Attaching Cloud Storage to a Campus Grid Using Parrot, Chirp, and Hadoop

Patrick Donnelly, Peter Bui, Douglas Thain

Computer Science and Engineering
University of Notre Dame

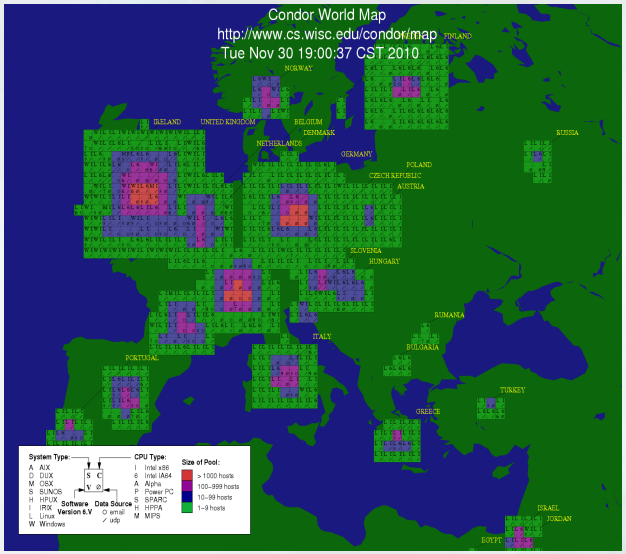
pdonnel3@nd.edu
pbui@nd.edu
dthain@nd.edu

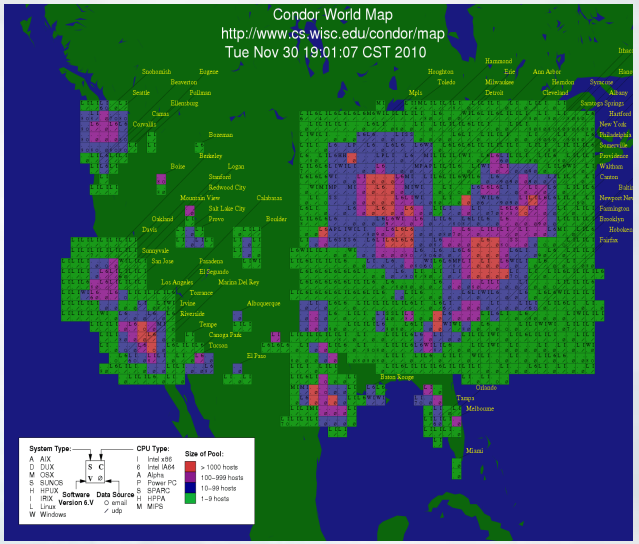
Overview of Talk

- **Problem:** Using our 1000-core Condor-based campus grid, we can generate much more data than we are actually able to store.
- **Idea:** Use Hadoop as a big, fast storage tank to service our campus grid!
- **Challenge:** Hadoop assumes a trusted local area network, which isn't the case on campus.
- **Solution:** Use Parrot and Chirp as a secure bridge between Hadoop and the campus grid.

- A campus grid is a collection of computing resources in a University setting or institution for idle cycle utilization.

