Managing Large Volumes of Distributed Scientific Data

Steven Johnston

School of Engineering Science
University of Southampton
sjj698@zepler.org

23 June 2008

UNIVERSITY OF Southampton
School of Engineering Sciences

Overview

BioSimGrid

Generic Data Challenges

File Object Database

Future Work

Questions

Introduction

- Biomolecular simulation repository
- Distributed across many geographical locations
- Large data volumes

Data

Large volumes of 'Data'

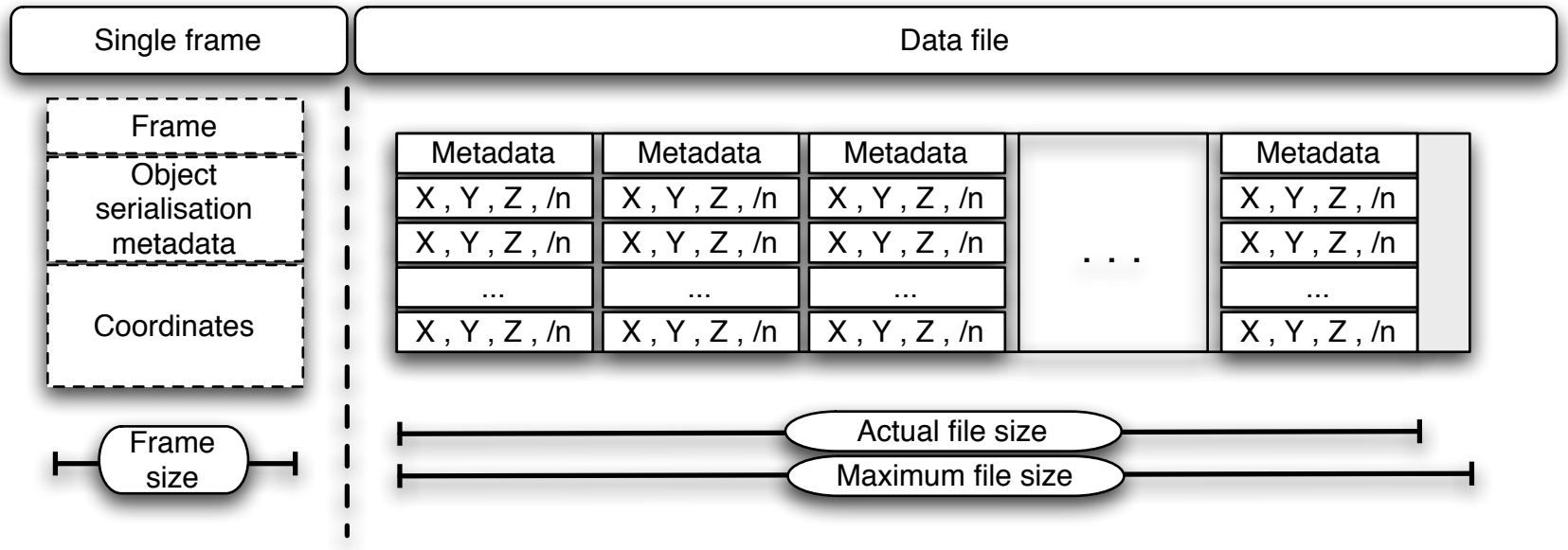
- Large data usually stored in a File System
  Trajectory of frames, each containing atom positions
- Metadata usually stored in a Database (small)
  - Information about a Trajectory (Simulation)
  - Information about the 'Data' (Size,locaion)

