of Grid and P2P

- Large scale
- Lack of global centralized authority
- Highly variable participation patterns
- Strong diversity in:
 - Shared resources
 - Sharing characteristics
- Lack of homogenous administrative support

Four Axes of the Solution Space

1. **Membership protocol** : how new nodes join the network and how nodes learn about each others
2. **Overlay construction function** selects the set of active collaborators from the local membership list.
3. **Preprocessing** refers to the off-line preparations for better search performance, independent of request. E.g dissemination of resource descriptions
4. **Request processing:**
 - Local processing: looking up the requested resource in the local information
 - Remote processing: refers to the request propagation rule: sending requests to other peers through various mechanisms (flooding, forwarding, epidemic communication)

Modeling the Grid environment

- 4 environment parameters that influence the performance and the design of a resource discovery mechanism:
 1. Resource information distribution and density: fairness of sharing
 2. Resource information dynamism: CPU load, available bandwidth
 3. Request distribution: pattern of users' requests for resources.
 4. Peer participation: influenced by incentives

Experiment setup

- Passive membership protocol. simple join mechanism: a node joins the grid by contacting a member node.
- The overlay function accepts unlimited number of neighbors.
- Assume no preprocessing
- Request processing mechanism: simple request and satisfiable only by perfect matches

Request Propagation strategies

- ***Random walks***
- ***Learning based:*** nodes learn from experience by recording the requests answered by other nodes.
- ***Best-neighbor rule:*** a request is forwarded to the peer who answered the largest number of requests
- ***Learning-based + best-neighbor***

Resource Distributions

- Average number of resources per node remains constant with the increase in the network size: N nodes:
 $100 + 5 \times N$
- The number of resources that can match a particular request increases as the number of nodes. The increase in the number of new resource types chosen for this experiments was 5% per node.
- Fair distribution with all nodes providing the same number of resources
- Unbalanced distribution: geometric distribution, in which most resources are provided by a small number of nodes.

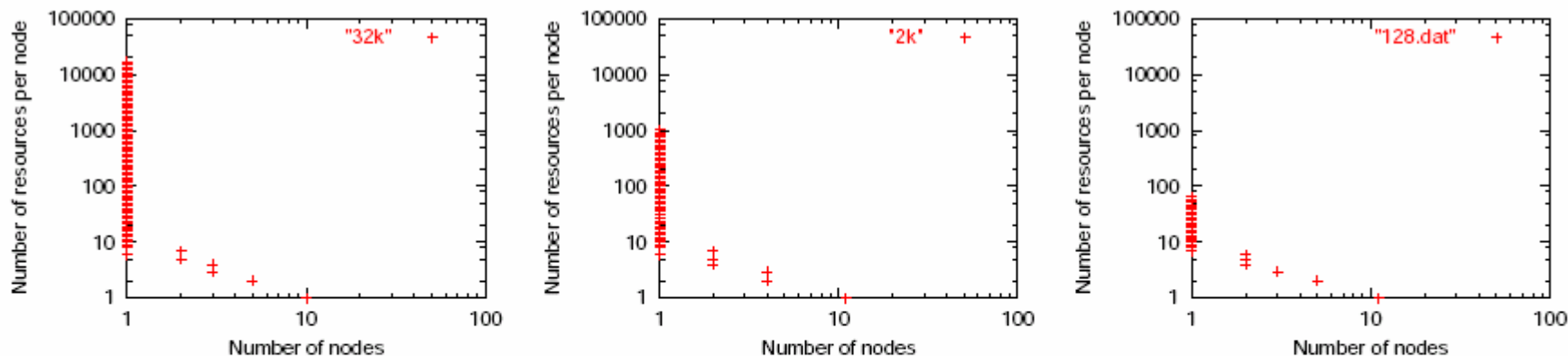


Figure 7: The highly unbalanced distribution of resources on nodes for three networks of sizes 32,768 (left), 2,048 (center) and 128 (right) nodes. **Left:** 1738 distinct types and a total of 163833 resources; number of identical resources varies from 67 to 139 (mean = 94). **Center:** 202 distinct types and a total of 10238 resources; number of identical resources varies from 34 to 70 (mean= 50) **Right:** 106 distinct types and a total of 640 resources; the number of identical resources varies from 2 to 13 (mean = 6)

User request

- We logged, processed, and used a one week's requests for computers submitted to the Condor pool at U. of Wisconsin: exhibit a **Zipf** distribution
- Synthetic user request distribution modeled as a uniform distribution
- We evaluated the above mentioned request propagation strategies in a set of overlay networks of size from 2^7 to 2^{15} nodes

Experimental Results

Objectives:

1. Estimate quantitatively the cost of simple resource discovery techniques based only on request-forwarding (no preprocessing)
2. Understand the effects of resource and request distributions on resource discovery performance
3. Understand the correlation between performance and various design decision in a realistic Grid environment: still in the process of collecting more data

Quantitative Estimation

- The learning-based strategies is the best performing under various sharing characteristics, with under 200 hops response time per request for the largest network in our experiment. It takes advantage of similarity in requests and uses a possibly large storage of cached requests
- Random forwarding algorithm: no additional storage space is required on nodes to record history but is the least efficient
- The learning-based + best neighbor is rather unpredictable (e.g. large standard error deviation for 1024 and 2048 simulated nodes).

