

Workload Characterization and Analysis of Storage and Bandwidth Needs of LEAD Workspace

Beth Plale
Indiana University
plale@cs.indiana.edu

LEAD TR 001, V3.0

*V3.0 dated January 24, 2007
V2.0 dated August 19, 2006
Original dated October 13, 2005*

The purpose of this study is to define a use workload for the LEAD grid based on the categories of users supported and an understanding of the kinds of tasks they need to accomplish. We then use this workload to analyze the storage and bandwidth needs of LEAD. New to V3 is a rethinking of the model for storage needs, a model that we think now more accurately captures deletes. Spinning disk and archival storage needs are now differentiated. New to V2 is a rethinking of the file size distribution, and a cost model based on the Amazon S3 pricing model (as publicized in a March 2006 user guide.)

The key calculations can be summarized as: file read/write rate is 5 files per minute uniformly distributed over a 12 hour period for a sustained bandwidth of 864 Mbps. The spinning disk needs are 31.2 TB, which supports the most recent 1 week of user generated data plus an additional 20% for cache from archival storage. In the absence of more aggressive compressing and pruning of data products (the number we calculate here has none), three months of data requires 312 TB of storage.¹

The LEAD grid is a service oriented architecture (SOA) framework running on a set of machines that provides the researcher with the capability to conduct computational science-based experiments and discovery. The LEAD SOA is a highly distributed application, running services located at partner institutions and utilizing the Teragrid for large-scale computation, such as running computation-intensive data assimilation and simulations.

In this document, we define a realistic user workload. Workloads are commonly used in experimental computer science to test the scalability and reliability of a software system. A *workload* is a set of requests, often based on an understanding of realistic user needs, then scaled synthetically to create workloads anticipated in the future, or large enough to stretch-test features of a system. For instance, a web server workload could be a collection of GET and PUT requests issued to a web server at some rate and in some particular order for static and dynamic content.

¹ This work funded in part by the National Science Foundation through cooperative agreement ATM-0331480.

Using the workload we have defined, we analyze the data-related resource demand, specifically bandwidth in and out of the repository, and spinning disk and tape archive needs.

In LEAD, work is carried out by means of workflows, which we define as a directed graph of tasks, where data flows between tasks as control transfers along the edges of the graph. We classify workflows into 4 types based on the targeted user groups in the project. Workload activity is modeled over a 12-hour period; 12-hours because that is the duration of the canonical simulation. During any 12-hour period, 125 users will each execute 4 workflows each. The number of workflows of each type executed in the 12-hour period is guided by an expected distribution.

Data products used in and generated during a workflow are stored to the LEAD workspace. The repository for the workspace supports private and shared collections. It must provide means for moving collections in and out of the repository. It ensures authorized and authenticated access to private and shared collections, and ensures availability, good performance, fault tolerance, and preservation for the data collections it maintains. This workspace storage system is in addition to community data collections, collections of observational and some model generated data that has been available to the meteorological community for a number of years. The LEAD workspace is new to LEAD, and it is the storage for this resource that is estimated here using the workload we define.

1 Workload: Job Types, Number of Users, and Duration of Activity

The use cases of LEAD can be categorized into four “job” categories, where a job category is a type of task that a researcher carries out on the LEAD grid. A **workflow** is a directed graph of tasks where a *task* could be for instance, assimilation, data transformation, data analysis, or simulation. A directed edge exists between tasks when data flows between them. A task reads in some number of data products and generates some number of data products. Data products are assumed to be read and written to the workspace repository. We have identified four kinds of workloads: canonical, educational, ensemble, and data access as follows. Note that the number of file writes varies between 4 and 100 for each task in the workflow. While the number of reads and writes by a single task can more accurately be modeled by using a random number ranging between 4 and 100 for each read/write, to simplify the study and keep it deterministic, we use the *mean*. For instance, the 1-task workflow consumes and generates 52 data products each.

