

Design and Implementation Tradeoffs for Wide-Area Resource Discovery

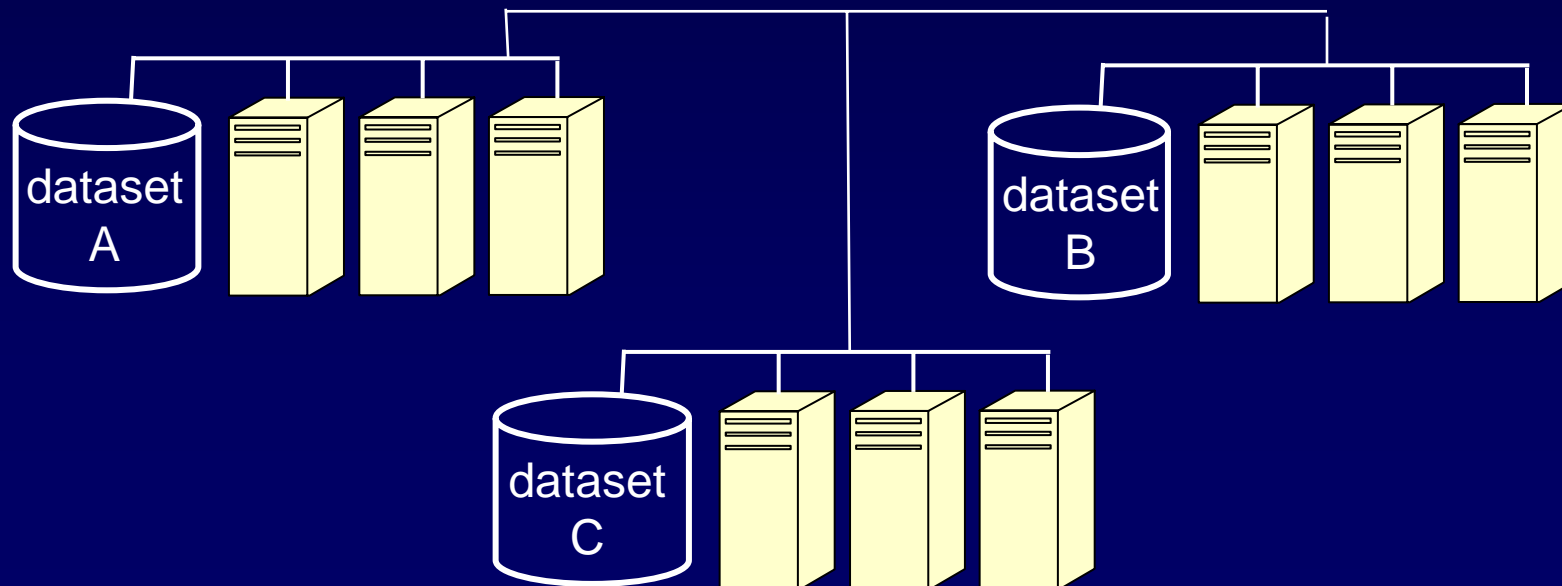
<http://www.swordrd.org/>

David Oppenheimer, Jeannie Albrecht,
David Patterson*, Amin Vahdat
UC San Diego / *UC Berkeley

HPDC '05
July 26, 2005

Application scenario

- Geographically-distributed datasets collected by particle accelerators
- Write a parallel program to analyze each dataset
- Application architecture
 - for each dataset, a group of compute nodes runs the parallel computation on that dataset
 - groups periodically checkpoint results to one another

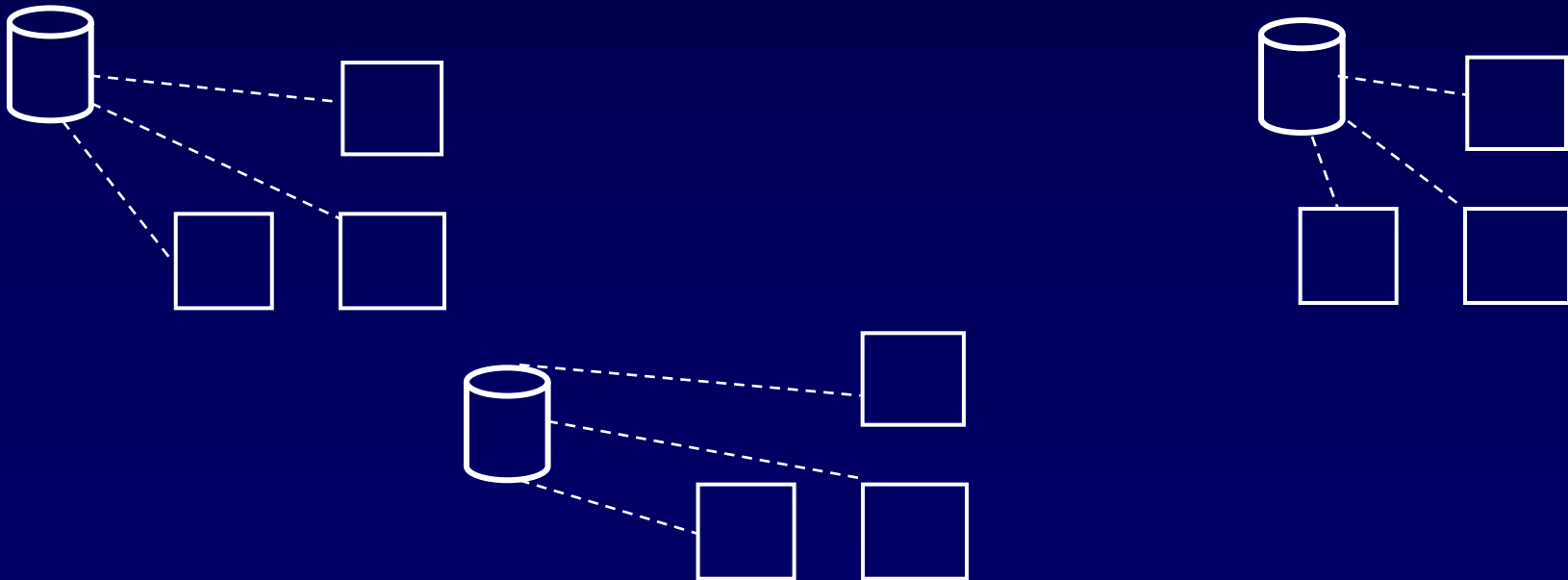


Application deployment environment

- Want to use compute nodes from a shared distributed infrastructure (Grid/PlanetLab)
 - 100s-1000s of nodes and network links
- Not all nodes will meet application's resource needs
 - some nodes too far from data sources
 - compute nodes shared by competing apps → some nodes won't have enough per-node resources
 - network links shared by competing apps → some network links have insufficient b/w, excessive loss
 - some nodes may have historically low availability
- Want to **pick best machines to run application**
- Could do it manually...
- Better: publicly-accessible resource discovery service