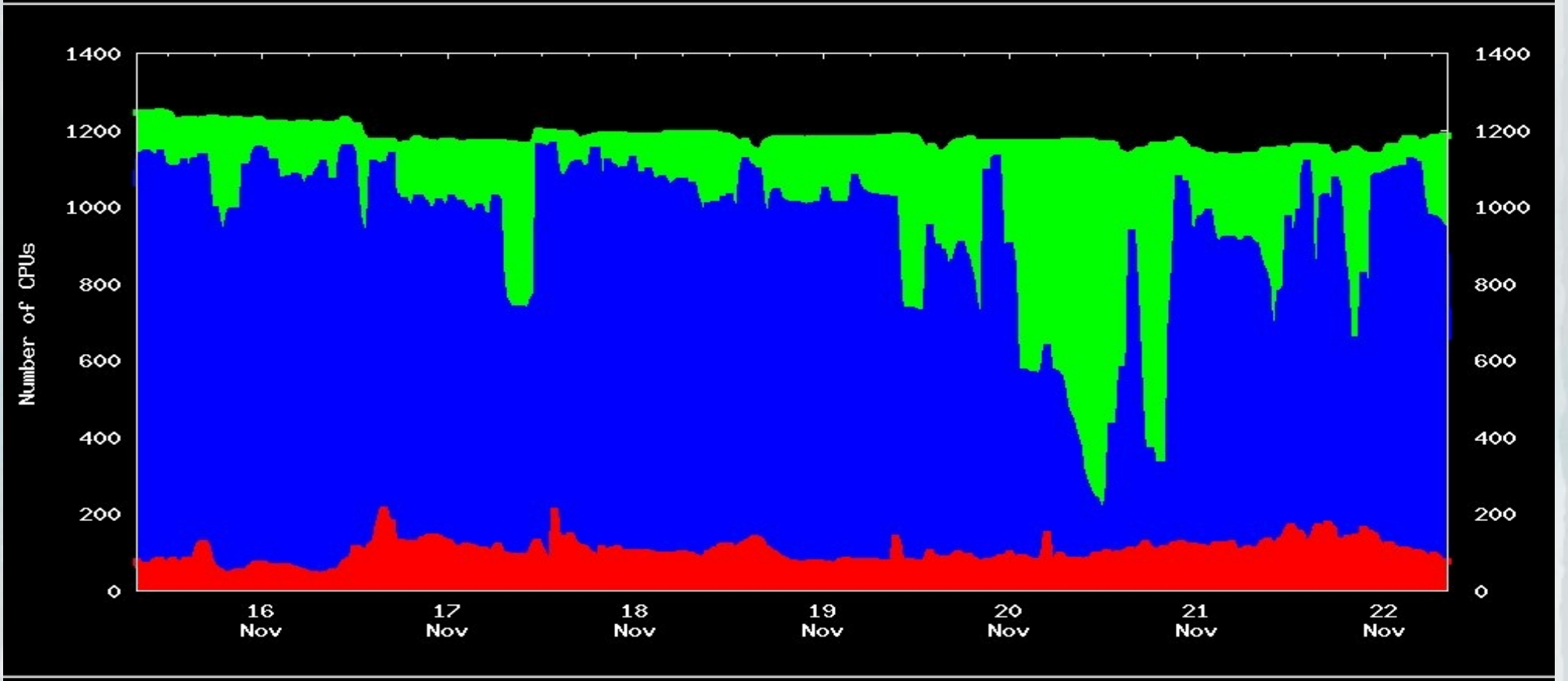
- Example Campus Grid Setups:
  - 1,100 cores at the University of Notre Dame
  - 20,000 cores in the Purdue BoilerGrid
  - 348,000 cores managed by Condor worldwide.





http://www.cs.wisc.edu/condor/map

CPU Utilization for the Last Week



44760	(22%)	CPU-Hours Unused
142277	(69%)	CPU-Hours Used by Condor
16351	(8%)	CPU-Hours Used by Owner
203388	(100%)	CPU-Hours Total

But...

1200 cores can generate a whole lot of data!

Can we store it in Hadoop?

Why Hadoop is Attractive for Campus Grid Computing

- Originally designed for web search engines that need highly scalable streaming access to large datasets.
- Usable for:
  - Processing thousands to millions of images in biometrics research.
  - Parallel read-mapping for next-generation sequence data in genomic research*.
  - Also used for machine translation, language modeling, and analyzing bulk text such as email or news papers **.

