

# Computing over the Internet: Beyond Embarrassingly Parallel Applications

**BOINC Workshop 09**

Barcelona

Fernando Costa



University  
of Coimbra

# Overview

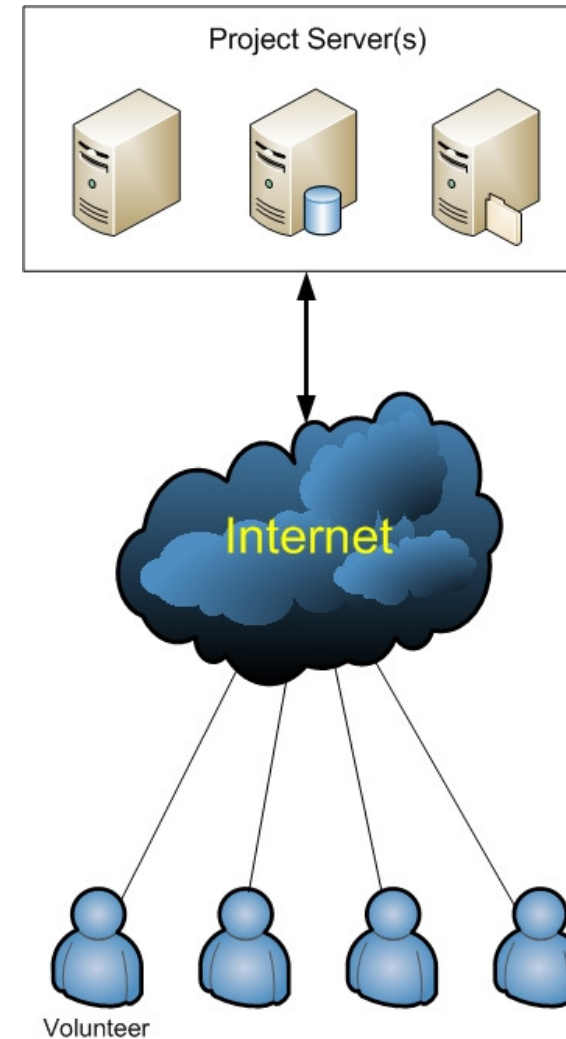


- Motivation
- Computing over Large Datasets
- Supporting new Applications
  - MapReduce over the Internet
  - Scientific Workflows
- Conclusion

# Motivation



- Volunteer Computing potential increasing
  - PS3, GPU
  - PCs have increased network and storage capabilities
- Limited to embarrassingly parallel apps
  - Master/worker model
- Limitations/Problems with current model