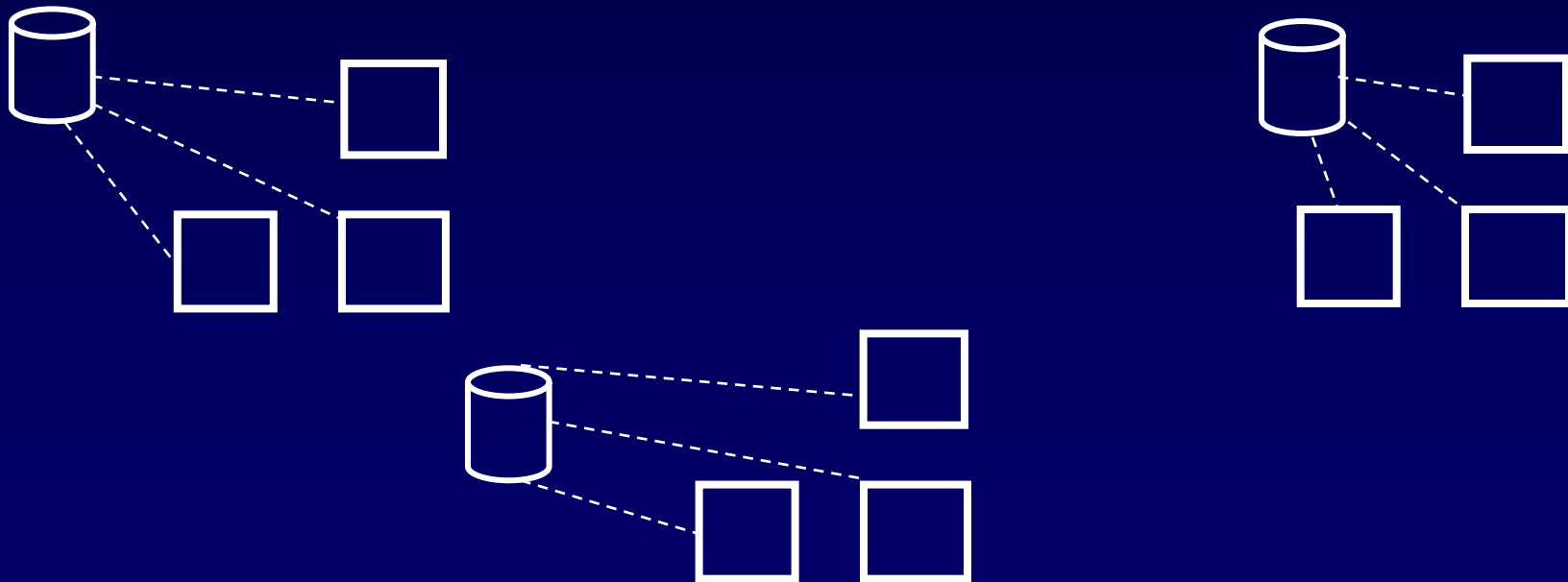
Resource discovery scenario (cont.)

- Resource discovery request
 - 3 3-node groups near specific locations in net topology



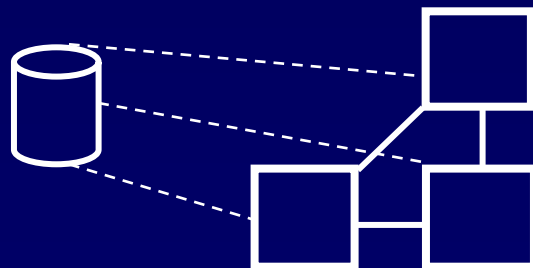
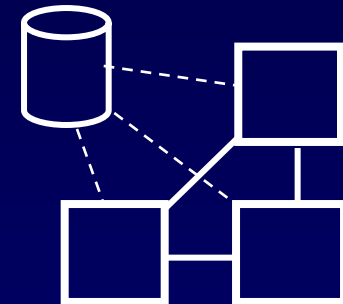
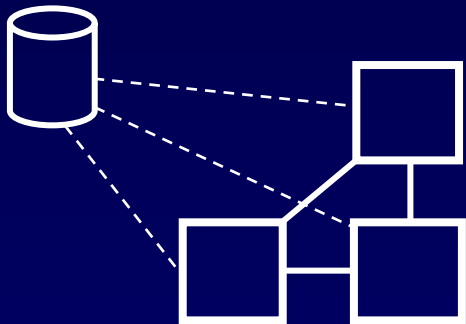
Resource discovery scenario (cont.)

- Resource discovery request
 - 3 3-node groups near specific locations in net topology
 - sufficient computational power



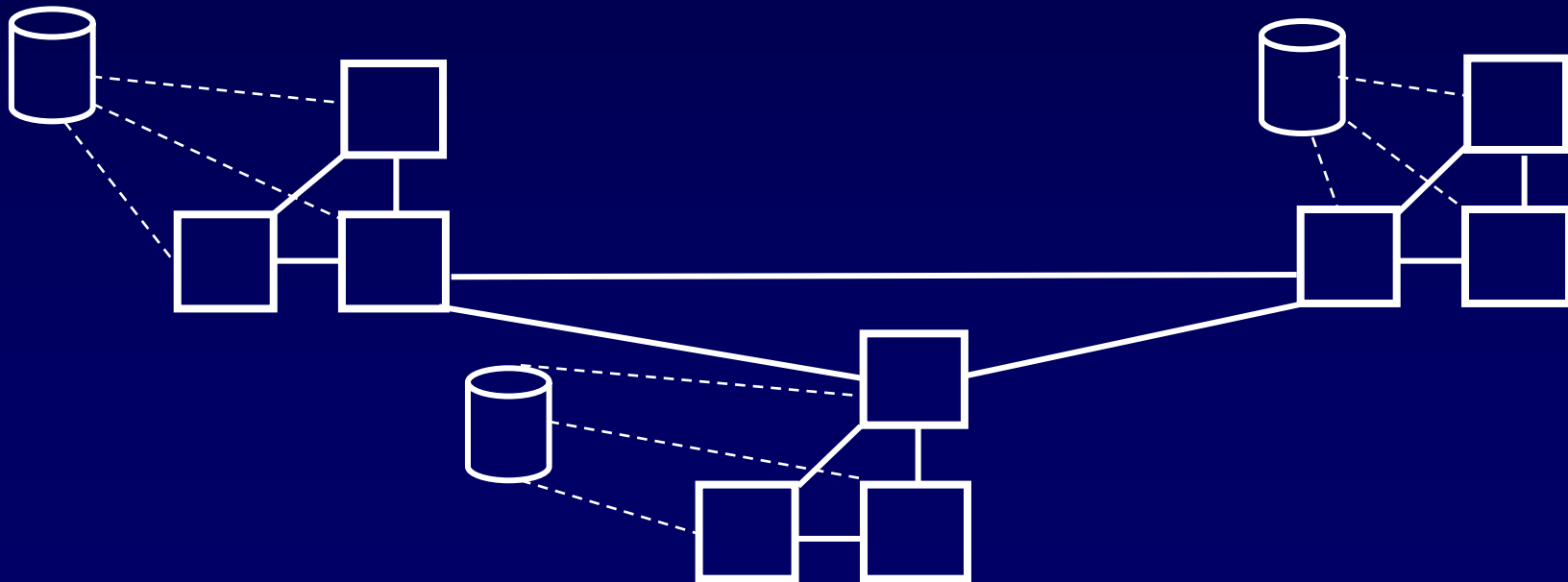
Resource discovery scenario (cont.)