- 1) **Canonical workflow** is a 48 hour forecast model run that takes many (12) hours to complete. Based on our observation, the average number of tasks in a canonical workflow can vary between 4 and 12. The number of data products generated or consumed by a node in the canonical workflow can range between 4 and 100. These are captured in Table 1 as variables “*p*” and “*t*” respectively. These variables are used to derive the equation for I/O rate. The derivation of the equation is described by Figure 1. The simplest canonical workflow consists of one task that reads between 4 and 100 data products from the workspace and writes between 4 and 100 data products to the workspace. This is labeled “1-node workflow. For a 2-task workflow (labeled “2-node workflow”), the second task adds an additional write of between 4 and 100 data products. Note that the second task does not read products from the workspace repository because it is assumed those products are in a scratch disk close to the where the workflow is running and can be reused. The general case of the *n-task* workflow is labeled “n-node workflow”.

- 2) **Educational workflow** is a simplified workflow of 4 tasks. The number of inputs and outputs to each node is estimated at 4 inputs and 4 outputs per node.
- 3) **Ensemble workflow** is a canonical workflow multiplied times 100. The ‘100’ is assumed to be a mean as well.
- 4) **Data access workflow** can be viewed as a simple 1-task workflow. An example is to issue a query to gather data products into a single place for an experimental run. We estimate the average number of files retrieved/written during this operation (and hence written to the metadata catalog and storage repository) to be 30 (15 reads/15 writes).

Table 1. Workload types and distribution of users in four categories. The number of workflows (500) is derived from 125 users executing 4 workflows each during a 12 hour interval.

Workload Type	Data products (p) read or written to repository per node	Functional tasks (t) per workflow	% total users running this kind of ‘job’ at any one time	Number of workflows by type executing during any 12 hour interval (500 total)
Canonical Workflow	$4 \leq p \leq 100$	$4 \leq t \leq 12$.10	50
Educational Workflow	$p \approx 4$	$t \approx 4$.50	250
Ensemble Workflow	$(4 \leq p \leq 100) * 100$	$(4 \leq t \leq 12) * 100$.01	4
Data Access Workflow	$p = 30$ (15 read, 15 write)	$t = 1$.39	195

We estimate that the LEAD grid will have **125 users**. Since an instance of the canonical workflow takes about 12 hours to complete, we use 12 hours as the smallest atomic time interval, and compute the activity within a 12-hour block. By taking a sufficiently large enough block, one can assume a uniform distribution of I/O activity and safely compute averages. While we could spread that activity over 24 hours, we simplify things by clustering activity during the “daytime” period, and assume users are quiescent during the 12 hour “nighttime” period.

The mix of jobs on the LEAD grid is estimated to be made up of 10% canonical workflow, 50% educational workflow, 1% ensemble workflow, and 39% data access workload. That is, over any 12-hour period, the 125 users will be actively executing jobs, with each user executing 4 jobs during that 12 hour period. Thus, within a 12-hour block, 12 users will each run 4 instances of a canonical workflow, 62 educational users will run 4 instances each of the educational workflow, 1 user will run 4 instances of the ensemble workflow, and 48 users will run 4 instances of the data access workload. Total number of jobs is not exactly equal to 500 due to rounding.

2 File Size Distribution

Before we can calculate read/write bandwidth and storage needs of the workspace, we need a file size distribution that tells us the frequency of occurrence of files at various sizes. We base this on key community datasets that have been identified as such in the LEAD project. From inspection of Table 2, we can see that data products range in size between 1 KB and 50 MB. Note that all but “Eta” data are observational data products, that is, generated by instruments that directly

measure the atmosphere. “Eta”, on the other hand, is a model generated data product. “CAPS” are the new shorter range CASA Doppler data products being developed as part of the NSF funded CASA Engineering Research Center (www.casa.umass.edu).