Data types

We can support

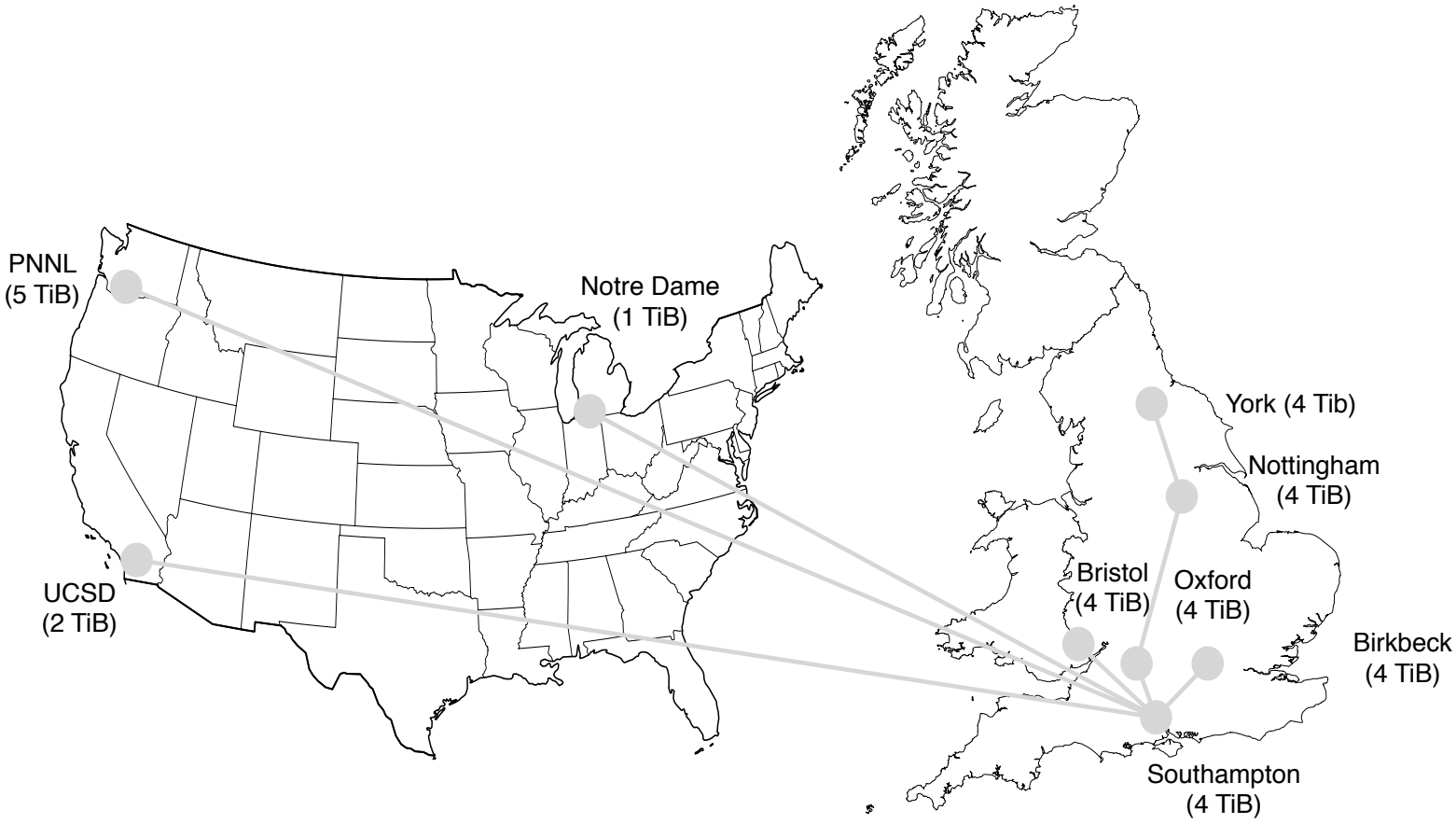
- netCDF
- Amber
- Gromacs
- Charm
- . . . supports plugable parsers