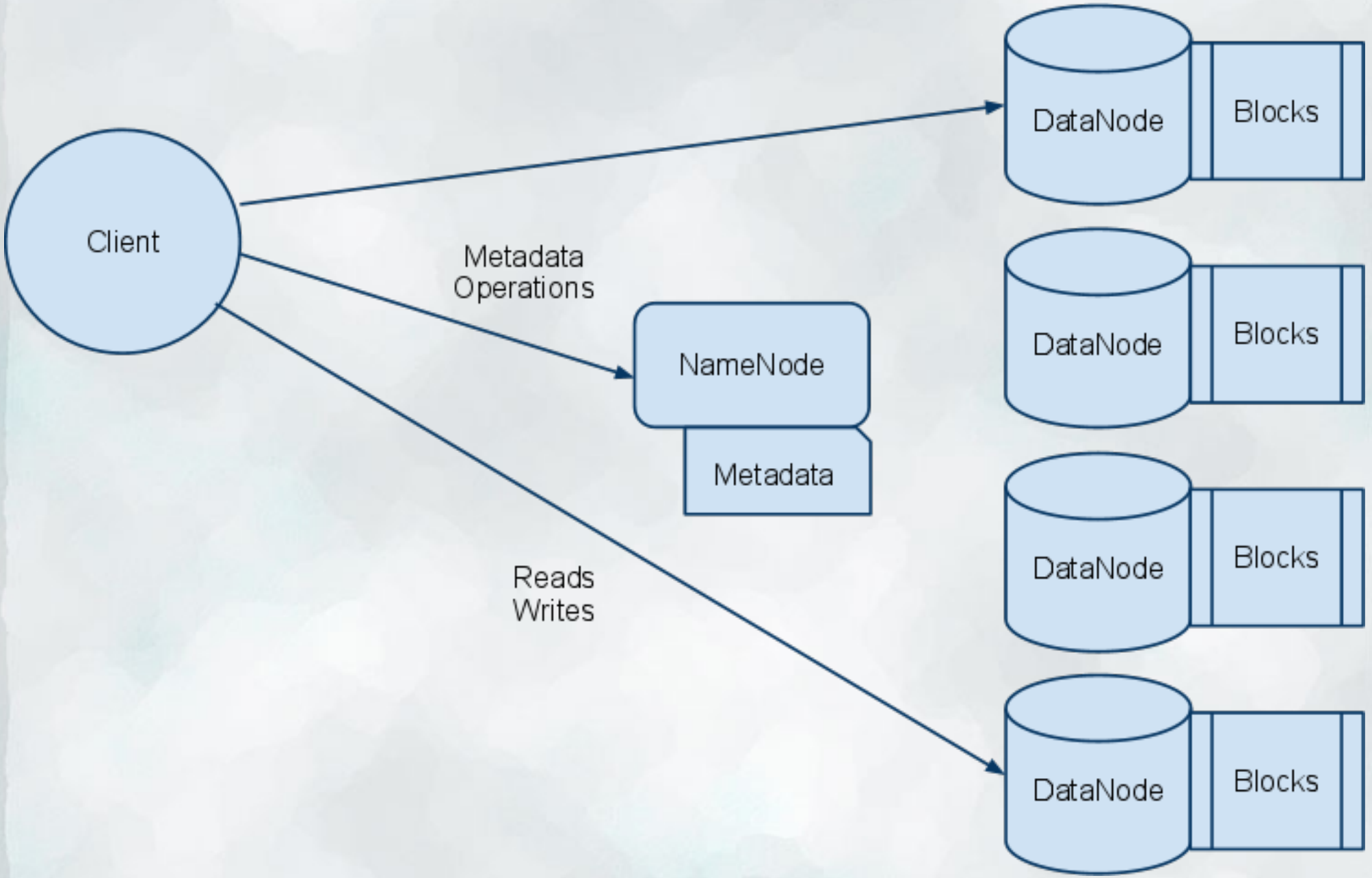
* Source: http://bioinformatics.oxfordjournals.org/content/25/11/1363.abstract
** Source: http://wiki.apache.org/hadoop/PoweredBy

The Hadoop Distributed File System

- Java open source implementation of the concepts in the Google File System.
- Offers very large file storage on the order of terabytes.
- Replicated file storage.
- Active Storage and Map-Reduce.
- Streaming data access.