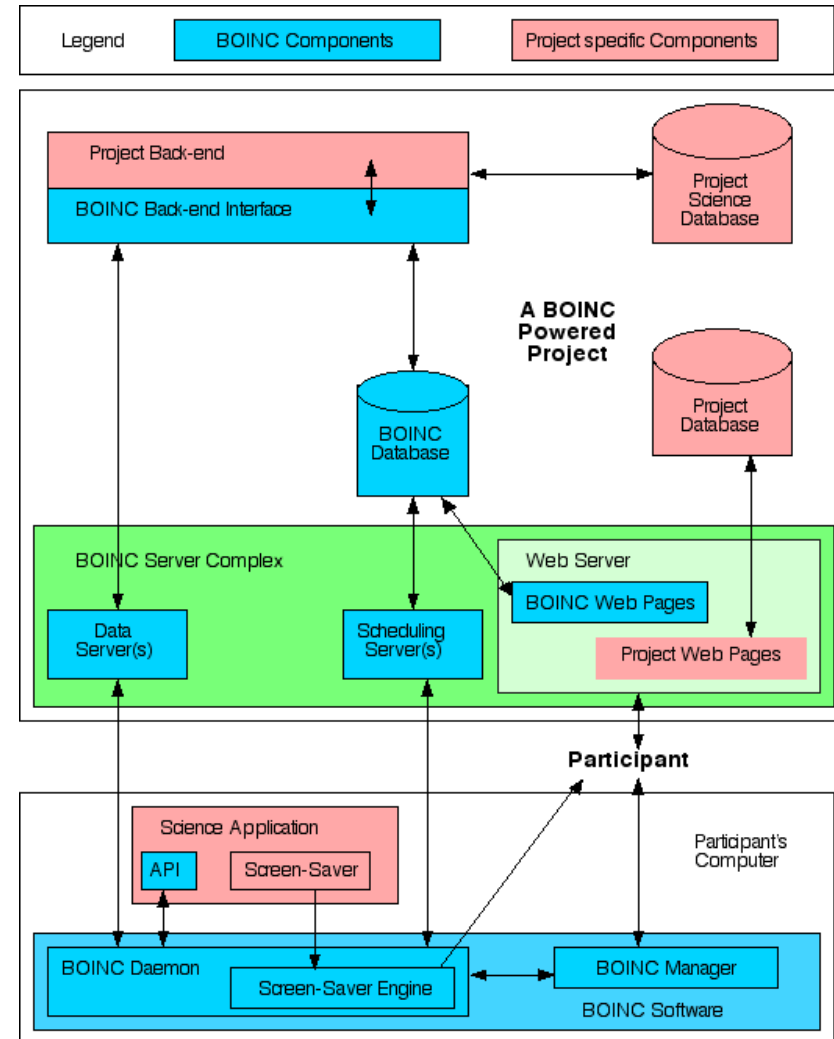


# Motivation - Problems



- Current Projects
  - Centralized architecture
  - Data distribution limitations
  - Storage problems
- Not many new projects
  - HPC stonewalled VC
  - New types of applications needed to reach new projects

Unable to take advantage of VC full potential



# Goals

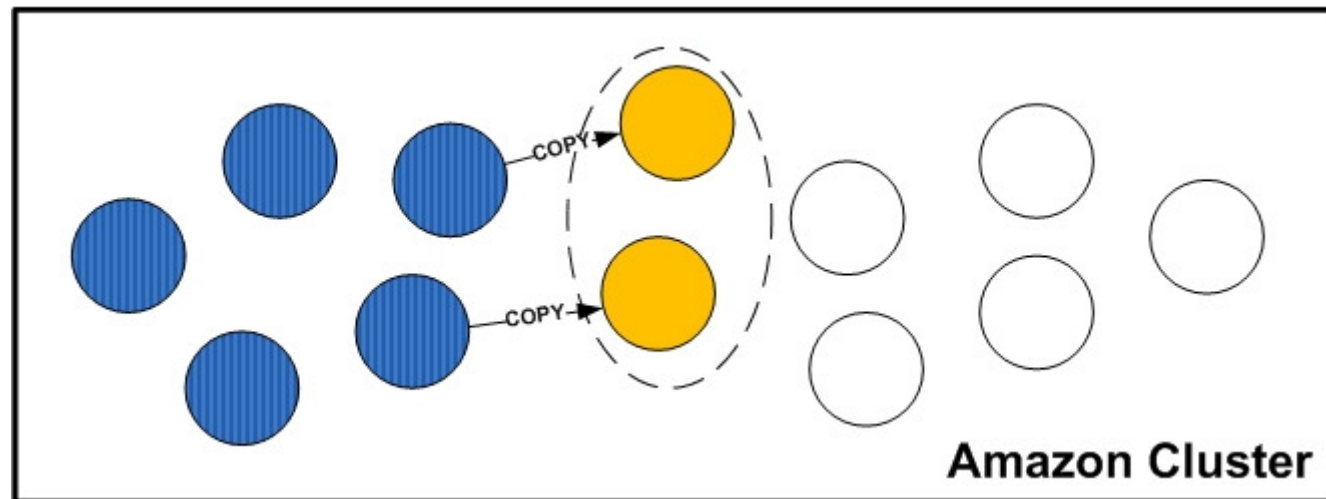


- Apply P2P techniques to solve scalability problems in current Volunteer Computing projects
- Introduce storage layer as support for new computing paradigms and application types
  - Allow new projects to use internet-wide computing
- Provide mechanisms to handle more demanding applications
  - Adapt existing Grid applications
  - Support data-intensive applications
  - Jobs with dependencies