- Resource discovery request
 - 3 3-node groups near specific locations in net topology
 - sufficient computational power
 - low-latency, high-bandwidth links within each group



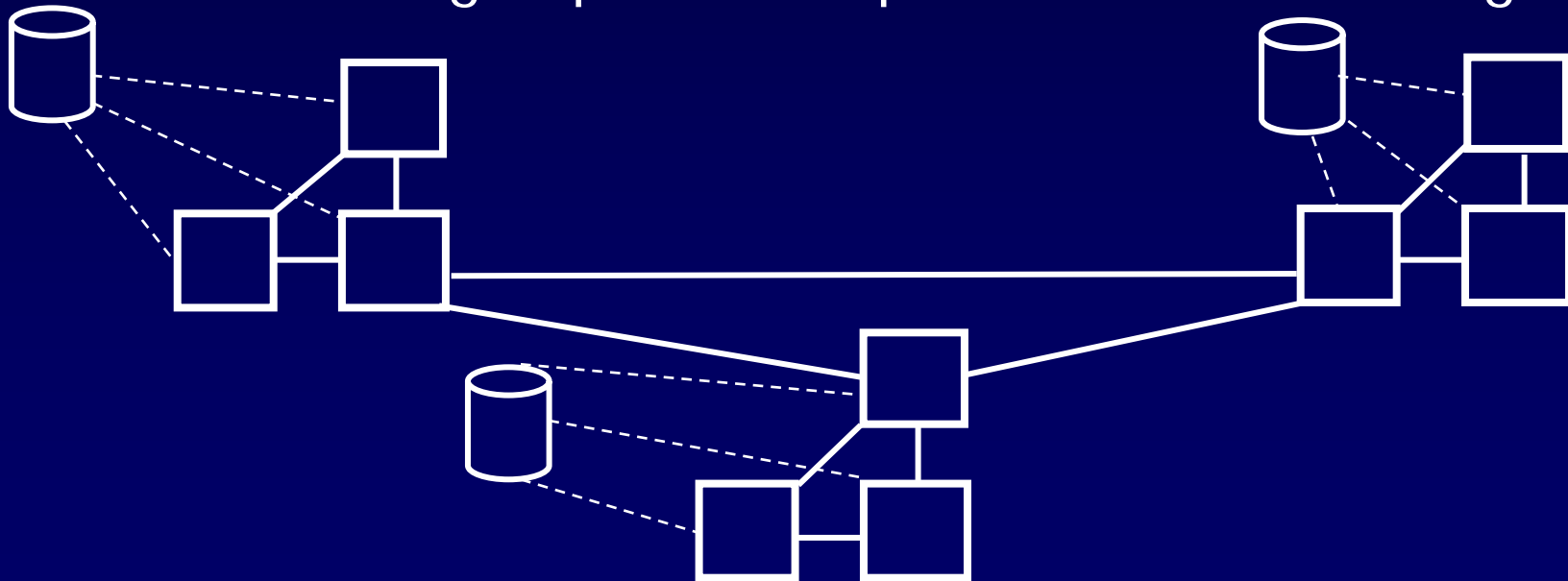
Resource discovery scenario (cont.)

- Resource discovery request
 - 3 3-node groups near specific locations in net topology
 - sufficient computational power
 - low-latency, high-bandwidth links within each group
 - at least one high-bandwidth link connecting each pair of groups



Resource discovery scenario (cont.)

- Resource discovery request
 - 3 3-node groups near specific locations in net topology
 - sufficient computational power
 - low-latency, high-bandwidth links within each group
 - at least one high-bandwidth link connecting each pair of groups
 - b/w within groups more imp't. than b/w between groups



Resource discovery scenario (cont.)

- Requirements for a resource discovery service
 - specify per-node & inter-node (net) properties
 - specify app sensitivity to resource constraints
 - continuously re-evaluate node selection
 - scale to large numbers of nodes and high update rates
 - robustly handle failure and recovery
 - impose minimal management burden (federated platform)
- No existing system met all requirements
- Built and deployed SWORD
 - publicly-accessible resource discovery service

SWORD on PlanetLab

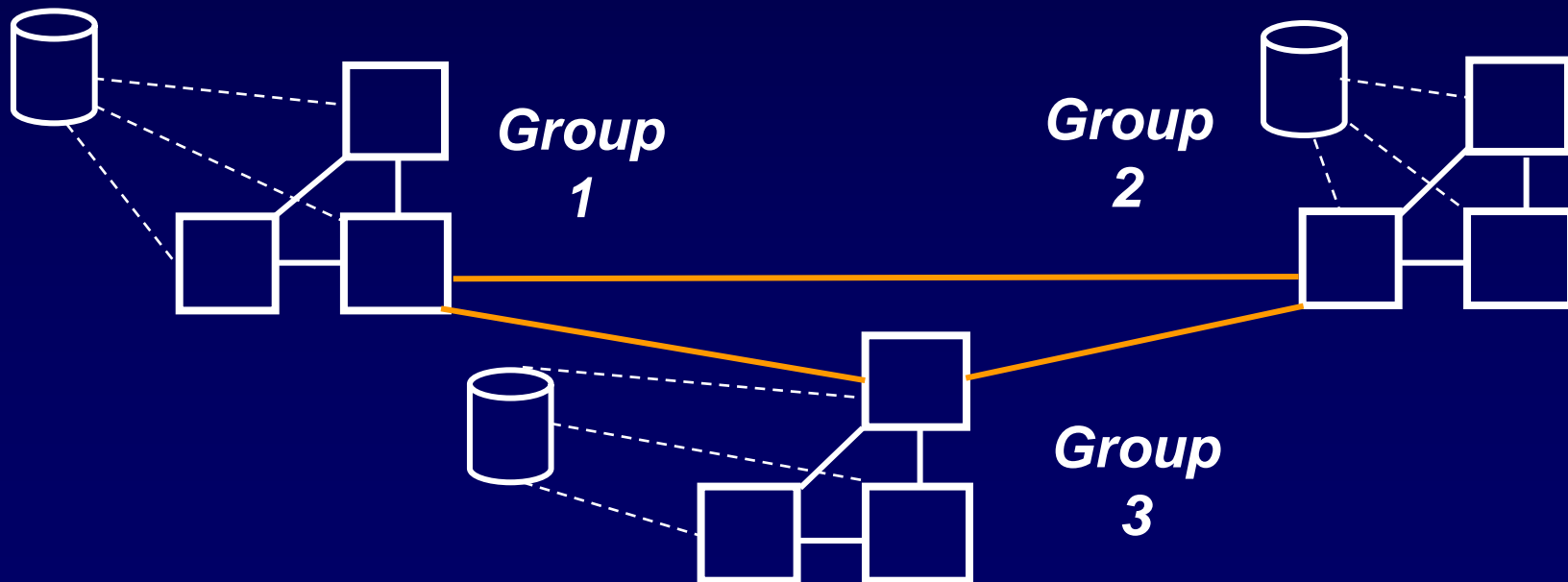
- A service running continuously on PlanetLab for a year
 - <http://www.swordrd.org/>
 - 200+ nodes
 - extensible set of measurements sent every 2 min.
 - **Ganglia** host measurements
 - **Trumpet** end-to-end host tests
 - **slicestat** information via CoTop
 - **Vivaldi** network coordinates
 - two PlanetLab services build on top of SWORD
 - Bellagio (micro-economic resource allocation system)
 - PLuSH (execution management system)
- Also, a research vehicle