Image source: hadoop.apache.org

Hadoop Architecture

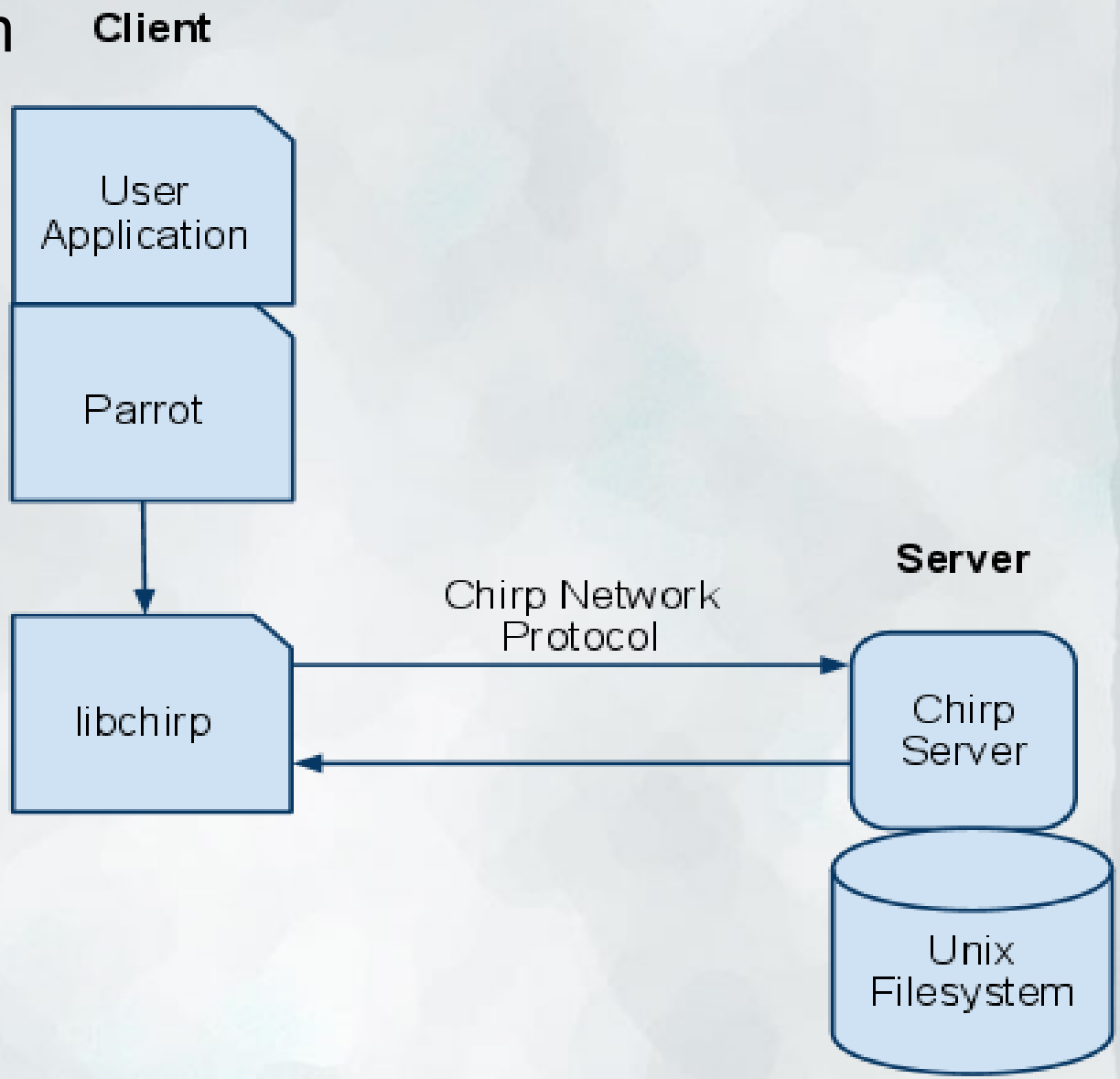


Suitability for Campus Grids

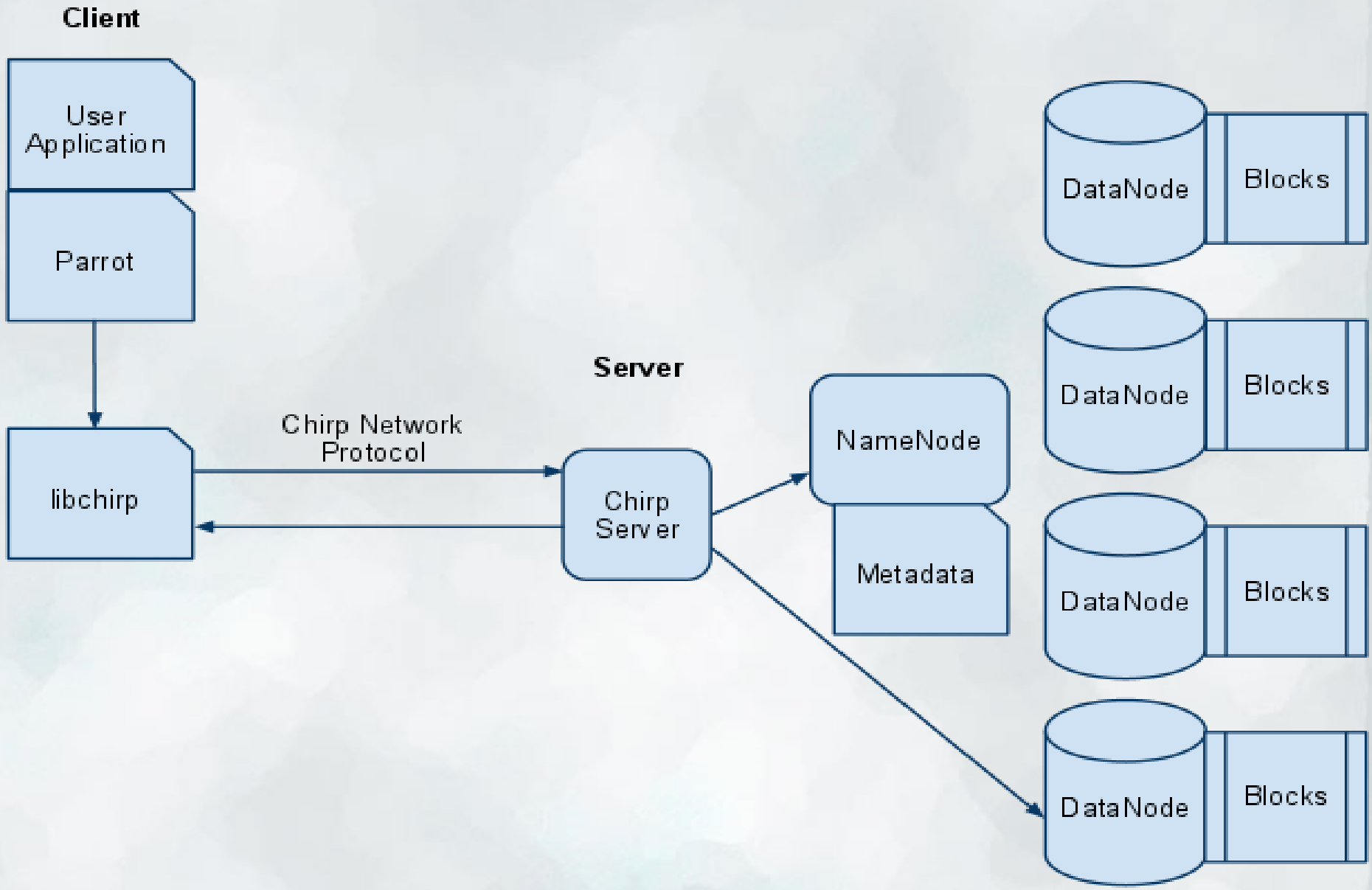
- Interface
  - Java API or POSIX-like C API
  - FUSE
- Deployment
  - Java Virtual Machine + Dependencies
  - FUSE
- Authentication and Security
  - No Authentication
- Interoperability
  - Tightly coupled components across versions of Hadoop.

