Nebulas: Using Distributed Voluntary Resources to Build Clouds

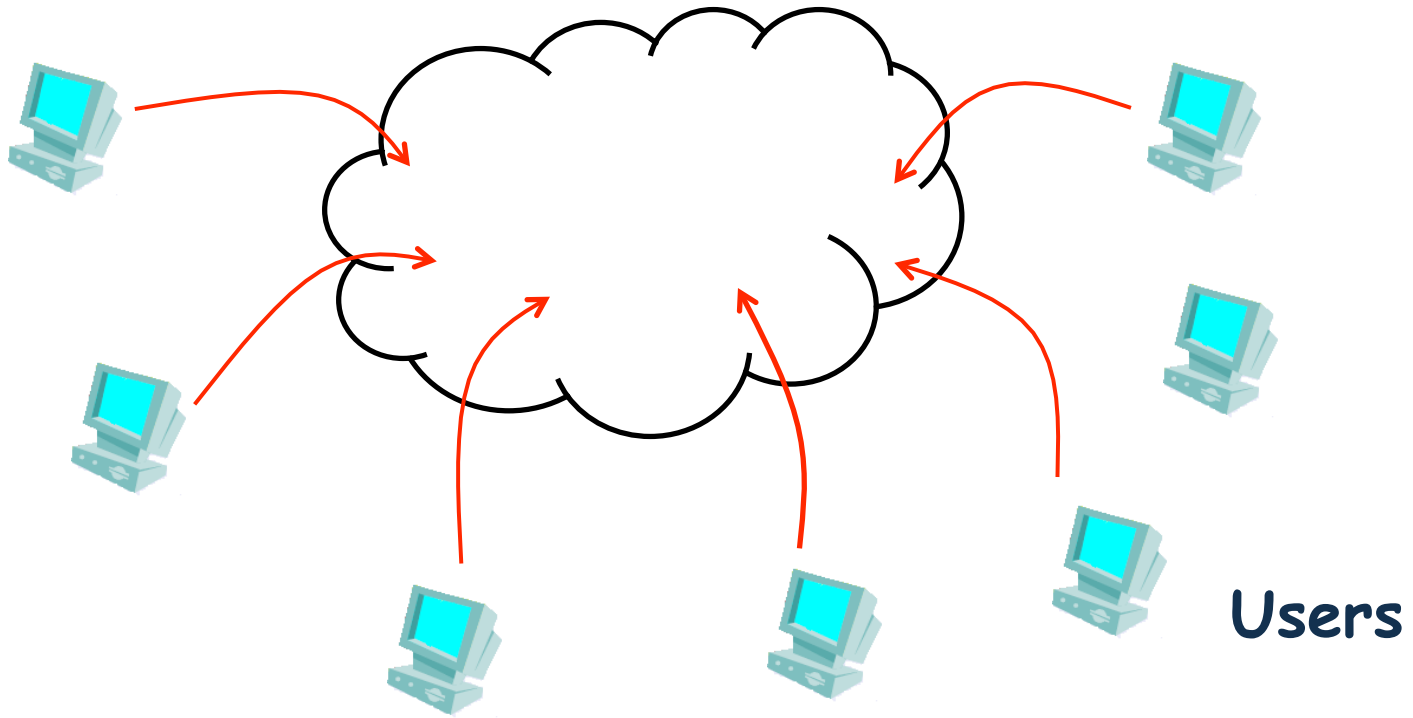
Abhishek Chandra and Jon Weissman

Department of Computer Science

University of Minnesota

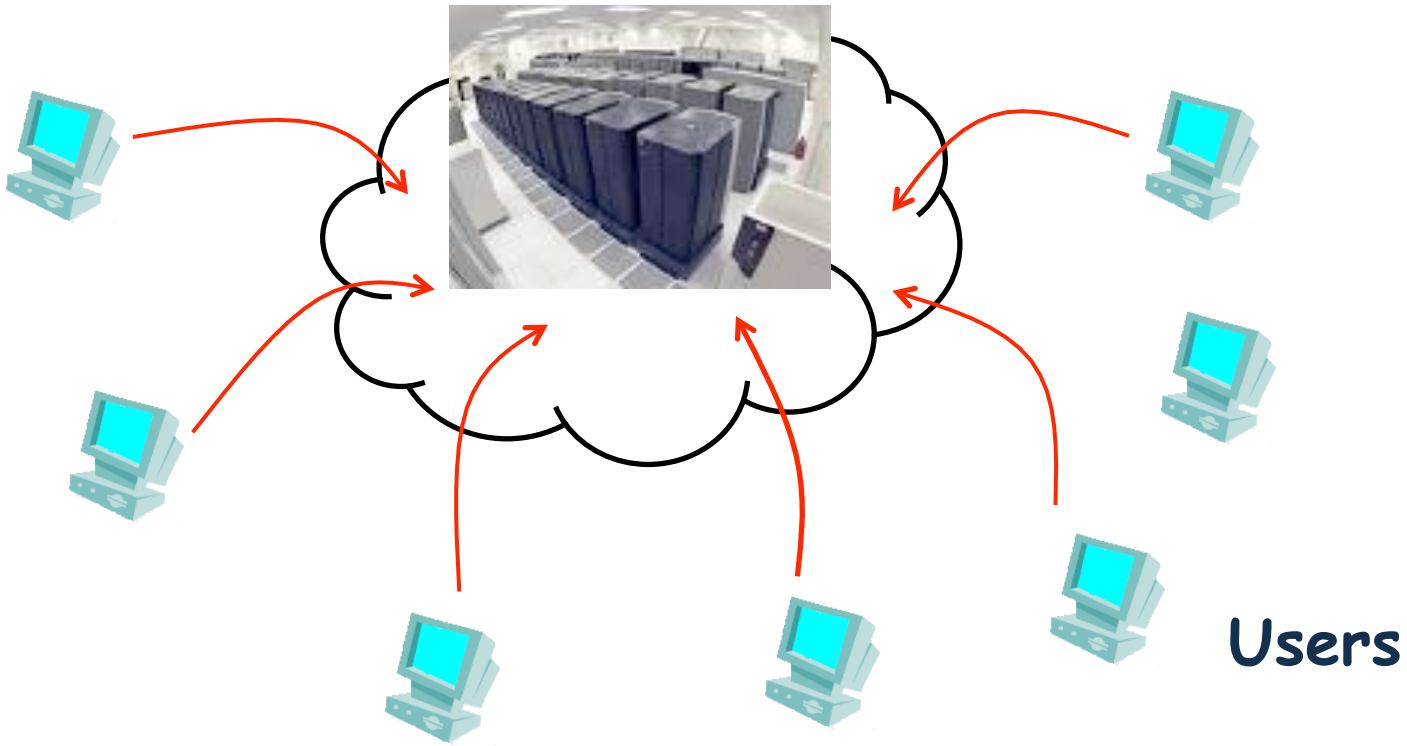
Clouds

- Cloud: Hides details of actual service deployment from users