Outline

- Resource discovery scenario and requirements
- Expressing queries
- Answering queries
 - distributed query processor alternatives:
fixed servers, Distributed Hash Table (DHT)-based, hybrid
- Evaluation
- Conclusion

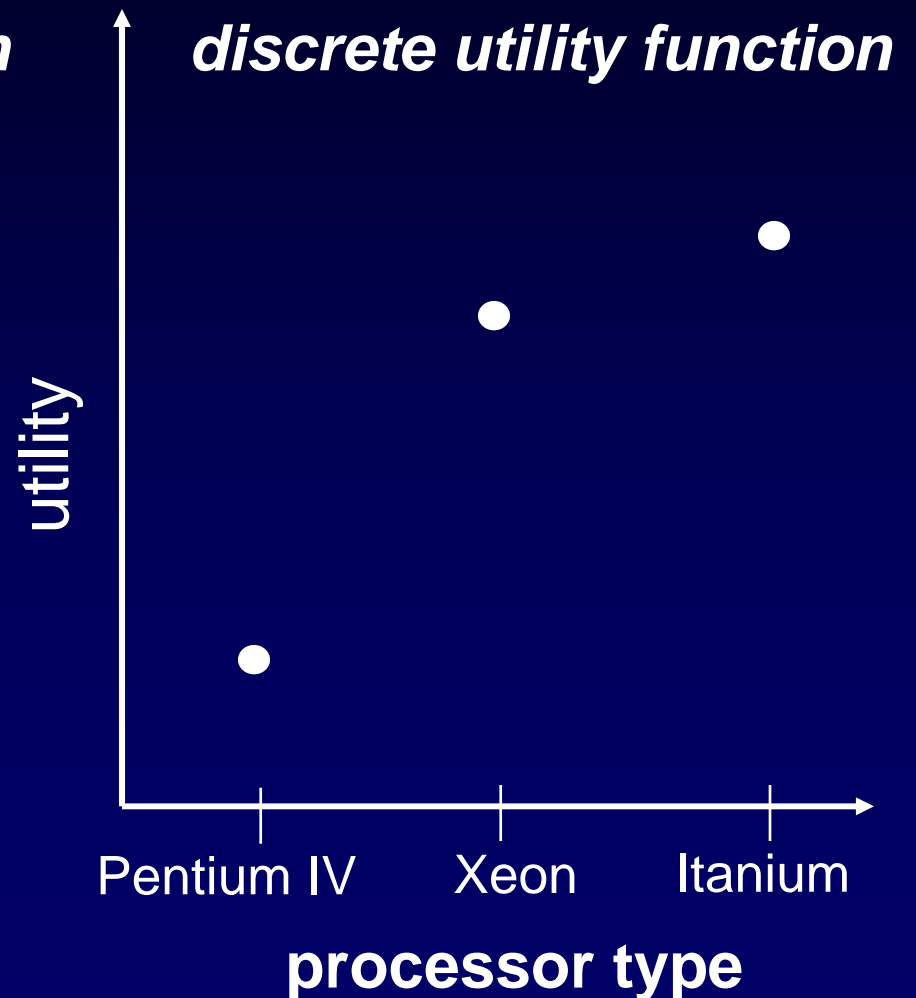
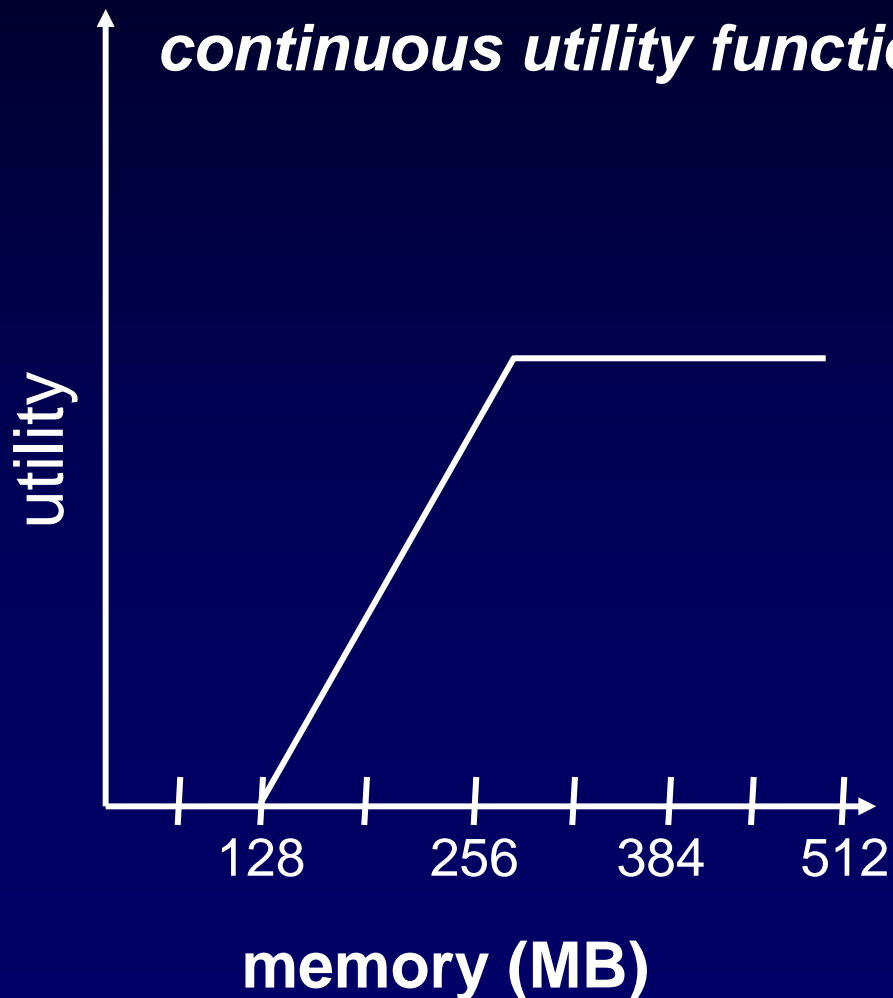
Expressing queries