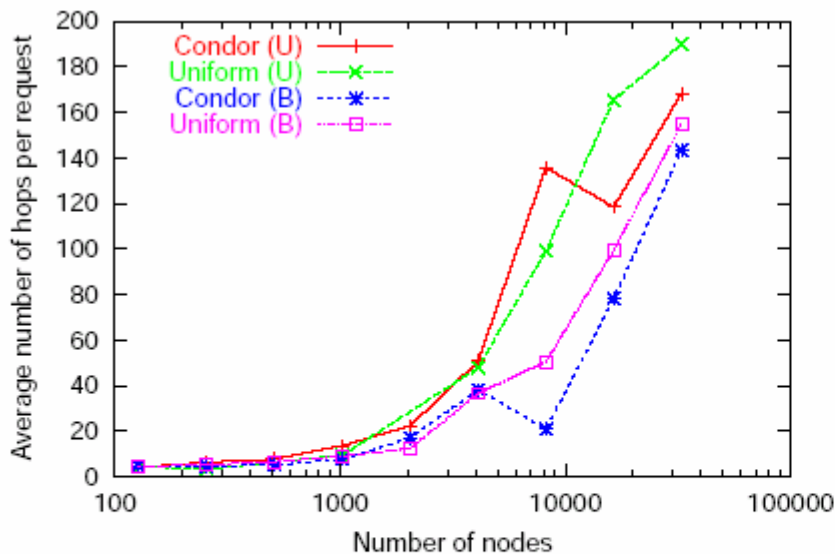


Figure 3: Performance (in average number of hops) of learning-based forwarding strategy for the three request distributions, in two environments with different resource sharing characteristics.

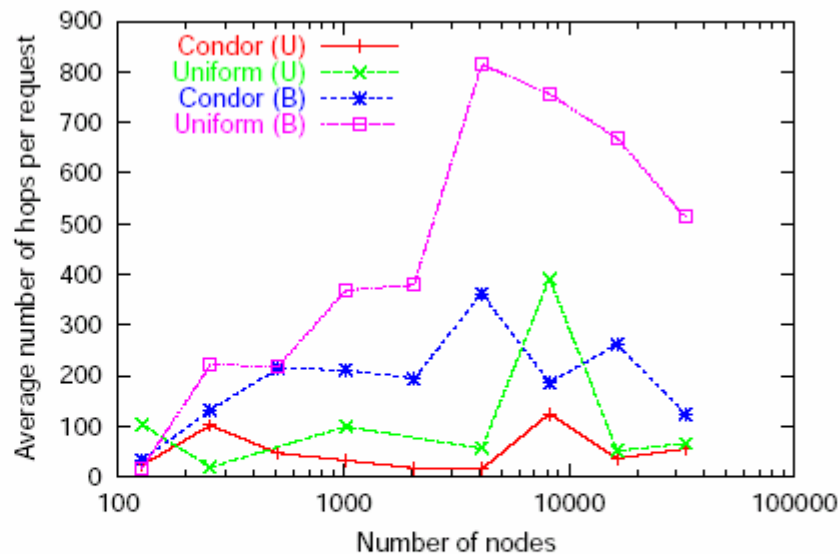


Figure 4: Performance (in average number of hops) of the best neighbor request forwarding strategy under different user request and sharing characteristics.

37

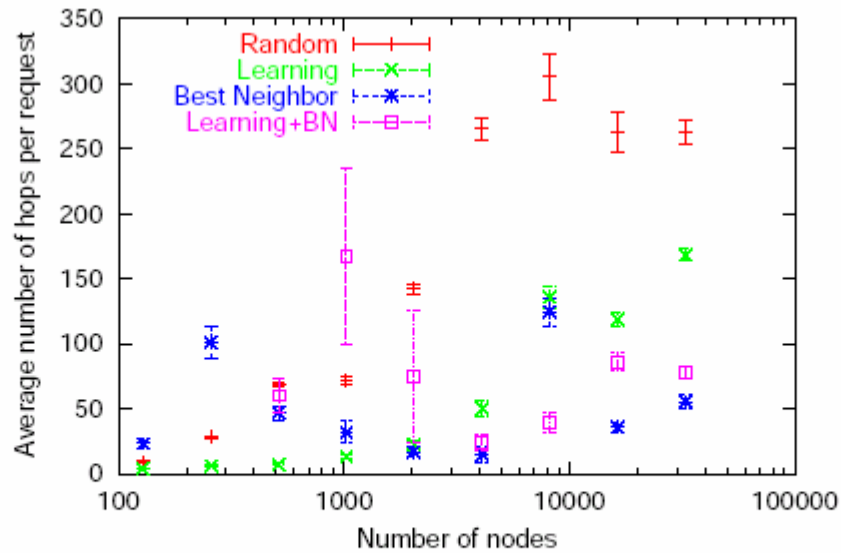


Figure 5: Performance off all four request forwarding strategies for a Condor request load in the unbalanced resource sharing environments.

Summary

- The characteristics and the design objectives of Grid and P2P environment will converge: Grids will increase scale and inherently will have to deal with more dynamism, P2P systems will provide more complex functionalities, integrating data and computation sharing with various security requirements.
- Propose 4 components that can define any decentralized resource discovery design: membership protocol, overlay function, preprocessing, and request processing
- Describe the emulator for evaluating resource discovery techniques and the quantitative measure of the influence of the sharing environment (i.e. fairness of sharing)