MooseFS and HTCondor in the design and implementation of Petabyte-scale computational clusters.

Daniel J. Shea

1 Introduction

Harvard Medical School's combined research data currently totals approximately 3.5 Petabytes (3,500 Terabytes) of data across several biological research domains. It is projected this data will double approximately every 16 months. It is essential to the continuance of modern biological research that this data is highly available and securely stored, enabling the application of modern computational methods to large data sets. This paper will outline the design and implementation of a distributed file system used within the Departments of Cell Biology and Biological Chemistry and Molecular Pharmacology here at HMS for the storage of research data. It is the author's intention to provide a framework by which other organizations may examine how they may implement the methodologies used in the creation of this system and apply the lessons learned within their own organization for the storage and retrieval of large data sets used in scientific research.

2 Design Considerations

There are several architectural considerations that must be examined in the deployment of large scale storage systems. The use case of the storage cluster must be complementary to the profile of the data sets to be stored. File sizes, overall numbers of files and most importantly the interaction with the end user of the system are design factors that must be addressed when determining optimal system configurations and tuning parameters. Our example architecture is used in a single lab for the processing of NMR (Nuclear Magnetic Resonance) data. For the purposes of our design in consideration of the data profile, NMR data formats used in spectral analysis of molecular models is that of many small files often held within a single directory. This data is utilized by computational software packages (such as sparky) to perform signal analysis of the data generated by many runs of NMR magnets.

TBD: Add average number of files per directory, size of files and other relevant information here.

3 Architectural overview

Our storage cluster makes use of the MooseFS Filesystem. MooseFS is a fault tolerant, network distributed file system. It spreads data over several physical servers which are visible to the user as a single resource. For standard file operations, MooseFS acts as other Unix-alike file systems providing a hierarchical directory structure, POSIX file attributes (permissions, last access and modification times), special files (block and character devices, pipes and sockets), symbolic links (file names pointing to target files, not necessarily on MooseFS) and hard links (different names of files which refer to the same data on MooseFS.) Access to the file system can be limited based on IP address and/or password.

TBD: Add further overview of MooseFS, overview/implementation/architecture of HTCondor, system architectural diagrams and overall system architecture here.

4 Performance Tuning

TBD: Outline performance tuning optimizations, reasons for each tuning parameter choice and overall net benefit achieved by the choice

5 Conclusion

TBD: Summarize findings of use of cluster, performance numbers, etc.