# Computing over large Datasets



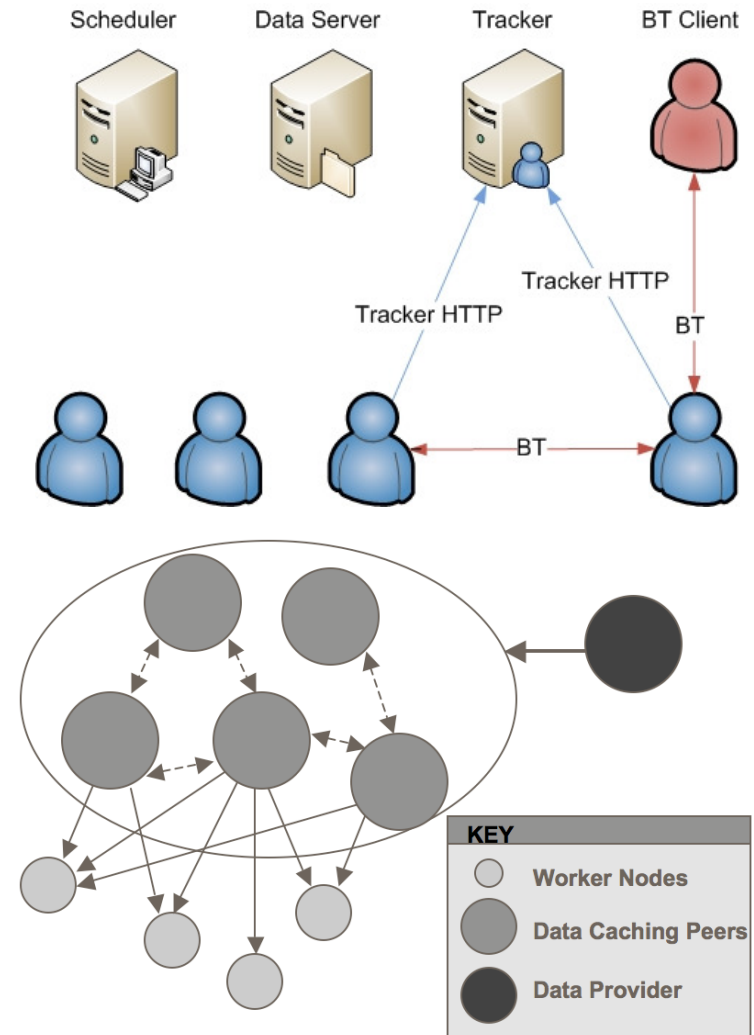
- Amazon Model
  - Store large datasets for free
  - Clients pay for computation and storage used by their applications
- How to adapt to BOINC?
  - Take advantage of previous work with BitTorrent



# Previous Work



- Improve data distribution
- BitTorrent
  - Shared input files
  - Proposal for a Collaborative CDN
- Super-peer organization (P2P-ADICS)
  - Data Centers
  - Data Lookup Service: DHT with volunteers



# Computing over large Datasets



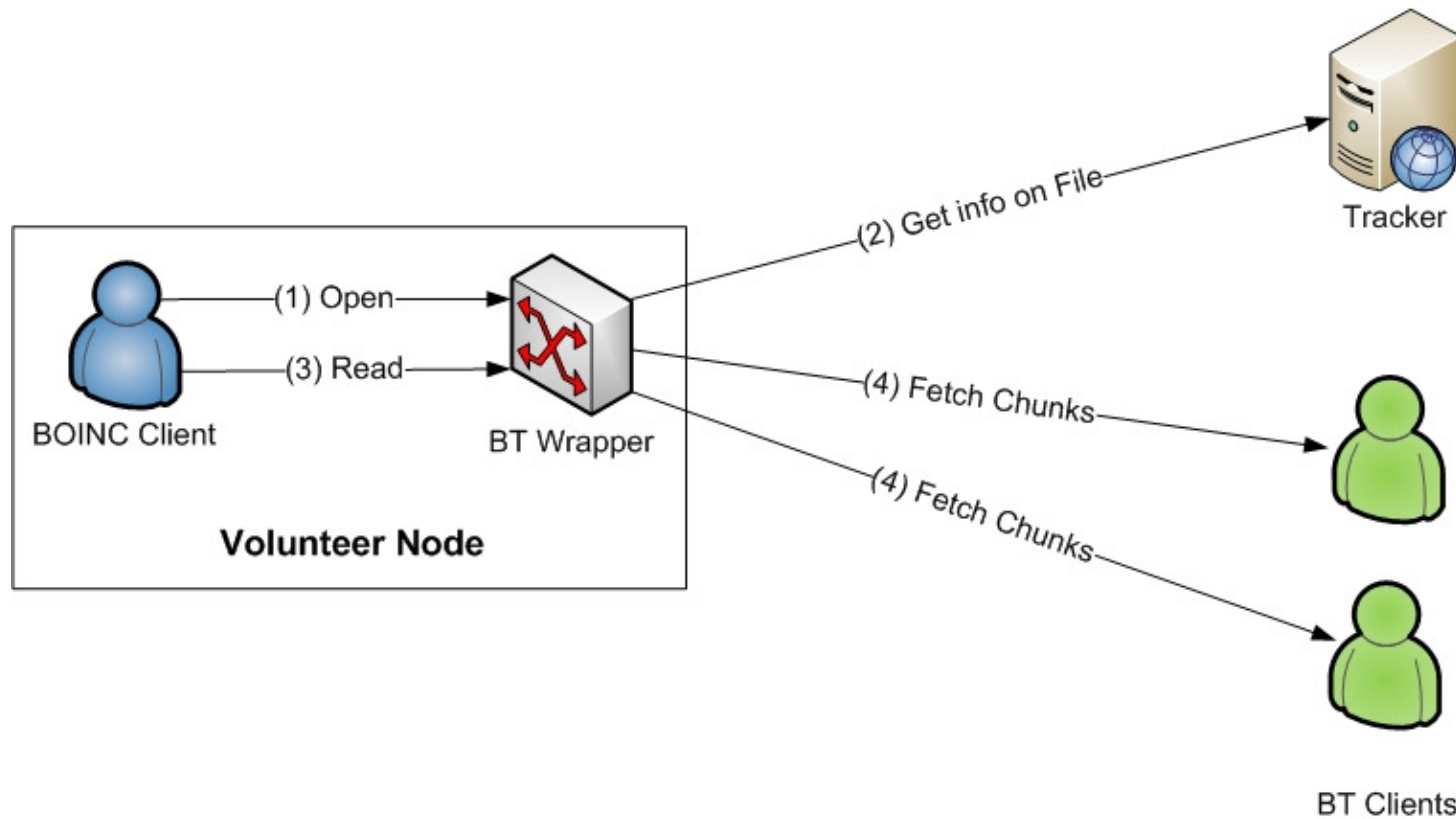
- BOINC + BitTorrent library
- Wrapper to set BitTorrent as read-only filesystem
- Use large datasets as inputs
- Possible command sequence:
  - `fd = open(tracker, objID)`
  - `read(fd, buffer, offset, len)`