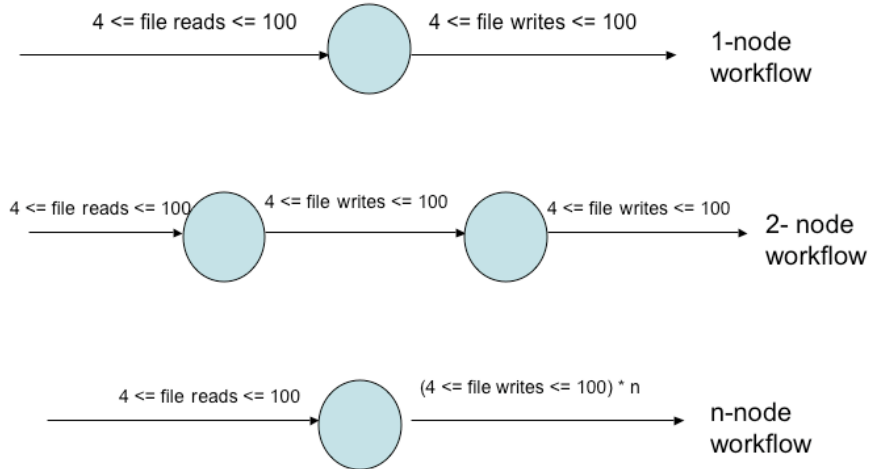


Figure 1. Read and write behavior of a 1-, 2-, and n-task workflow. Intermediate tasks/nodes incur no read from the workspace repository since it is assumed the data products will be cached at the computational nodes. Intermediate products, are, however, stored to the workspace.

From the file sizes of Table 2 (where we assume events are stored one per file), we computed a size distribution (shown in Table 3). This table should be read to mean, for instance, that 50% of all writes of files are of 62KB files. In a big network such as a grid, WAN, or Internet, file reads and writes are frequently done using the upload/download model, that is, where whole files are read and written. The upload/download model does not support retrieval of file chunks, nor random access to chunks.

Data Source	No. sources	Ev. Size (KB)	Ev. Rate (ev/hr)	Cum. Rate (event/hr)	Cum. BW (Kbps)
Metars 1st order	27	1-5	3	81	0.9
Metars 2nd order	100	105	1	100	1.1
Rawinsondes	9	2125	0.08	0.75	0.04
Acars	30	100-700	10	300	466.67
NexRad II	5	163-1700	6-12	60	222.2
NexRad III	5	2-20	6-12	60	2.67
GOES	1	4400	2	2	19.6
Eta	4	41500	0.17	0.67	615
CAPS	10	62.5-15.6	12-60	600	20800

Table 2. Data product sources in 80-mile region surrounding New Orleans. In the region, there are 27 sources of METAR 1st order data, for example where each timestamped measurement is between 1 and 5 KB and is generated every 20 minutes. The cumulative generation rate computed for all 27 sources is 81 events per hour.

While the file distribution of community data products is a good start, it is only part of the picture of file distribution for two reasons: the workspace repository is driven by workflow execution, and second, because community data products are already commonly available. A file distribution for a workspace will be split across ingested community products, model generated data products, and visualization and analysis products. Their sizes and estimated frequency are given in Table 4. A file size distribution that reflects the weightings given in Table 4 is shown in Table 5. Table 4 is new to V2.0 of the technical report. The new distribution we believe more accurately reflects the file size distribution found in a workspace. As can be seen in Table 5, the larger model generated data products (41MB) now represents 50% of the workload, as opposed to 1% of community data products.

File Size	Frequency
5 KB	6.7 %
20 KB	5.0 %
62 KB	50.0 %
100 KB	8.0 %
700 KB	25.0%
1,700 KB	5.0 %
41,000 KB	1.0 %

Table 3. Distribution of file sizes for community (LDM ingested) data products.

Data Product type	<i>Ingested community products</i>	<i>Model products</i>	<i>Images, analysis results</i>
File size	<i>Follows Table 3 distribution</i>	<i>40MB</i>	<i>2MB</i>
Frequency (weighting) of product in workflow activity	<i>15%</i>	<i>50%</i>	<i>35%</i>

Table 4. Frequency of product type. Observational products make up only 15% of the total workspace because these products reside at community archives.

File Size	Frequency
5 KB	0.1 %
20 KB	0.8 %
62 KB	7.5 %
100 KB	1.2 %
700 KB	3.7 %
1,700 KB	36.0 %
41,000 KB	50.0 %

Table 5. Distribution of file sizes for products stored to the LEAD workspace. Reflects the frequency of product type from Table 4. Image and analysis results of 2MB are rounded to 1.7MB here to simplify comparison between Table 3 and Table 5.

