Technical Report NEESgrid-2002-06
www.neesgrid.org

(Version: 1.0
Last modified: September 30, 2002)

Storage Technologies for the NEESgrid Curated Data Repository

Joe Futrelle[1]

[1] National Center for Supercomputing Applications, Urbana-Champaign, IL 61820

Feedback on this document should be directed to futrelle@ncsa.uiuc.edu

Acknowledgment: This work was supported primarily by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0117853.

1 Overview

The NEESgrid curated data repository requires high-volume, high-throughput, reliable, secure storage technology that can provide ready access to heterogeneous data ranging from highly structured metadata to relatively opaque formats such as digital video. In addition, the technology needs to be mature and flexible enough that data can be preserved even when storage technology and data practices inevitably evolve. A number of storage and access technologies exist that meet some of these requirements, including commercial database management systems, standard data and metadata formats, standard data exchange protocols, network file systems, digital library technologies, and storage brokers. However none of these technologies by itself meets all of NEESgrid's requirements, so an appropriate combination of them should be employed. In the end, it is our opinion that maturity and flexibility are by far the most important requirements, since by meeting them we can minimize the risk that the storage technology will become obsolete and therefore necessitate extensive re-implementation of the repository.

NEESgrid is also a distributed system. Although the curated data repository will be centrally located in order to take advantage of economies of scale and favorable network location, it will still need to interoperate with local storage mechanisms at the sites. For this reason, the technologies used should require as little specialized expertise as possible to maintain. Otherwise, there is a risk that sites will have difficulty moving their data to and from the repository, and that new sites will find it difficult or costly to join NEESgrid.

2 Requirements

2.1 High volume and throughput

Current EE data practice involves gathering time series kinetic data from sensors, along with video and/or audio recordings. As the practice evolves, it is reasonable to expect that the volumes of both kinds of data will grow as the price of sensors and cameras drop and their resolution increases. A site today might use three or four analog video cameras and 64 sensors for an experiment. In the foreseeable future, we can expect that sites will use dozens of high-resolution cameras and hundreds of sensors – and there's every reason to expect that those trends will continue.

In general, the storage requirements for video far exceed the storage requirements for sensor data. This is because sensor data is in effect two-dimensional (sensor × time) whereas video data is four-dimensional (camera × x × y × time). The difference in storage requirements can be expressed in terms of the data rate required to transmit the data in real time. The data rate of sensor data is on the order of 10Kb/s, depending on the sampling rate. Data rates for various video formats are shown in Table 1.

Table 1: Video data rates

Data rate (Mb/s)	Description	Format
<0.384	Video conference	MPEG-4
<1.5	Video in a window	MPEG-1

1-2	VHS quality full screen	MPEG-2
8-10	Professional PAL	MPEG-2
32-40	Professional HDTV	MPEG-2
168	Raw NTSC	Uncompressed
216	Raw PAL	Uncompressed
270	Raw contribution PAL	Uncompressed
1-1.5 Gb/s	Raw HDTV	Uncompressed

Audio rates are comparable to low-quality compressed video. For instance, the data rate of uncompressed CD-quality audio is 1.35Mb/s. Since most video formats encode audio along with video, and audio and video data formats involve similar technology, let us consider the two together as one class of data.

Lossy compression can reduce the video data rates considerably (as Table 1 shows), but can also make it difficult to manipulate video without additional losses in quality resulting from re-encoding, and can impair the performance of video analysis codes.

The implications of video data rates on the NEESgrid storage strategy are significant. Whereas sensor data for a significant collection of EE experiments could be stored on any desktop machine and transferred over wide-area networks much faster than real time, video data for a single experiment could easily be so voluminous as to make both impossible. NEESgrid therefore requires terabyte-scale storage capacities and Gb/s bandwidth across wide areas (or better, if such a WAN existed), as well as technologies that take maximum advantage of these capacities, such as high-performance networking protocols.