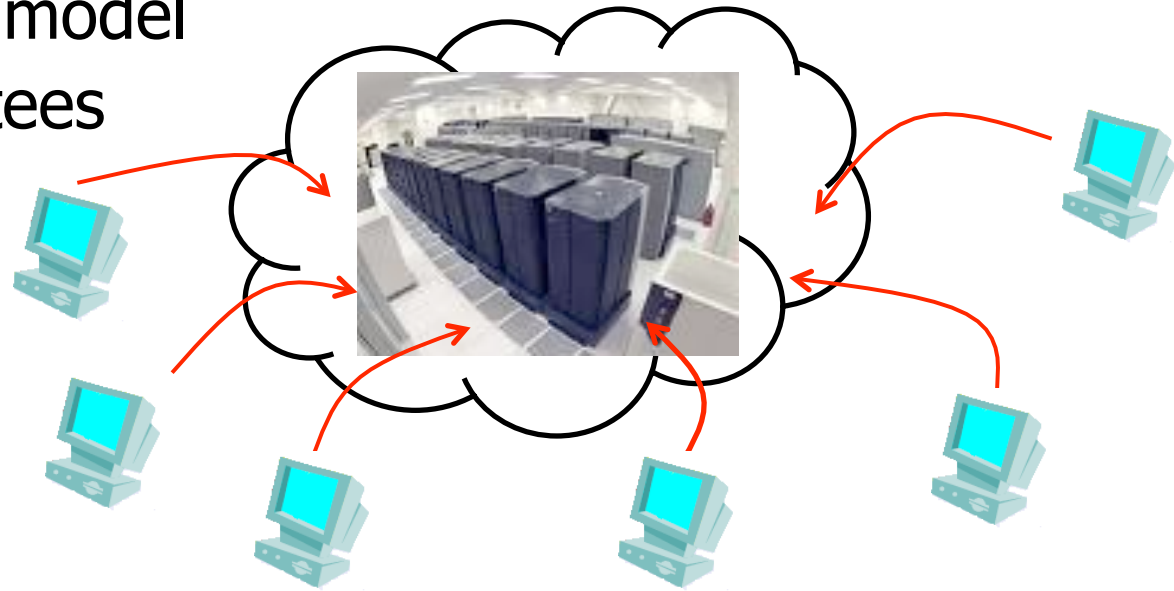
Current Cloud Model

- Cloud: Hides details of actual service deployment from users



Current Cloud Model

- Largely centralized (or small degree of distribution)
- Pay-as-you-go model
- Strong guarantees