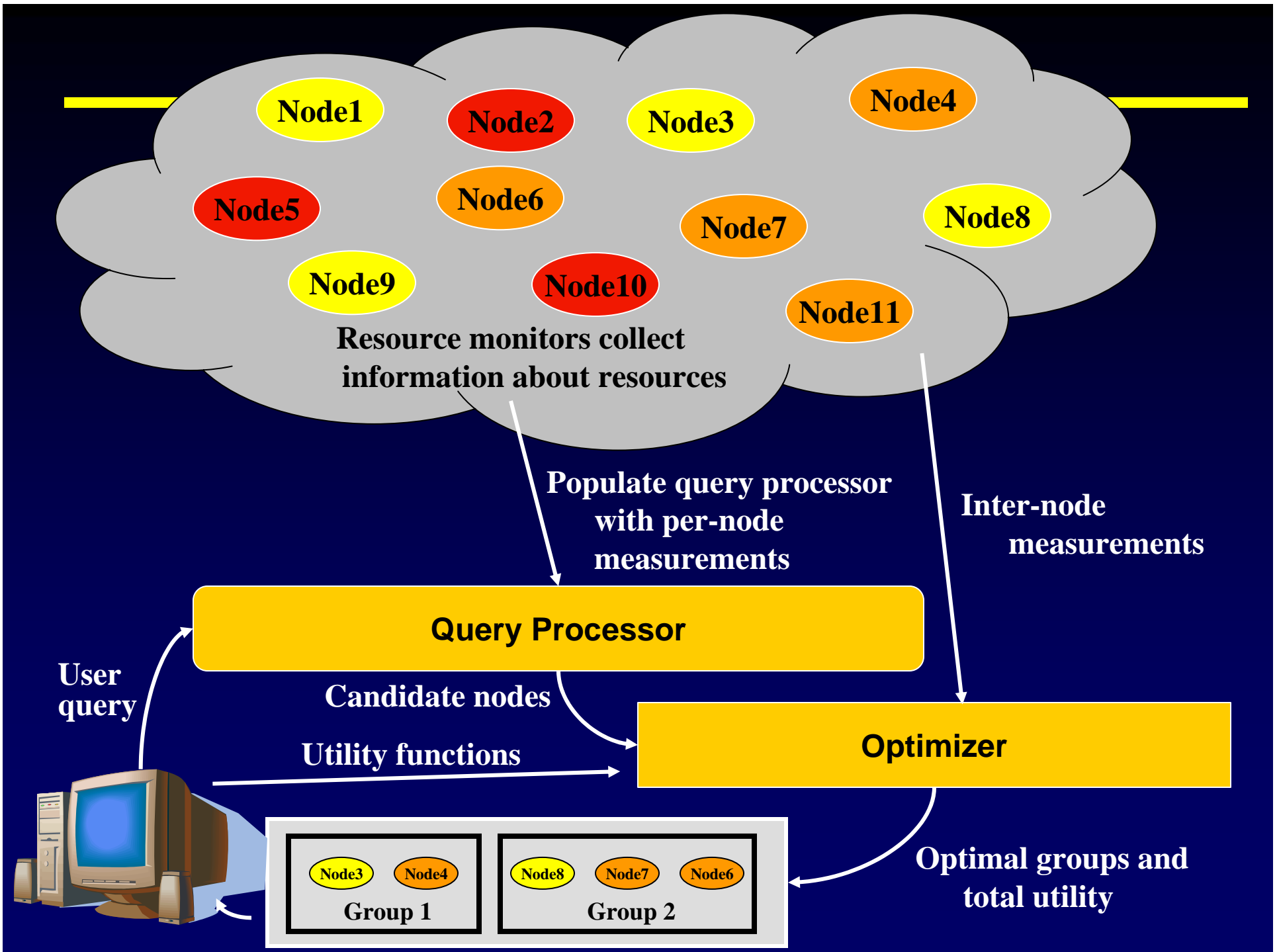
- Groups: equivalence class of nodes
 - per-node properties (load, OS, net. pos., ...)
 - inter-node properties (latency, b/w, ...)
- **Inter-group properties (latency, b/w, ...)**
- Utility function associated with each property



Utility functions

- SWORD uses query's utility functions to find maximum-utility mapping of available nodes to groups



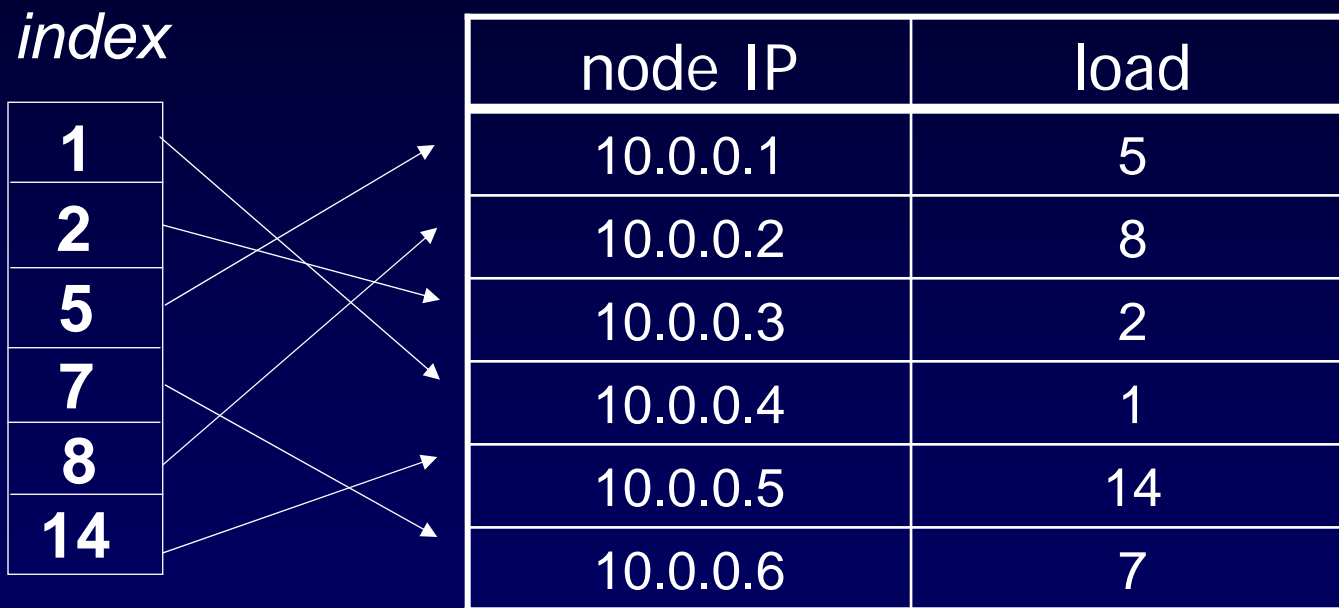


Outline

- Resource discovery scenario and requirements
- Expressing queries
- Answering queries
 - distributed query processor alternatives:
fixed servers, Distributed Hash Table (DHT)-based, hybrid
- Evaluation
- Conclusion