Enter Chirp

- Distributed File System for use on a Grid.
- Exports file system on host.
- Userlevel filesystem.
- Secure authentication mechanisms.
  - Grid Security Infrastructure
  - Kerberos
  - Hostnames
- Security through Access Control Lists.



Chirp + HDFS


Client
User Application
Parrot
libchirp
Chirp Network Protocol
Server
Chirp Server
NameNode
Metadata
DataNode
Blocks
DataNode
Blocks
DataNode
Blocks
DataNode
Blocks

Back-end File System Multiplexer

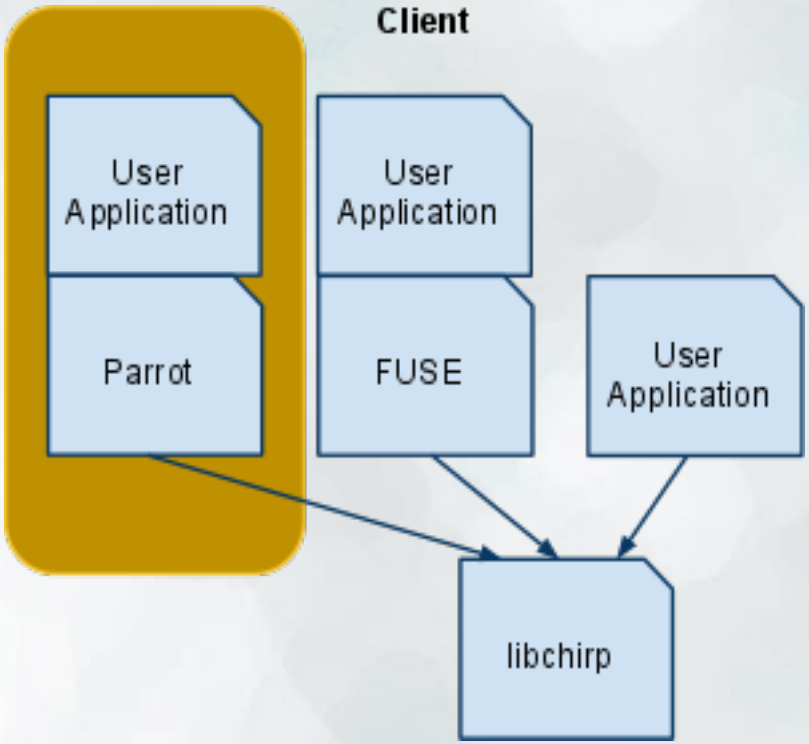
- Chirp multiplexes which underlying file system to access data.
  - Client need not know where the actual data is.
  - Applications can be programmed for a single interface without needing abstractions for different file systems.
- Unix VFS (local) filesystem and HDFS currently supported.

Parrot

- Chirp provides a *libchirp* library for client communication.
- We use Parrot to allow unmodified user application access to Chirp.
- Intercepts IO system calls on x86, amd64

```
$ parrot app /chirp/hostname:port/myfile
```

```
$ parrot /bin/sh
```



Actual Setup

Server setup:

```
$ chirp_server_hdfs -x namenode:9100 \
                    -p 9094 -r /path/to/root
```

Client:

```
$ parrot app /chirp/server:9094/file
  --> app /path/to/root/file
```

A Summary: Using Chirp and Parrot to bring Hadoop to the Grid

- Users can setup Chirp servers to give Grid access to a Hadoop cluster.
  - Strong Authenticated access. (Firewall Hadoop.)
  - Access Control Lists.
  - Easy userlevel deployment.
- Unmodified Application access to the Chirp server can be achieved using Parrot.

Questions?

Website: http://www.cse.nd.edu/~ccl

Chirp: http://www.cse.nd.edu/~ccl/software/chirp/
Parrot: http://www.cse.nd.edu/~ccl/software/parrot/

Patrick Donnelly: pdonnel3@nd.edu
Peter Bui: pbui@nd.edu
Douglas Thain: dthain@nd.edu