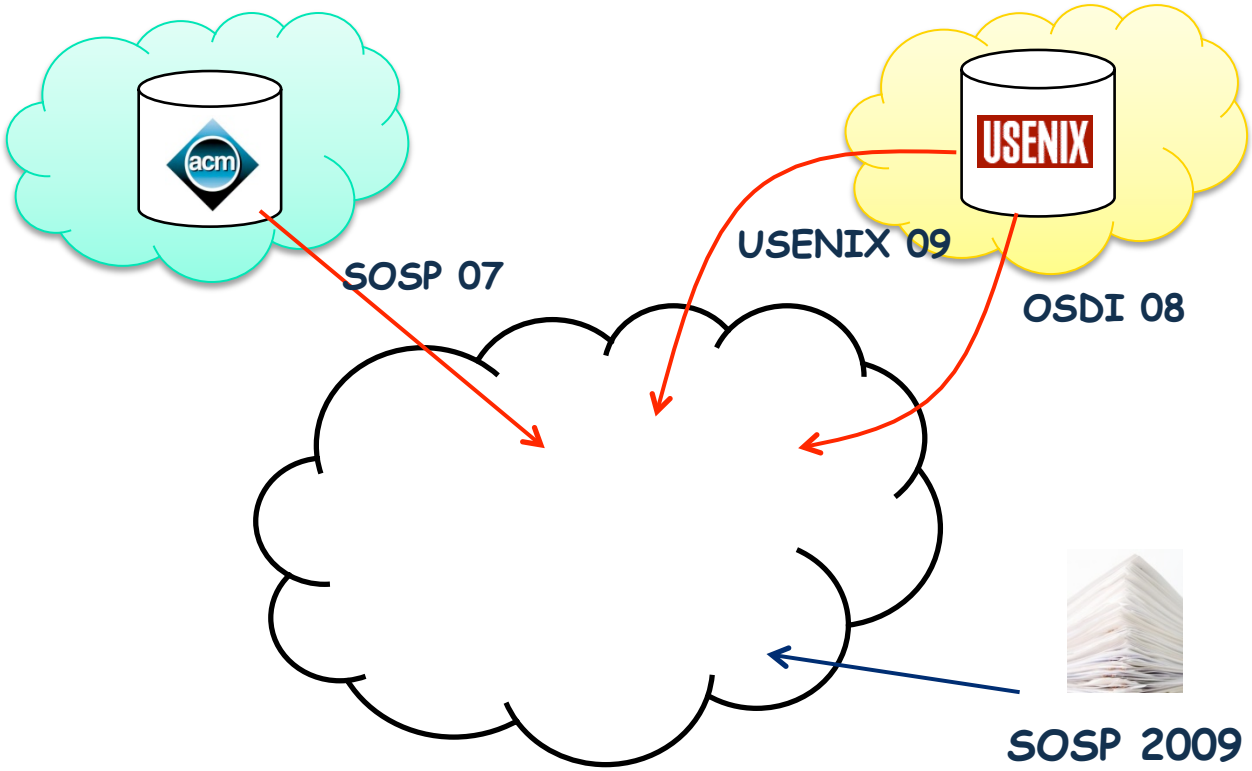
- **Question:** Are there services that do not need/ fit this cloud model?