# BOINC + BT model



```
fd = open(tracker, objID)  
read(fd, buffer, offset, len)
```





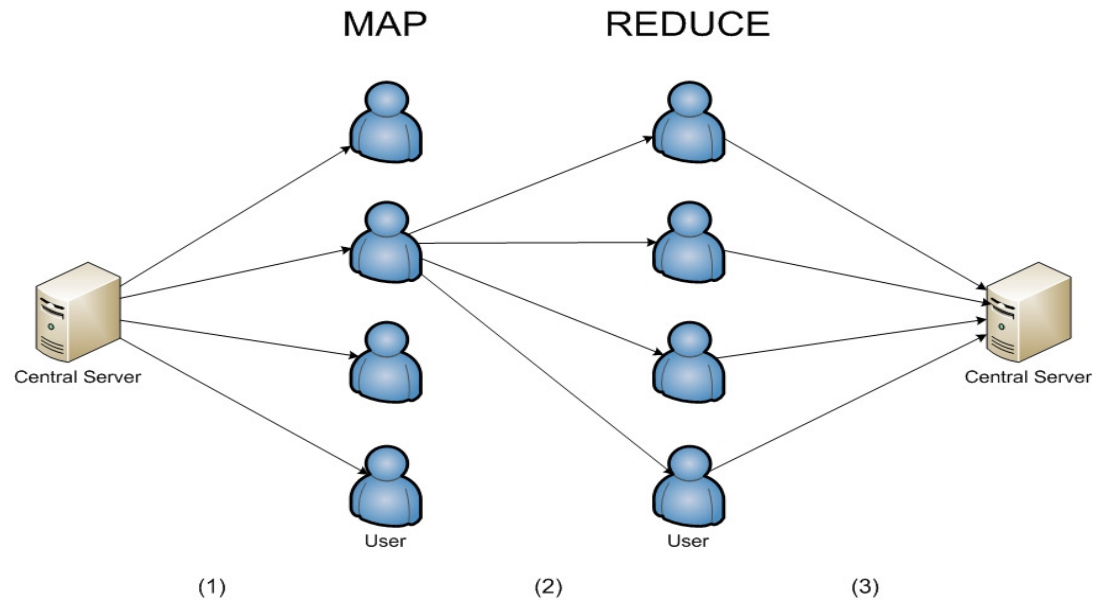
- Advantages
  - Easy to implement as first version
  - Allows initial testing to evaluate the solution
  - Possible to add read/write support
  - Next step: large outputs or intermediate results as inputs
- Problem
  - Assumes inter-client communication...
    - Solution: Guarantee that at least N% are accessible (public IP)
    - Communication over UDP – hole punching techniques
    - Turn this into a super-peer scenario?