The BioSimGrid internal data format

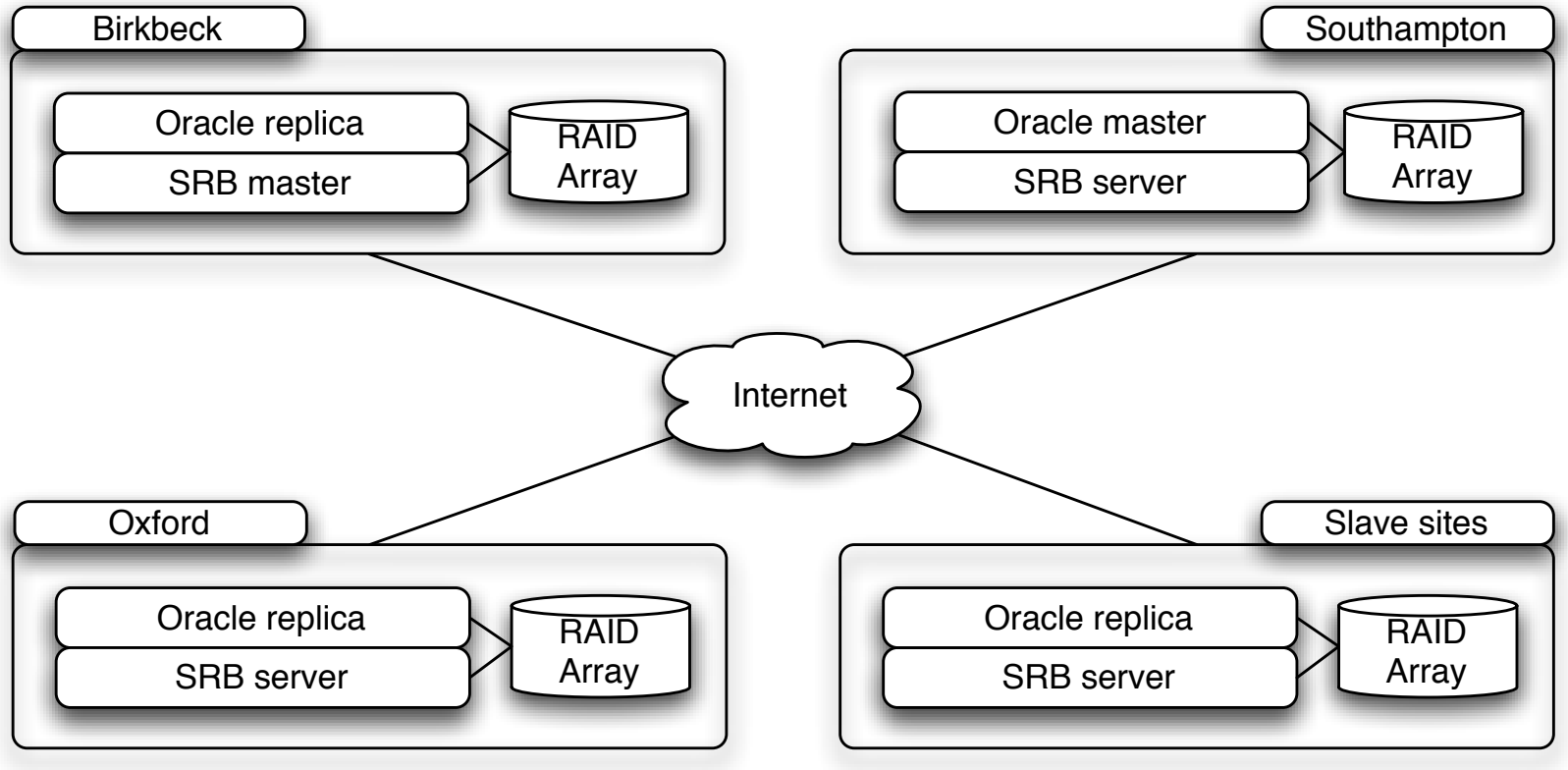


Need a parser for every format and format version.

BioSimGrid data storage locations



Infrastructure



Data Access

BioSimGrid can provide transparent access to . . .