Class 1: "Experimental" Services

- Experimental deployment for:
  - Debugging, viability, requirement estimation

Class 1: "Experimental" Services

- Experimental deployment for:
  - Debugging, viability, requirement estimation

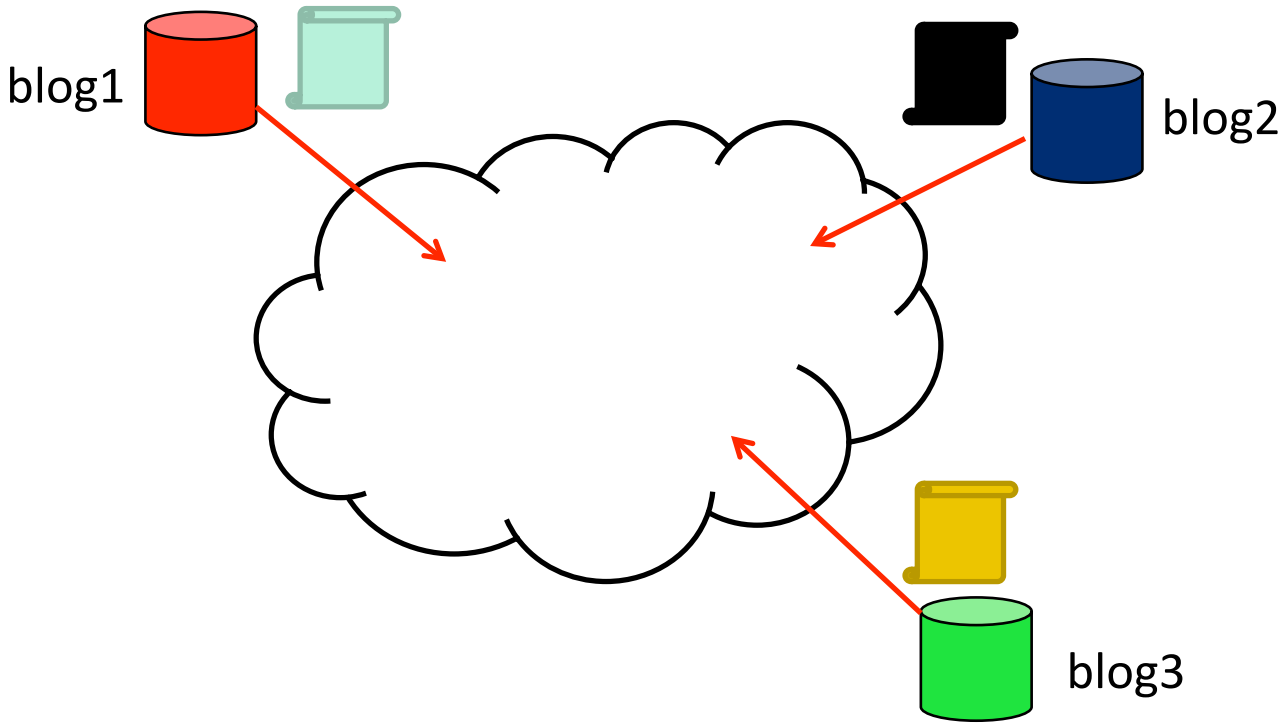


Class 2: Dispersed-Data-Intensive Services

- Data is geographically distributed
  - Costly, inefficient to move to central location

Class 2: Dispersed-Data-Intensive Services

- Data is geographically distributed
  - Costly, inefficient to move to central location