3 Performance and Storage Estimates

Up to this point in the study, we have quantified the characteristics of the LEAD portal workload. We generalized the tasks a user may carry out as falling into one of four categories, estimated the anticipated number of users, the distribution of users across the task categories, and the size and distribution of files used in the tasks. We further distinguished between the portions of a user's space that will be accessed frequently from the portion accessed less frequently by discussing the notion of an 'active' and 'historical' snapshot. These estimates are sufficient to approximate I/O rate and sustained bandwidth needs to the storage server.

As discussed in Section 1, we model activity over a 12-hour block, and assume a particular fixed workload. Specifically, in any 12-hour block each user will execute four workloads. Thus, within any 12-hour block, 12 users will run 4 instances of a canonical workflow, 62 educational users will run 4 instances of the educational workflow, 1 user will run 4 instances of the ensemble workflow, and 48 users will run 4 instances of the data access workload. Total number of jobs is not exactly equal to 500 due to rounding.

3.1 Read/Write Rate

Read/write activity is a measure of the file reads and writes to/from the storage repository. The assumed interaction with the storage repository is by an upload/download model of interaction (such as through FTP, gridFTP, or HTTP), where the file is the atomic unit that is read or written. Where estimates of the number of data products a task generated are given in Table 1 as a range, we remind the reader that we use the mean. For instance, in the canonical workflow case, the number of data products written per workflow node is estimated to be between 4 and 100. The mean is 52.

The reads and writes over a 12 hour period are calculated as follows:

Canonical writes = 52 writes (p) * 8 tasks (t) * 50 jobs	= 20,800 writes/12 hr
Canonical reads = 52 reads (p) * 1 tasks (t) * 50 jobs	= 2,600 reads/12 hr
Educ writes = 4 writes (p) * 4 tasks (t) * 250 jobs	= 4,000 writes/12 hr
Educ reads = 1 read (p) * 1 tasks (t) * 250 jobs	= 250 reads/12 hr
Ensemble writes = 52 writes (p) * 8 tasks (t) * 100 versions * 4 jobs	= 166,400 writes/12 hr
Ensemble reads = 52 reads (p) * 1 tasks (t) * 100 versions * 4 jobs	= 20,800 reads/12 hr
Data access writes = 15 writes (p) * 195 jobs	= 2,925 writes/12 hr
Data access reads = 15 reads (p) * 195 jobs	= 2,925 reads/12 hr

Total reads/writes per 12 hours = 18,391 reads-writes/hr

File read/write rate = 5 files read/written per min
--

3.2 Sustained Bandwidth

Sustained bandwidth, measured in megabits per second, is a measure of bandwidth consumption imposed by the workload on the storage repository. Sustained bandwidth utilizes the file level read/write rates calculated in Section 4.1 and the file sizes and distribution from Table 4.

Read/write bandwidth utilization is measured as follows:

5KB files:	$18391 \text{ file rw/hr } (.01) * 5KB$	$= 920 \text{ KB/hr}$
20KB files:	$18391 \text{ file rw/hr } (.008) * 20KB$	$= 2,942 \text{ KB/hr}$
62KB files:	$18391 \text{ file rw/hr } (.075) * 62KB$	$= 82,518 \text{ KB/hr}$
100KB files:	$18391 \text{ file rw/hr } (.012) * 100KB$	$= 22,069 \text{ KB/hr}$
700KB files:	$18391 \text{ file rw/hr } (.037) * 700KB$	$= 476,326 \text{ KB/hr}$
1700KB files:	$18391 \text{ file rw/hr } (.36) * 1700KB$	$= 11,255,292 \text{ KB/hr}$
41000KB files:	$18391 \text{ file rw/hr } (.50) * 41000KB$	$= 377,015,500 \text{ KB/hr}$

Total read/write bandwidth = 388,855,567 KB/hr,
which is approximately 389 GB/hr and 4,668 GB/day (considering active 12 hr period followed by quiescent 12 hours)

Total sustained read/write bandwidth = 864 Mbps

3.3 Spinning Disk and Archival Storage Needs

Storage requirements for the workspace consist of fast-access to products of immediate need (such as recently immediate results are more difficult to predict we do not know the rate at which users will remove them. This can only be measured by directly observing users.