Compared to sensor data and video, metadata imposes very light storage requirements. Although current EE practice for the collection and storage of metadata ranges from *ad hoc* to nonexistent, we can expect that metadata will become more complex and voluminous as the community continues to develop data sharing practices. However, metadata is typically highly structured and therefore can be greatly compressed without loss.

2.2 Reliability

NEESgrid data must not be at risk of being lost, corrupted, or unavailable for significant periods of time. Infrastructure components such as uninterruptible power supplies and automated backup can go a long way towards meeting this requirement. Local storage resources at sites with their own backup facilities can ensure that data can survive WAN outages and catastrophic failure of the central repository. The most significant challenge NEESgrid faces with respect to reliability is how to back up and restore high-volume data such as digital video. This requires terabyte-scale archiving technology.

2.3 Security

The NEESgrid curated data repository requires secure access to data. Users must be authenticated and audit trails must be kept. Access control must be fine-grained enough to be applied to individual events or parts of experiments, rather than simply to files or

entire projects. Transactions with the NEESgrid repository should be confidential. Public-key encryption should be used to encrypt data and credentials in transmission.

2.4 Ability to handle heterogeneous data

NEESgrid data represents the varied efforts of multiple communities of EE researchers. In order to maximize its usefulness, this data must be represented using data structures that represent objects of interest to EE researchers as well as capturing the relationships between these objects. The meanings of the elements of the data structures must be agreed upon as much as possible by members of the community, so that the accumulated mass of EE data in the repository will be worth more than the sum of its parts. A certain uniformity of structure will go a long way towards enabling search and analysis functionality for the repository, so that users will be able to locate data of interest to them without immersing themselves in the specific details of each experiment.

In order to facilitate this, the NEESgrid repository must be able to deal not just with data files, but with data objects with complex sets of attributes and relationships to one another. Furthermore, it must be able to link these objects to relatively opaque file formats such as image formats. For instance, it should be possible to link objects such as specimens to image files containing digital photographs of them, so that a user could, for instance, find the photograph given the specimen.

Supporting complex objects and inter-object referencing requires an object-oriented data model, which is currently under development. NEESgrid must use storage technologies that support this model.

3 Storage and access technologies

3.1 Data formats

Distributed storage and transmission of data in heterogeneous computational environments requires data formats capable of completely representing the structure and content of the data. First and foremost, this requires data formats that are independent of the processor architecture of the machines on either end of a transaction. For instance, numeric data must be represented in a format that does not require that the machine reading or writing it use a particular byte ordering. EE practitioners solve this problem currently by representing numeric data as text, an extremely space-inefficient but otherwise very convenient platform-neutral means for representing such data. For text, formats should use specified character encodings, so that operating systems do not corrupt the data during encoding translation.

Secondly, standard data formats should be used whenever possible, to capitalize on the existing set of free or inexpensive software available for processing data in those formats. For instance we should not invent a NEES video format when there are so many adequate standards and variants of standards from which we can choose. Nor should we develop a specialized markup language for metadata when XML provides a standard generalized markup language which is compatible with countless tools and related technologies. In fact, for many categories of data, NEESgrid should simply adopt existing data formats

with minimal adaptation. Table 2 enumerates several categories of data along with relevant standard data formats and possible adaptations required.

Table 2: Data formats and adaptations

Kind of data	Data format(s) or frameworks	Adaptations required
Video/Audio	IEEE 1394 ("FireWire"), MPEG	Specify resolutions, frame rates, codecs
Sensor data	ASCII, XML, HDF	Specify syntax, structure
Metadata	XML, RDF	Specify schema, ontology, data dictionary
Solid models	STEP	Build library of reusable components
Simulation models	OpenSEES	Extend to support import/export of models
Documentation	XML, PDF	Specify syntax, structure

The EE community uses a variety of commercial software tools that use proprietary data formats. Some of these tools can be replaced with tools that use standard formats, but some can't. In these cases, NEESgrid can provide the means to store and retrieve these formats, but will only be able to provide limited access to the contents of data objects so formatted. In some of these cases, format conversion codes will be able to be provided. In general, the repository should provide the ability to translate between several formats where appropriate and possible. For instance XML metadata may be able to be translated to and from a set of NEES-specific Excel templates, for users who may want to be able to view and manipulate the metadata using that application.