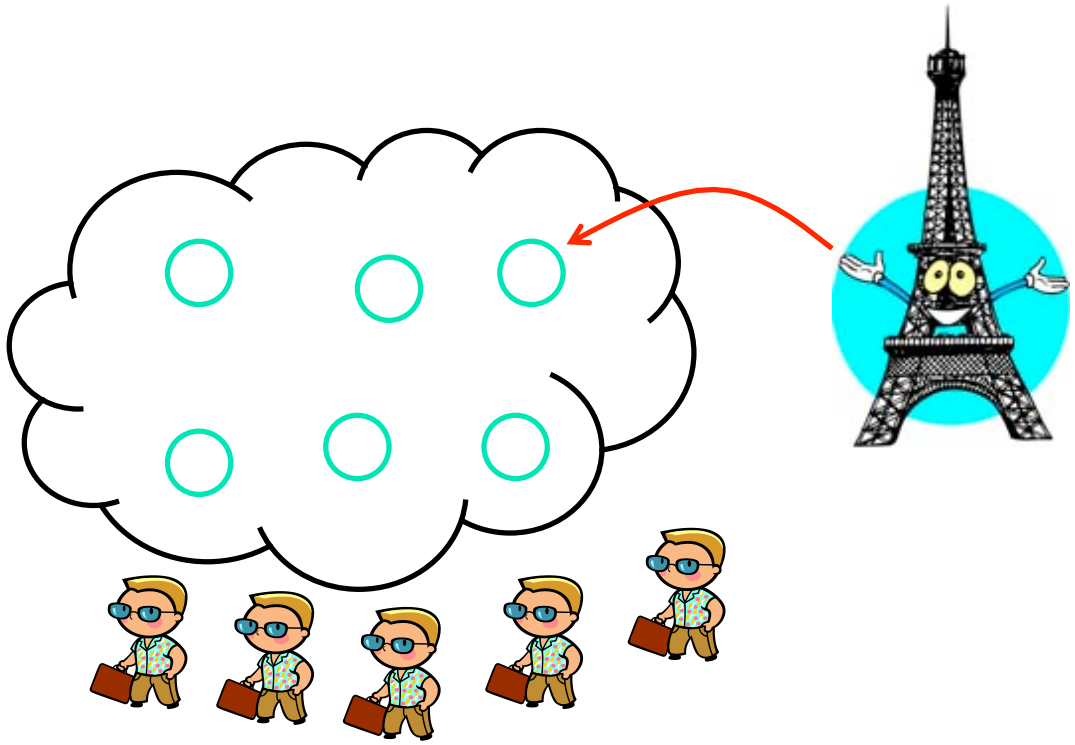


Class 3: Shared "Public" Services

- Personal application offered as free service
  - User-demand driven, scale-up/scale-down needed

Class 3: Shared "Public" Services

- Personal application offered as free service
  - User-demand driven, scale-up/scale-down needed

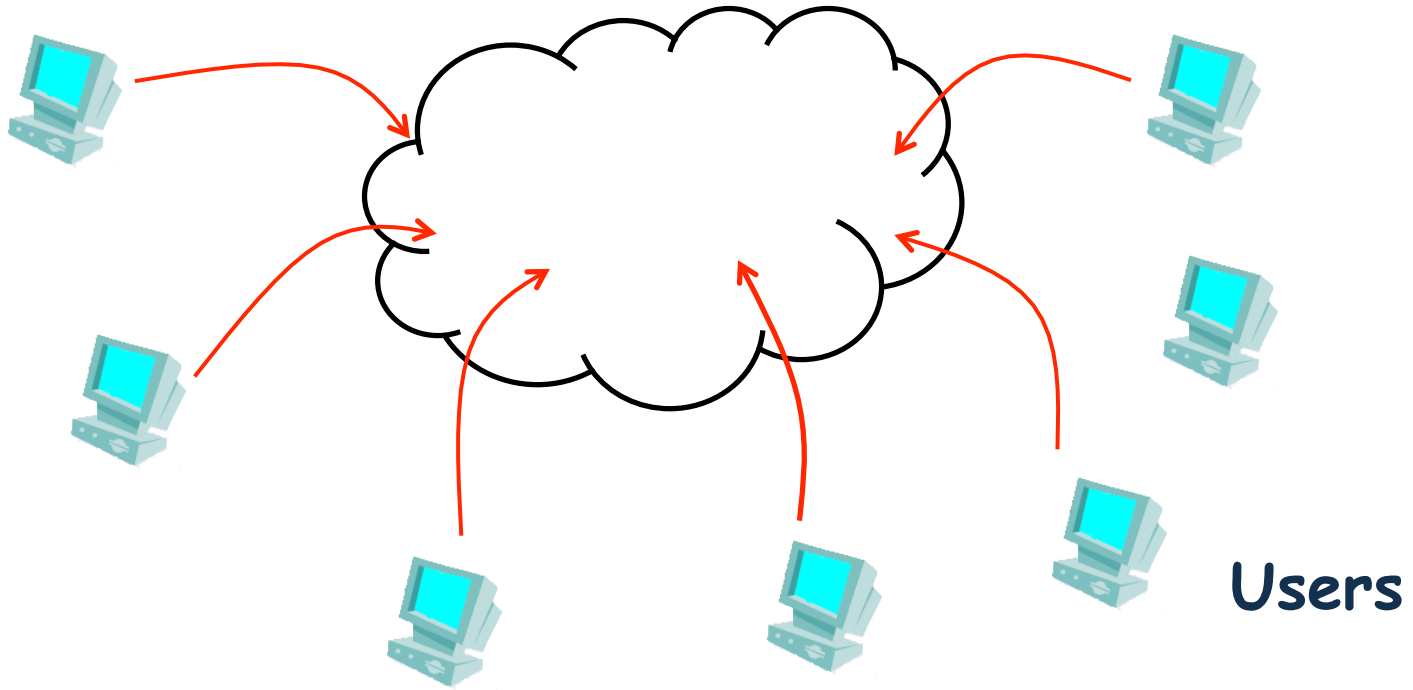


Common Service Characteristics

- Elastic resource consumption
  - Scale up/down based on demand
- Geographical data/user distribution
  - Execution dependent on location of data/user
- Low/no cost
  - Do not want to pay for resources
- Weak performance/robustness requirements
  - Some failures may be ok