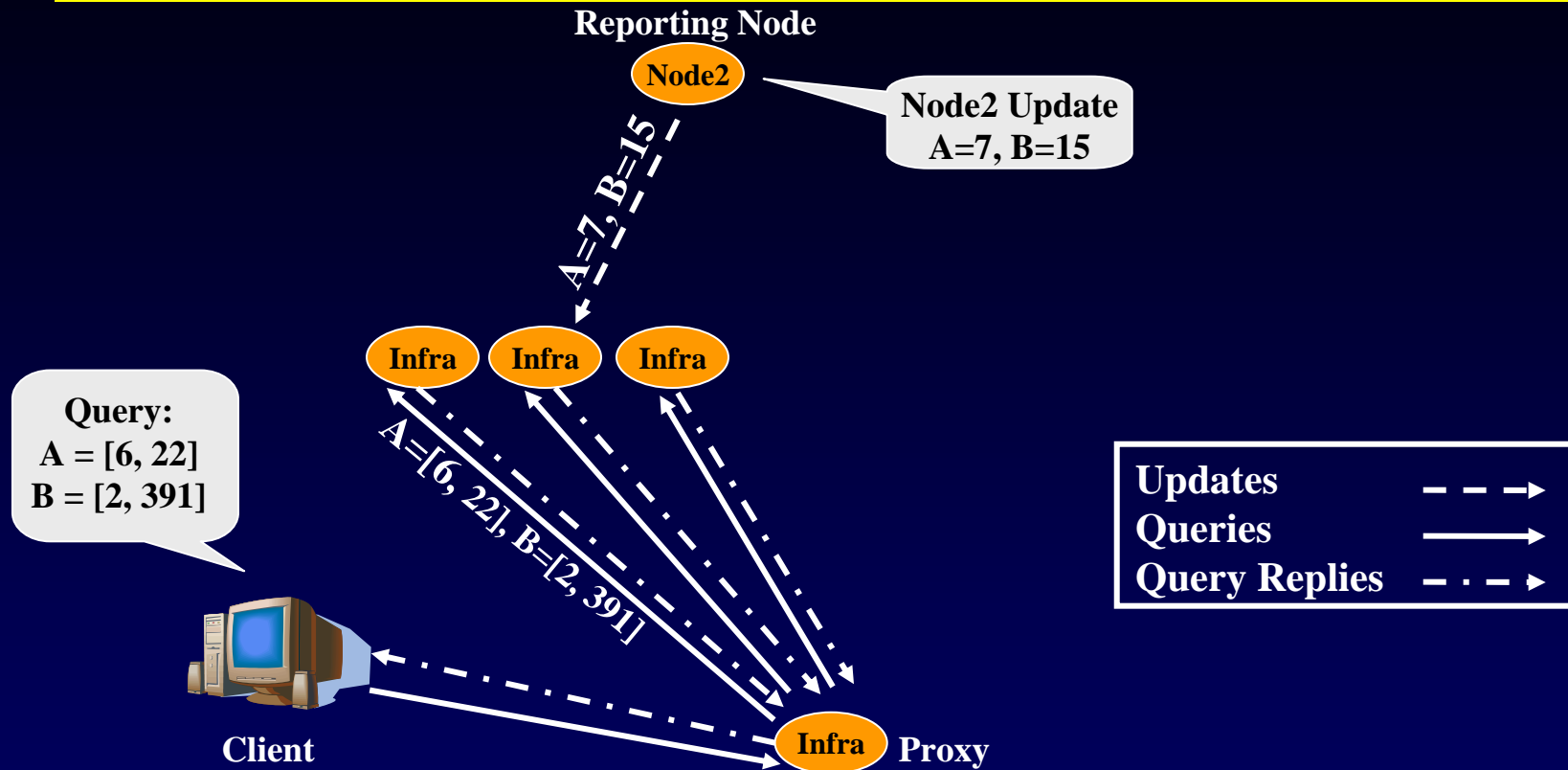
Query processor

- Logical database of periodically-updated resource measurements from each node + methods to query
- Core query method: multi-attribute range search



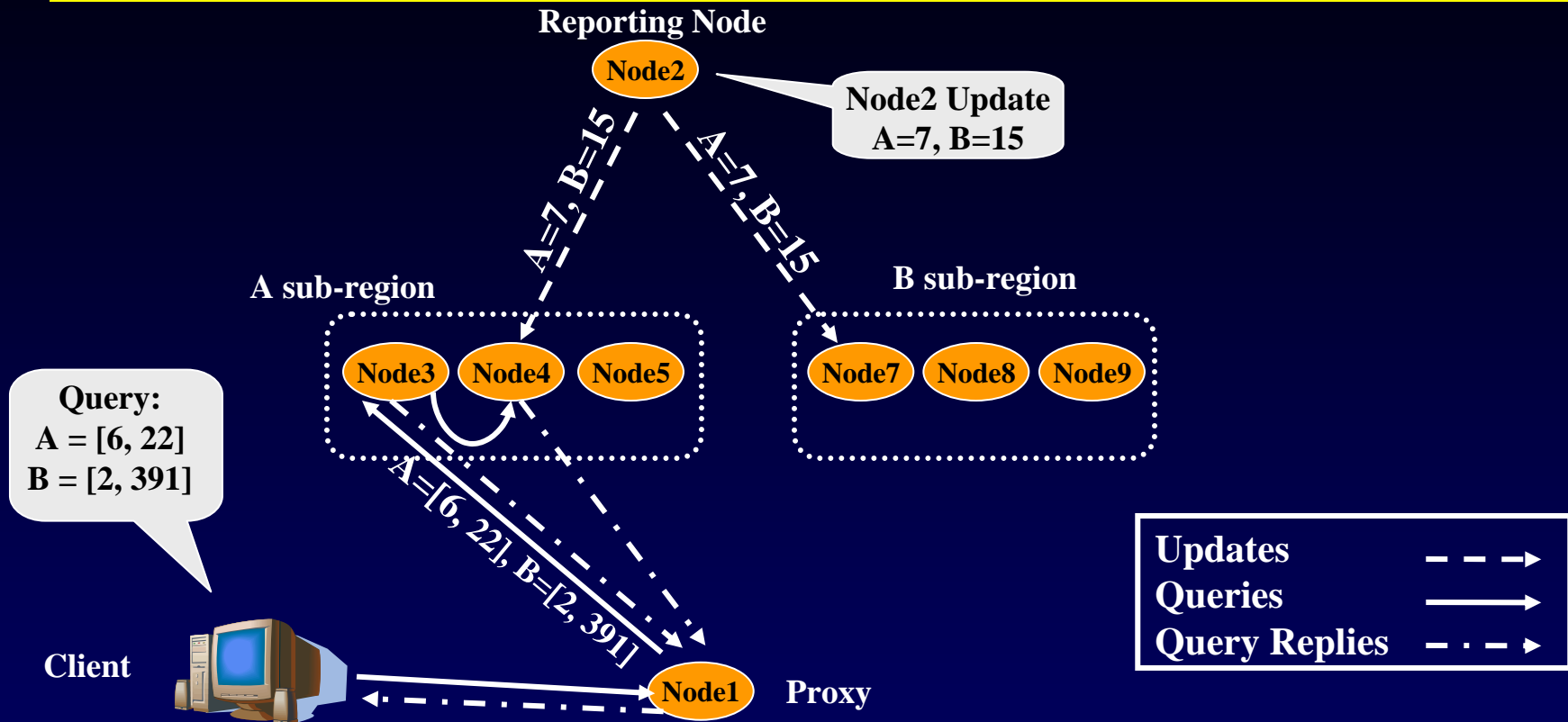
- Support large #s of nodes, high update rates, low-latency queries, high availability → **distributed q.p.**
- Alternatives: fixed servers, DHT-based, hybrid