# New Applications on BOINC



- Build over storage layer
  - Leverage direct transfers
  - Export information for applications
- MapReduce over the Internet
  - Wider use, but harder to find application
- Scientific Workflows
  - Not too complex for a VC environment

# New Applications - MapReduce

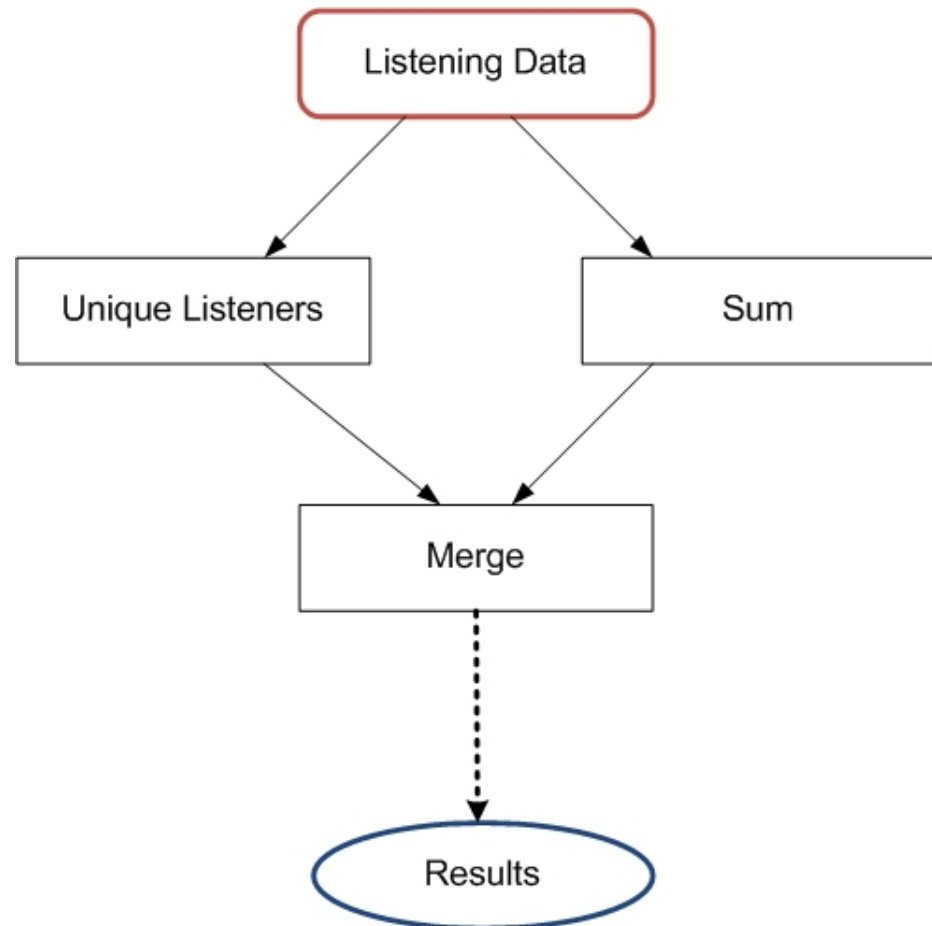


- MapReduce over the Internet
  - Adapt “Hadoop” to internet-wide computing
  - Volunteer Cloud Computing?
- Problems with typical applications...

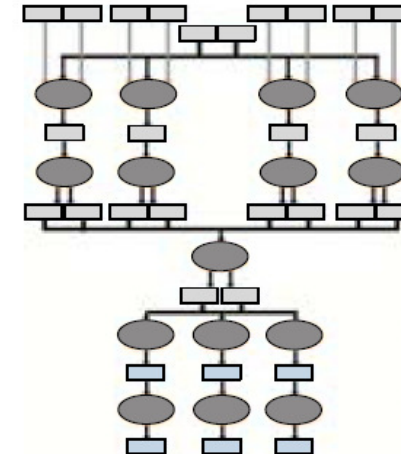
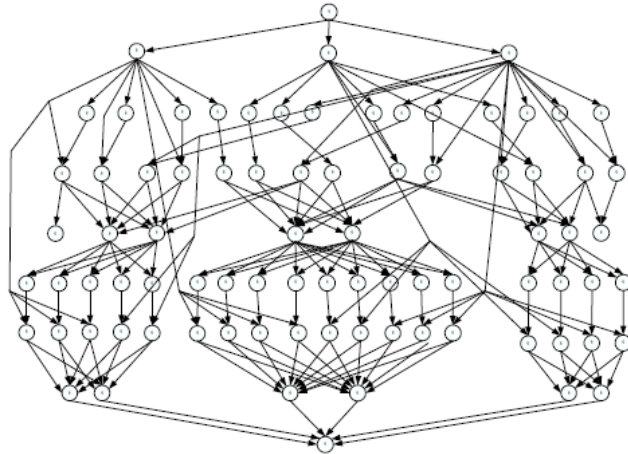
# MapReduce over Internet