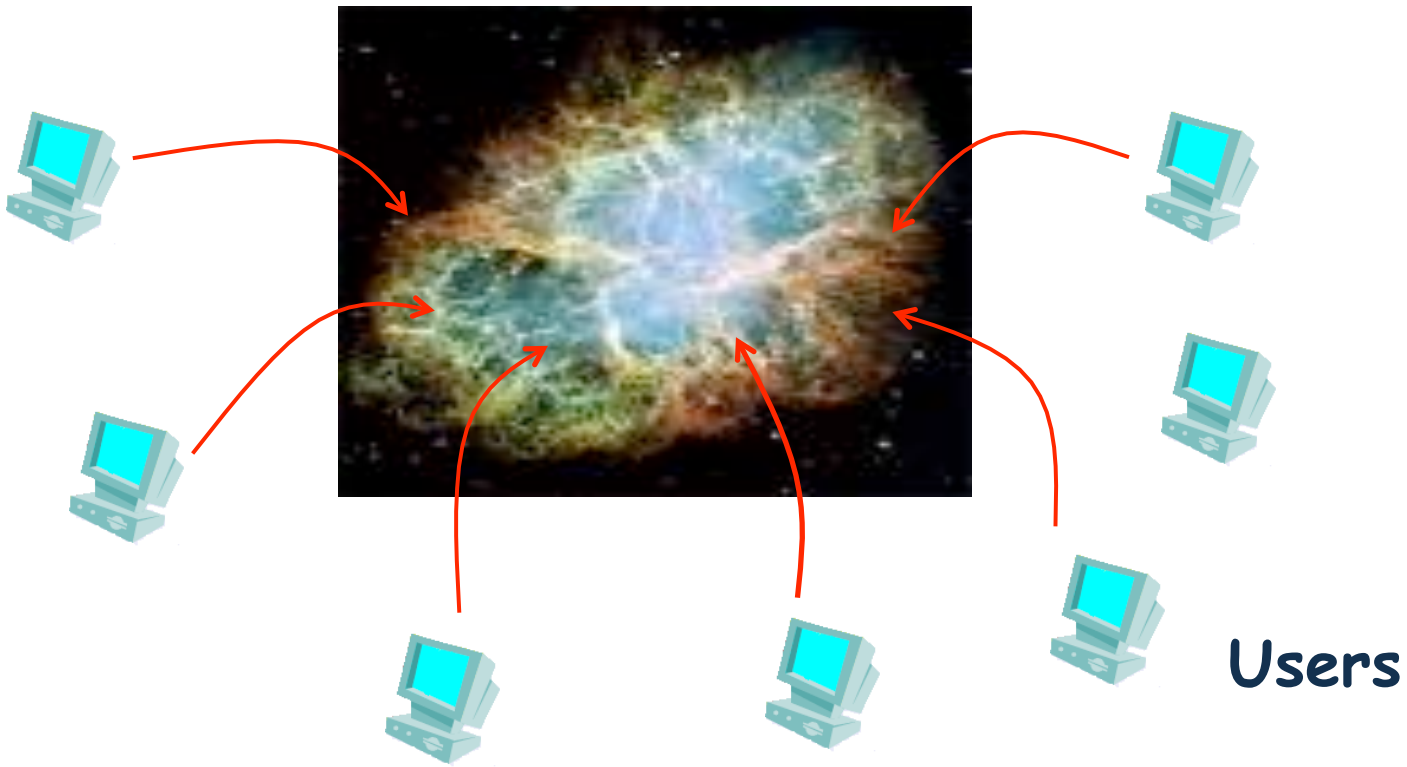
Cloud

- Cloud: Hides details of actual service deployment from users



Nebula

- Decentralized, less-managed cloud
  - Dispersed storage/compute resources
  - No/low user cost



Building Nebulas

- **Idea:** Use distributed voluntary resources
  - Resources donated by end-users
  - ala @home, P2P systems

Why Voluntary Resources?

- Scalability: Large number of resources available
  - SETI@Home: Over 2.2 million computers contributing ~510 TFlops of compute power
  - Kazaa: Over 3.5 million users
- Low cost:
  - Minimal deployment, management costs
  - [Kondo09]: 2 orders of magnitude difference in EC2 vs. SETI@home resources/$
- Dispersion: Geographically distributed
  - Users can be located worldwide

How is Nebula different from @home?