Fixed servers approach



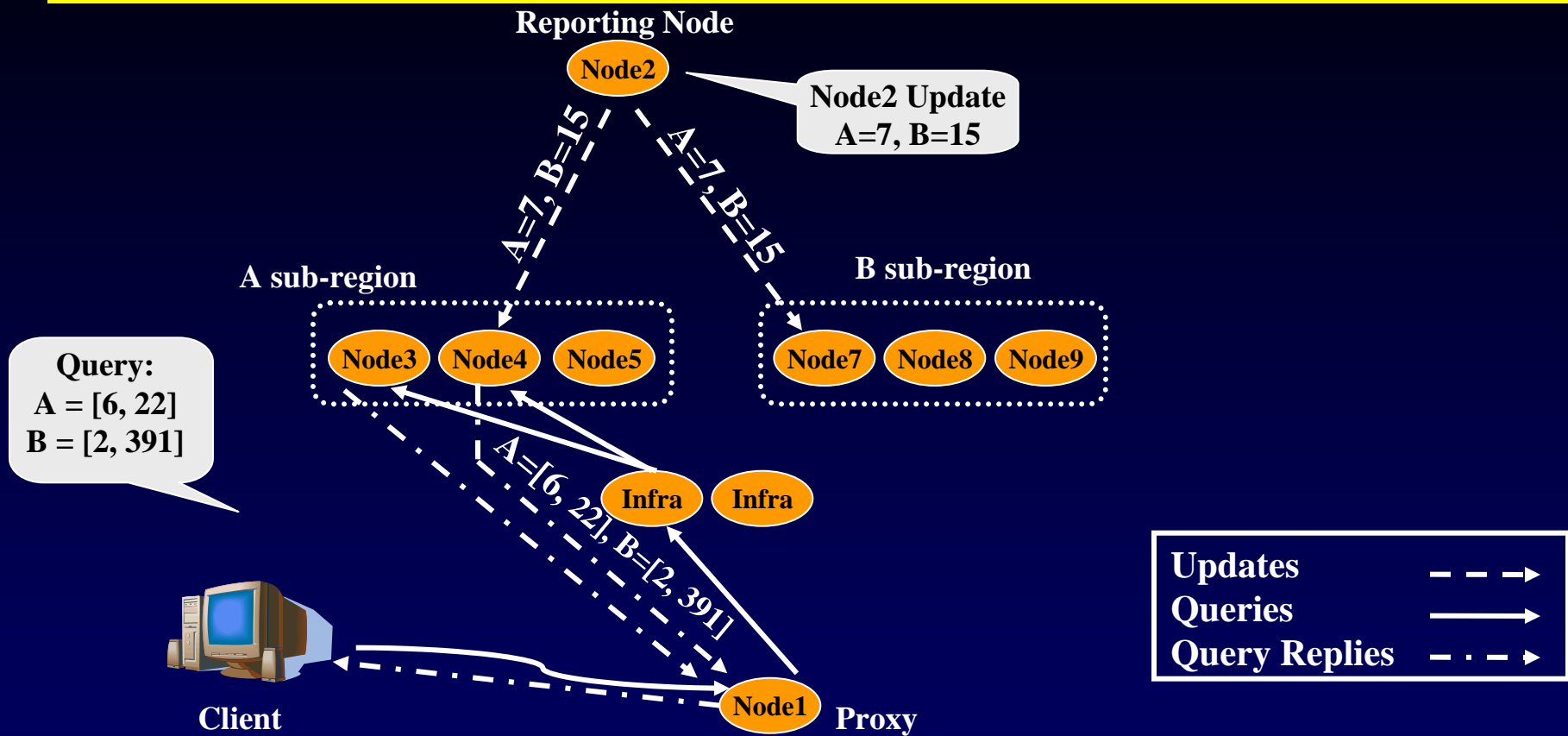
- Service provider with fixed datacenter infrastructure
- **Advantages:** simple, low-latency
- **Disadvantages:** no automatic load balancing, failover, or self-configuration; concentrates bandwidth usage at a few sites (performance and \$\$\$ bottleneck)

DHT approach 1: SingleQuery



- Use a DHT to map $\langle \text{attribute, value} \rangle \rightarrow$ server node
 - SWORD adds efficient multi-attribute range queries and load balancing
- **Advantages:** failover and self-config; load-balancing; leverage existing resources; organic scaling; spread b/w use
- **Disadvantage:** increased latency b/c queries take multiple hops

Index (hybrid) approach



- Service provider only maps $\langle \text{attribute}, \text{value} \rangle$ to node
 - updates still routed through DHT
- **Advantage:** all queries answered in 5 network hops
- **Disadvantage:** need a scalable, robust index service