- Each frame
- Each Atom in a frame
- An atom across a selection of frames
- A selection of frames

Data Analysis Tools

You can analyse data using a set of analysis tools

- Inbuilt tools
- Custom tools

Supports a plugable framework for tools

Data Analysis Model

- Move the compute to the data (or near it)
- Script are run locally accessing data across a distributed infrastructure

BioSimGrid Advantages and Disadvantages

Advantages

- Transparent data access
- Inbuilt tools
- Extensible framework

Disadvantages

- Security
- Data round trip
- Centralised database
- Data format compatibility/parsers

Generic Data Challenges

- The 'data' + 'Metadata' is a common scenario
- Large volumes of data are increasingly common
- Next step is to develop a method to address the general problem

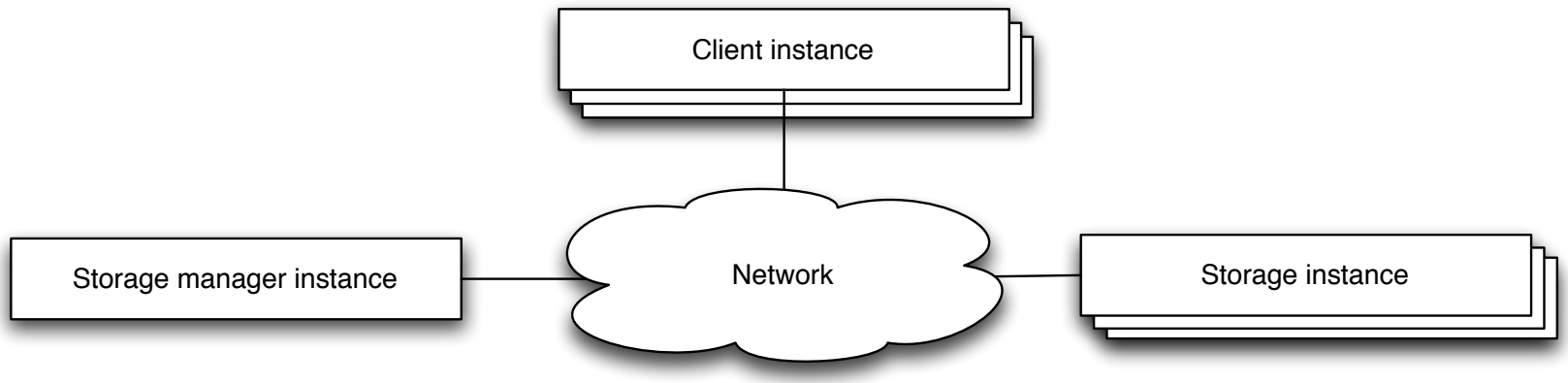
File Object Database

Storage layer

- Storage service
- Storage Manager
- File Object

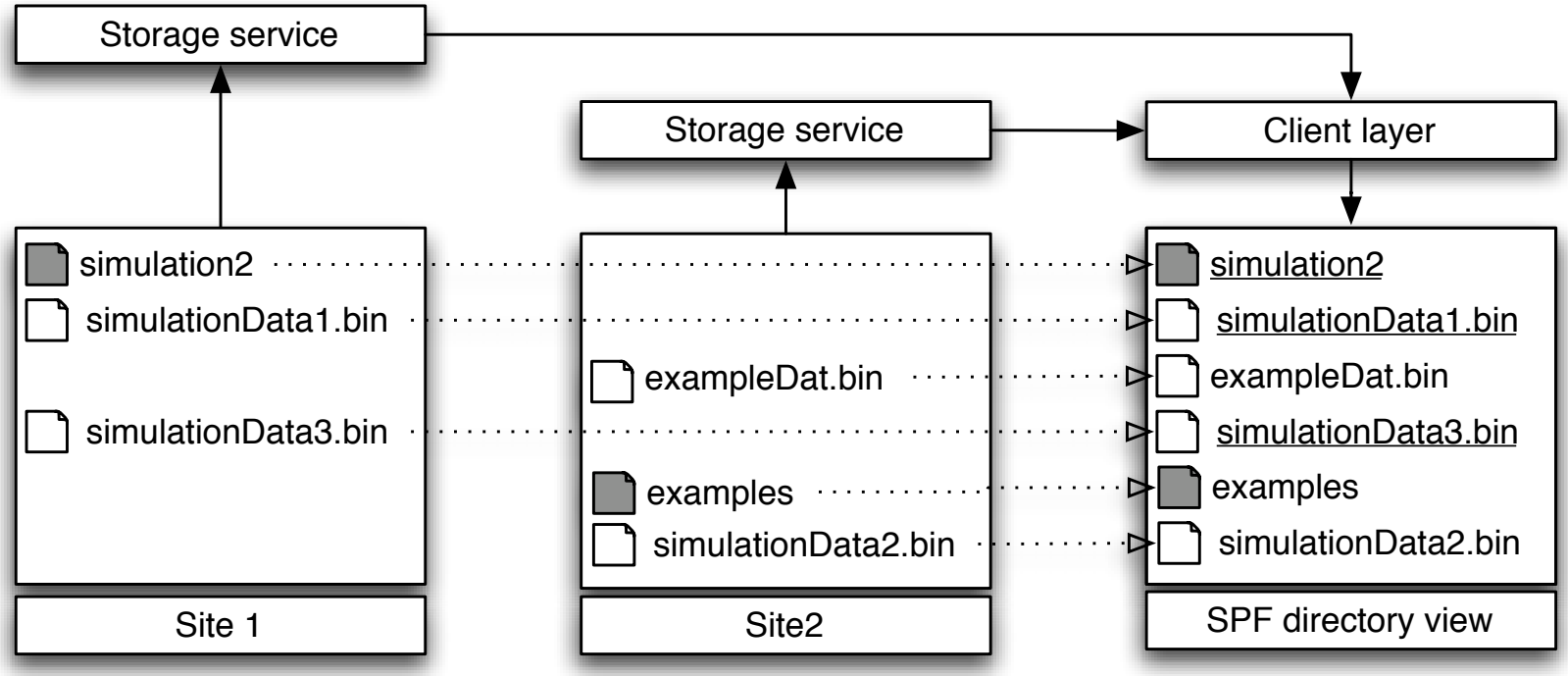
Client layer

- Client API



Storage Layer

- Storage Manager merges file view
- Client access the Storage Service directly
- Option to use a different filesystem



Storage Service

- Published known files
- Can register custom code to execute on filetypes
- Executes custom code

Storage Manager

- Unlimited number of Managers (23 in reality)
- Multi master, replicated database