- Cloud-oriented services impose new requirements

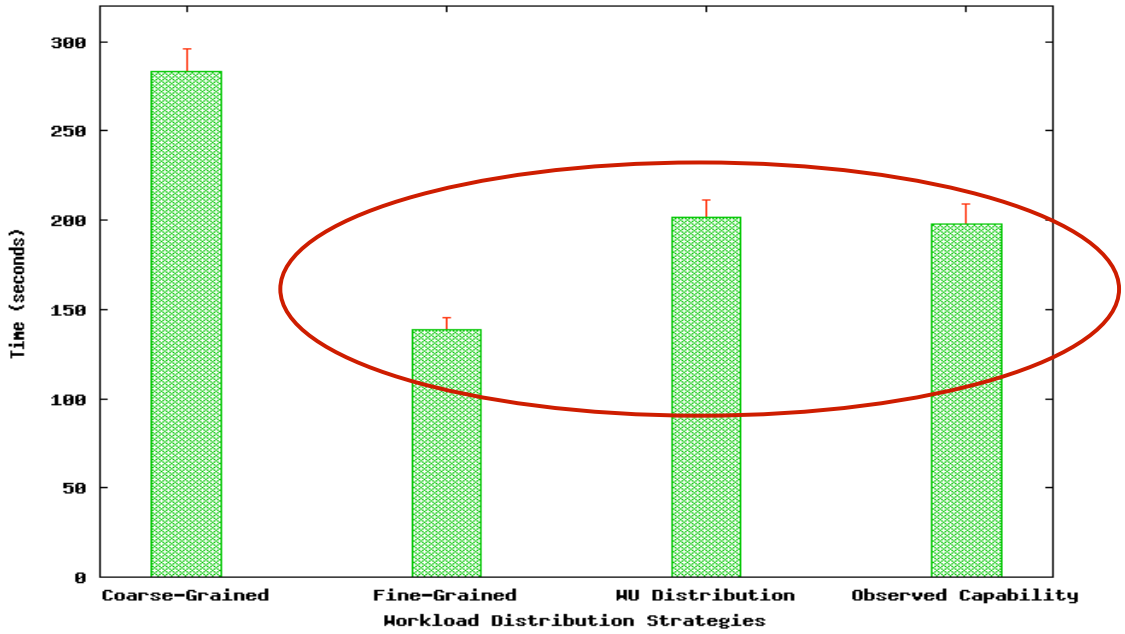
Requirement	Nebula	@home
Collective performance	High	None
Locality/Context-awareness	High	Low
Statefulness	High/medium	Low

Challenges

- Heterogeneity
  - Different nodes have different CPU speeds, network bandwidth, loads
- Resource dispersion
  - Data sources and compute resources may be widely distributed
- Unreliability
  - Node/link failures, high churn

Handling Heterogeneity

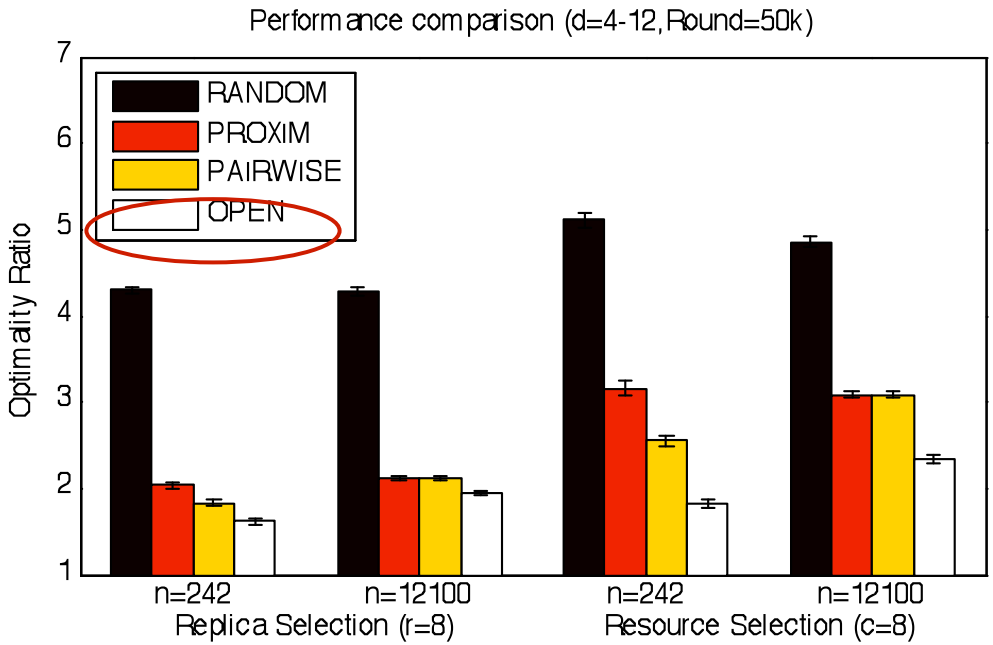
- Heterogeneity-aware resource selection and allocation
  - Allows better collective performance
- Trivedi et al. [IJHPCA06]: Fit tasks to node capability