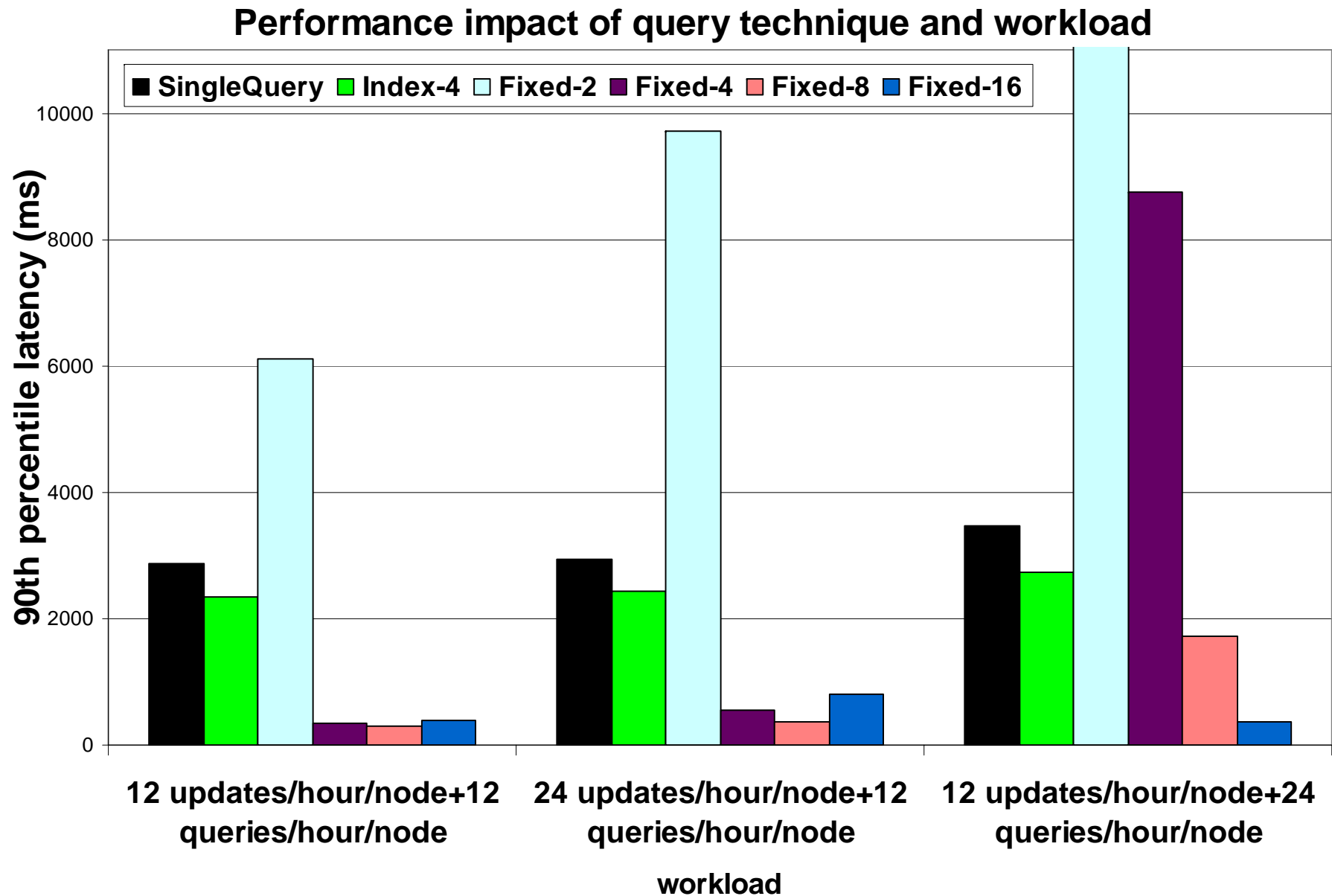
Evaluation questions

1. How do query processor architecture and workload intensity affect performance?
2. How does network bandwidth consumption compare among the approaches?
3. How much does use of “representatives” reduce query latency?
4. How do optimizer heuristics impact optimizer performance and accuracy?
5. How does the system perform on PlanetLab?

Evaluation environment and configurations

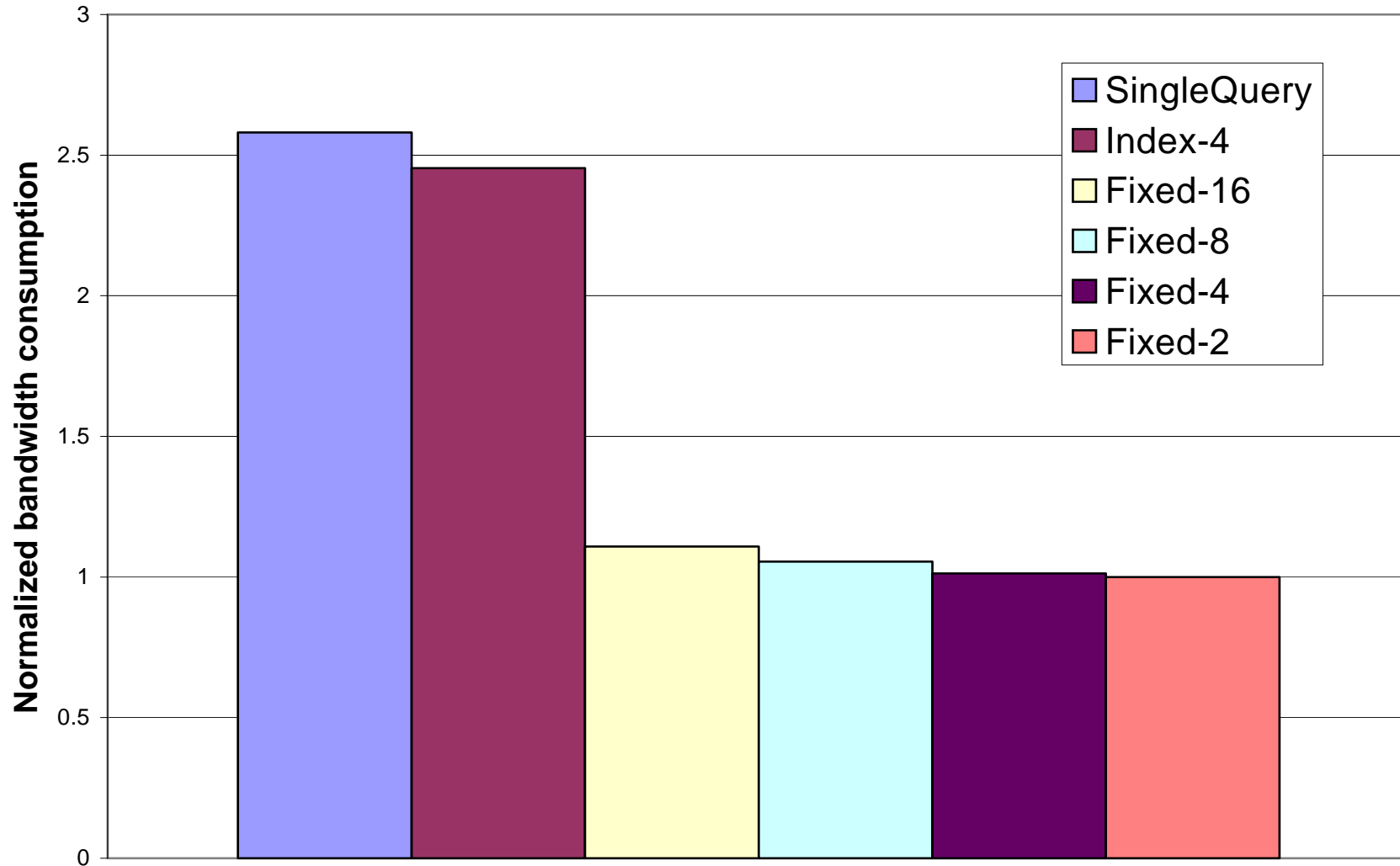
- Cluster of 40 PCs running ModelNet
- INET transit-stub topology, 1000 virtual nodes
- Updates
 - 32 metrics taken from PlanetLab trace + Vivaldi network coords.
 - 3-24 updates/node/hour
- Queries
 - 5 per-node attrs (load, free disk, free mem, bytes sent/rcvd), Zipf
 - 1 inter-node attr (inter-node latency)
 - returned median 120 nodes, 90th percentile 160 nodes
- SingleQuery, Index-4, Fixed-{2,4,8,16}
 - Bamboo DHT
 - DHT nodes 384 Kb/s, 4 infrastructure nodes 155 Mb/s

Latency: SQ vs. Index vs. Fixed



Aggregate bandwidth consumption

Normalized aggregate bandwidth consumption



Related work

- Resource discovery
 - vgDL/vgFAB (CCGrid '05)
 - XenoSearch (HPDC '03)
 - ClassAds, gang matching, set matching, Redline, R-GMA, RGIS
 - MDS- $\{2,3,4\}$
 - Network-Sensitive Service Discovery
- Internet-scale query processors
 - hierarchical: IrisNet, Astrolabe
 - flat: PIER, Sophia
- Range search: Karger and Ruhl, Mercury, PHT, RST
- Many monitoring data sources/aggregators

Conclusion

- SWORD: a publicly-accessible resource discovery svc.
 - specify per-node & inter-node properties
 - specify app sensitivity to resource constraints
 - continuously re-evaluate node selection
 - robust, scalable, minimal management
- For today's workloads and platform sizes, small number of well-connected fixed infrastructure server clusters provides best performance
- Building on top of DHT → leverage existing resources
 - and provide organic scalability, robustness, self-config.
 - with moderate performance impact
- If future platforms contain orders of magnitude more resources, financial benefit of spreading bandwidth usage among all participants may dominate

Design and Implementation Tradeoffs for Wide-Area Resource Discovery

<http://www.swordrd.org/>

David Oppenheimer, Jeannie Albrecht,
David Patterson*, Amin Vahdat
UC San Diego / *UC Berkeley

HPDC '05
July 26, 2005