We assume that of all the activity carried out over a 12 hr period, 20% of the data written to disk is deleted by the user. This could be due to incomplete runs or results determined to not hold relevance to a particular study. We also assume that spinning disk will contain the latest 1 week of data for all users, and another 20% on top of that for caching results off archival storage.

$GB \text{ written in 1-12 hr interval} = 4,656 \text{ GB/12 hr} - 931 \text{ GB/12 hr} = 3,725 \text{ GB/12 hr}$
 $GB \text{ written in 7 days (7-12 hr intervals)} + 20\% \text{ cache} = 26,075 \text{ GB} * 5,215 \text{ GB} = 31,290 \text{ GB}$

Spinning Disk Needs = 31.2 TB (1 week plus 20% warm cache)

$GB \text{ written in 1week} = 26 \text{ TB}$
 $GB \text{ written in 3 months} = 312 \text{ TB}$

Archival Storage Needs = 312 TB (3 months) or 1.2 PB (1 year)

4.0 Estimated Monthly Storage and Use Cost

Amazon S3 claims to be the “storage for the Internet”. It has a simple web service interface,

minimal feature set, with a focus on simplicity and robustness. It provides access control and authentication. Amazon S3 pricing is based on usage of storage space and network traffic. Its pricing policy* is as follows: **Storage used:** \$0.15 per GB-Month of storage. This fee applies to all object data and metadata stored in buckets under one account. **Network data transferred:** \$0.20 per GB of data transferred. This fee applies anytime data is read or written to a user's file space.

Using the figures calculated earlier in this report, the total number of bytes read/written to/from the workspace is 4.668 GB/day or 140 TB/mo (from Section 3.2). At 20 cents a GB read-written, the total charge for network data transferred is \$28,000/mo. Disk usage cost is based on the 1,248 TB sustained storage needed for 1 year of data (from Section 3.3). At 15 cents a GB storage cost, the total monthly cost for disk storage is \$187,200.

The total monthly charge for maintaining a LEAD workspace on Amazon S3 for 125 users executing 500 activities on the LEAD grid (including things like simple data searches) in any 12 hour daytime period is \$215,200 per month.

Monthly cost = \$28,000 (bandwidth) + \$187,200 (disk) = \$215,200/month

* Pricing on 03/01/2006 as quoted in the Amazon S3 Developer Guide (API Version 2006-03-01)

5.0 Conclusion

We have undertaken a study to characterize the use of the LEAD grid a few years out, and use this characterization to estimate the needs on the LEAD grid resources (i.e., back end storage and network). The results presented here show that the storage needs can be met by current technology, but LEAD will need to seek resources beyond its current resource base to satisfy the needs.

Were the LEAD Workspace to use Amazon S3 for its storage, what would be the impact on myLEAD? We maintain that the myLEAD metadata catalog must still remain because metadata and search in Amazon S3 is limited. Amazon S3 maintains system metadata and user metadata. The metadata for a data product, however, is limited to 2KB in size. This is small. All user level metadata must be added at time of product creation. myLEAD, on the other hand, allows annotation after-the-fact, and in fact this is one of its strengths. User level metadata is not visible (interpretable) by S3. This means that search on such attributes as spatial area over which a model was computed, or grid spacing used, cannot be done if there didn't exist a metadata catalog outside S3. The metadata catalog outside S3 would certainly want to use the S3 metadata to its fullest extent so as to not duplicate functionality. Search in S3 is limited to prefix filtering on object names.

6.0 Acknowledgement

My sincere thanks to Sangmi Lee, postdoc at IU, for her early thoughts and discussions on this problem, to Nithya Vijayakumar, PhD candidate at IU, for her characterizations of the stream rates shown in Table 1, to Anne Wilson of Unidata for capturing the details of the ingest data products early on in the project. My thanks also to Suresh Marru, research scientist at IU for his

help in characterizing the workflows and Yogesh Simmhan, PhD candidate, for his careful reading and edits.