Heterogeneity-aware allocation reduces execution time

Handling Data Dependence

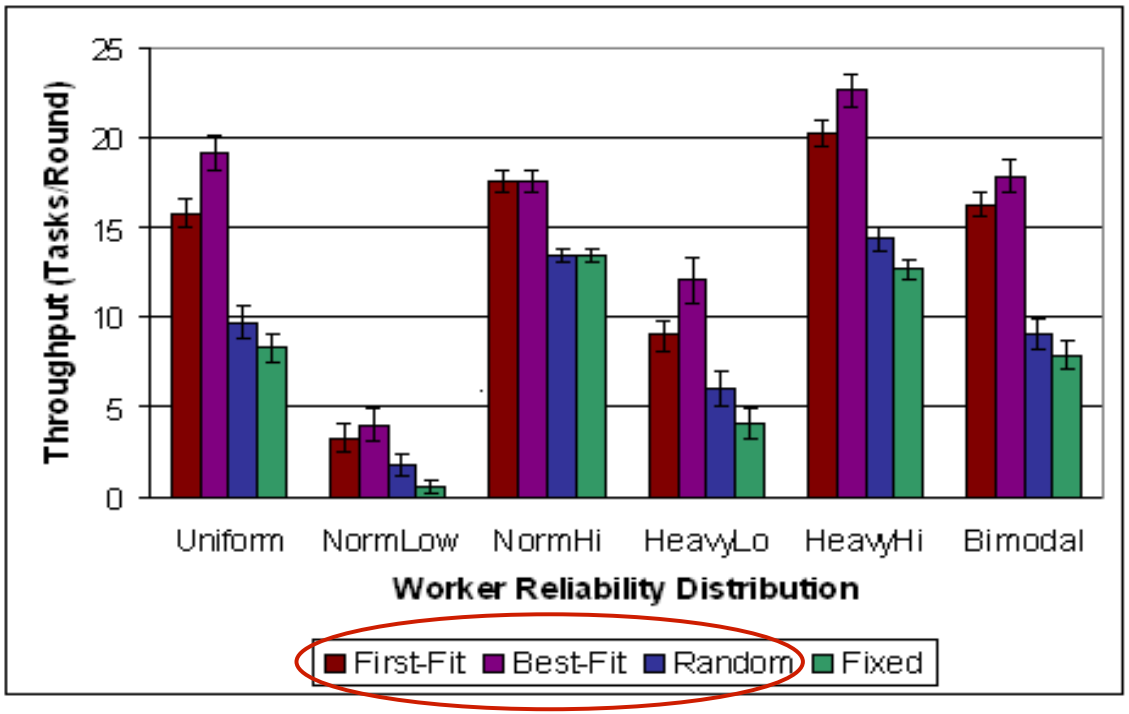
- Find compute nodes and data sources with high accessibility to each other
- Kim et al. [UM-TR08]: Use passive accessibility estimation



Data accessibility-based selection improves download time

Handling Failures

- Replication, state-maintenance
- Sonnek et al. [TPDS07]: Reliability-aware dynamic replication



Dynamic replication improves performance, reliability

Other Issues/Challenges

- Incentivizing Nebulas
  - Market economy, bartering, auctions
  - How to prevent cheating/freeloading?
- Deployment tools/APIs/client support
  - Virtualization, Middleware?
- Privacy/security issues
  - How to secure systems and applications?
  - We think: Nebulas not suitable for privacy-sensitive services

Summary

- Current Cloud models:
  - Well-provisioned, well-managed, centralized
- Some service classes:
  - Need loose performance, low/no cost, distributed data-intensive
- Nebula: Distributed, less-managed clouds
  - Use voluntary resources