3.2 File systems

Data such as video and sensor data will be stored in files in file systems. File systems differ greatly with respect to performance and security, so it is worth considering what kind of file system ought to be employed for various data components in NEESgrid. In particular, network file systems are notoriously slow, although in other respects they offer important advantages such as transparency. Local file systems tend to be tied very closely to operating system implementations, so their performance tends to be much more closely related to storage hardware.

For the central repository, NEESgrid will require a multi-terabyte storage system with redundancy, security, and automated backup. The NCSA mass store system (mss) meets several of these requirements. Based on UniTree, mss provides a multi-level file system consisting of RAM, disk cache, and tape. Recently-used data is kept in the cache, and less recently-used data is transparently written out to tape. Access to the data is through GridFTP, which results either in retrieval from the cache, or from tape using a tape-loading robot. The mss does not provide the redundancy or performance required for near real-time access to data for NEESgrid. This requires yet another level of disk cache interposed between NEESgrid users and mass store, which takes the form of a high-

performance disk array on a NEES-dedicated machine located in close network proximity to the mss.

For local storage, we recommend that sites maintain local storage systems with sufficient capacity for several experiments' worth of data. For a site which does a moderate amount of video, this can easily add up to 0.5-1TB of disk space. For performance reasons, we recommend that the local storage system be accessible using GridFTP.

Enforcing security policies on distributed, heterogeneous file systems is complicated. The Grid Security Infrastructure (GSI) goes a long way towards addressing this issue. In all but a few instances, NEESgrid components will exchange files using GridFTP, which will ensure consistent authentication throughout NEESgrid as well as providing public-key encryption for data being sent over the network. For data components, access control will be enforced through higher-level, NEES-specific mechanisms.

3.3 Database management systems

Highly-structured data such as metadata requires storage technologies capable of structured access modalities, such as queries against collections of objects, or partial updates to complex objects. The most mature and robust technology of this sort is the relational database management system (RDBMS) and the associated standard query language (SQL). Numerous commercial and non-commercial RDBMS's are widely available, and each has its strengths and weaknesses.

Since the NEES data model is object-oriented, it raises the issue of whether the data repository should be implemented using an object-oriented DBMS (OODBMS) or object-relational DBMS (ORDBMS). Although much research has been done in this area and several commercial implementations are available, it is not clear that the technology is mature enough to be deployed for NEESgrid. Standardization on the object-relational extensions to SQL has not been adopted industry-wide, and it may be difficult or impossible to achieve comparable performance from one system to another. Using an object-relational system to store data may make it difficult to move the data between one commercial database system and another, which could make it costly should such a move become necessary. Object-oriented databases are even less mature than object-relational databases, and for that reason should be considered only as a research topic.

Fortunately, even without object-oriented technology in the DBMS, RDBMS's provide a number of important capabilities upon which we can build logic to support NEESgrid's data model. For reasons of flexibility, the implementation should not use any vendor-specific database features; instead it should be based on SQL92, a version of SQL which virtually every commercial database vendor and several important non-commercial RDBMS's support.

4 Conclusion

This report has covered requirements and technologies for the NEESgrid curated data repository. Table 3 summarizes the requirements and the technologies that address them.

Table 3: Requirements and technologies

Requirement	Technologies
High volume	Disk arrays, tertiary storage (e.g. mss)
High throughput	Gb/s networking, GridFTP
Reliability	Redundancy, automated backup
Security	GSI
Heterogeneity of data	Standard file formats, object-oriented data model, RDBMS's