- Applications that would fit
  - Lower Communication - Computation ratio
  - Longer running time
    - Lower latency requirements
  - More shared files
  - **Volunteer genomic computations ?**
- MapReduce Workflows
- Separate Dimensions



# New Applications - Workflows



- New types of applications
  - Data-intensive applications
    - E.g.: Handle CERN data-intensive computations
  - Workflow
    - Extremely variable characteristics: long or short running, data-intensive or compute-intensive

# New Applications – Current Work

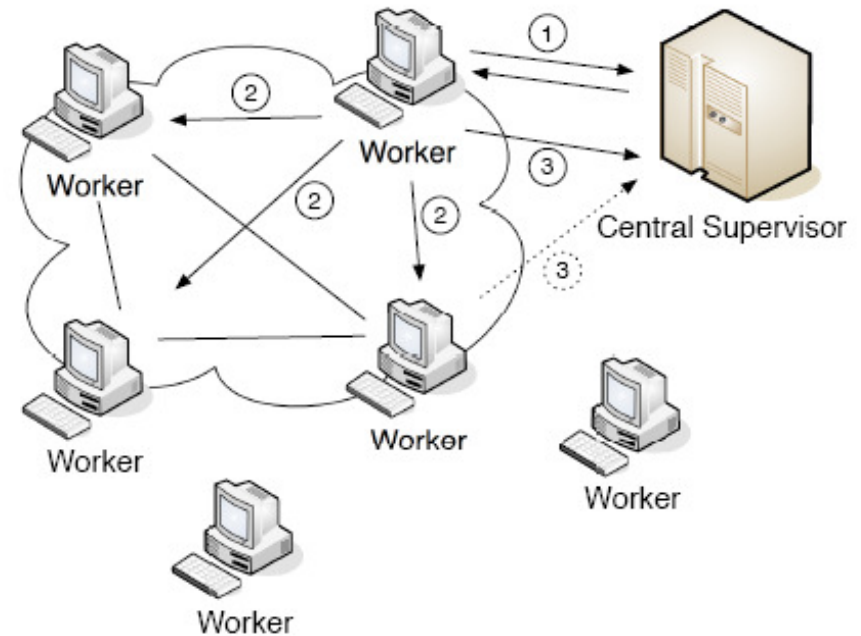


- Handling new applications
  - Science Workflows
- Volunteer storage system
  - Store intermediate results and final output
- Two alternatives
  - **Data stored in all nodes: metadata in central server**
  - Chosen nodes act as data centers

# Current Work - Cliques



- Clique
  - *Complete graph: each peer is connected to every node*
- Building the overlay/P2P system
  - Peers replicate data between themselves
- **Event-driven Simulator**





# Current Work - Cliques



- Advantages
  - More resources; Higher availability; Higher transfer speed;
- Disadvantages
  - Connectivity; Security; Upload bandwidth restrictions;
- New Issues
  - Accountability
  - Byzantine and selfish/rational behaviour
  - Fault Tolerance
  - Security
    - Authorization
    - Authentication

# Supporting New Applications



- Problem
  - How to find a suitable application?
  - Current Focus
    - Virtual machines, GPU and multiprocessor applications
- Build around existing application
  - Don't develop system that may never be used...
- No requests for computing against large datasets or workflow apps
- Solution: Collaborations with existing/new projects

# Conclusion



- Current Work
  - Building Volunteer Storage Platform
    - Wrapper to use BT as read-only file system
- Leveraging the Storage Layer
  - Working on simulator that uses Cliques to support workflows
  - MapReduce Paradigm
  - Data-intensive applications
  - Combining with virtualization: Volunteer Cloud Computing?
- Finding partners
  - Research is meaningless unless it is advantageous to **SOMEONE**