- Stores files location(s), accessed time and MD5 hash
- Coordinates file requests from the client layer
- Coordinates requested custom code execution

Remote code execution

Storage service

- Associate code with a file type
- Client layer displays all 'methods' for a given file
- Request the execution of the code
- Code executes and returns an object

Advantages

- Keep the data in the original format/name
- Cache results
- Analysis on usage/performance
- Tools can be in any .Net language
- Utilise 'trusted' code
- Replication of data (Utilise excess bandwidth)
- Schedule automatic code execution

Disadvantages

- Metadata and data are managed independently
- Do I have the latest copy?
- Malicious code
- Hard to manage returned data volumes

Future Work

- Filestream in Microsoft SQL server 2008
- Custom iFilter and FullText search

For Further Reading

Related topics:

- SQL Filestream, FullText search

  http://www.microsoft.com/sqlserver/2008/en/us/default.aspx
- .NET
- BioSimGrid – http://www.biosimgrid.org
- SRB http://www.sdsc.edu/srb
- GEODISE – http://www.geodise.org
- Python – www.python.org

Thank you....

Questions, comments or suggestions
are always welcome.

Steven Johnston
sjj698@zepler.org

Acknowledgments
Prof. Simon Cox and Dr Hans Fangohr.

UNIVERSITY OF Southampton
School of Engineering Sciences