DKRZ Workload Analysis

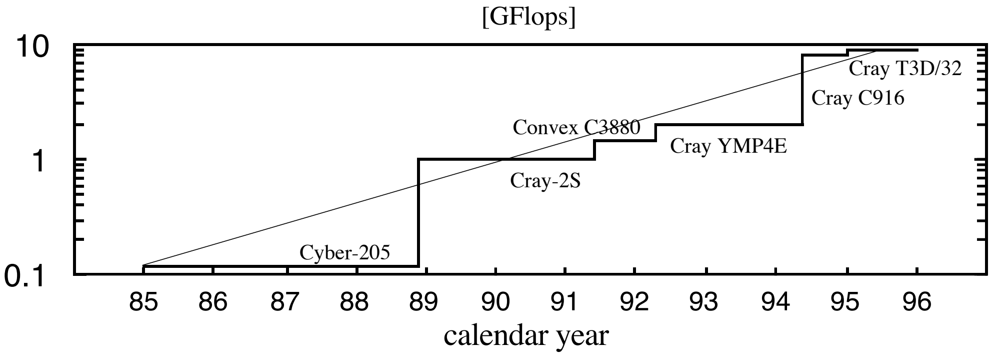
Hartmut Fichtel
Deutsches Klimarechenzentrum GmbH
Bundesstr. 55
D-20146 Hamburg
Germany
e-mail: fichtel@dkrz.de
phone: +49(40)41173 220
fax: +49(40)41173 174

Introduction

DKRZ is a primarily federally funded institution to provide the German climate research community with the necessary computing resources to perform numerically highly complex climate simulations. This task implies a virtually unlimited requirement for installed compute power at the outset; moreover, directly correlated with available compute power are the corresponding data services requirements, both in terms of static capacity and dynamic data access.

The following diagram shows the development of installed compute power at DKRZ from 1985 to 1996.

CPU performance available at DKRZ



This local development seems to be well synchronized with the overall technological development of supercomputer technology showing an average increase of almost a factor of two every 2 years.

Tightly coupled to the compute power dedicated to climate simulations is the data rate in terms of long term storage. As a general rule, it turns out that **1 Flops compute power** (sustained) generates **1 KByte of data** per year which is of sufficient interest to justify long term storage. So, the above diagram can also be interpreted as the development of actual annual long term storage rates if the y-axis is replaced by [TByte].

There are two more interesting rules of thumb defining the overall correlation between available compute cycles and the resulting requirements for the data management system which have been remarkably constant in the past. The second rule refers to data generation rates as opposed to long term storage: roughly 25 % of the total data generated will be stored for less than a year and thus does not contribute to long term storage requirements. These rates refer to climate model output only which is the dominating sort of data at DKRZ. Other types (e.g. observational data for model validation, general system backups etc.) display different characteristics. The third rule defines the resulting overall data access requirements: for every byte generated there will be 3 bytes accessed, so model data turns out to be relatively active.

With these three rules not changing and the expectation that continuous funding will be available, it is rather easy to estimate the future requirements for the data management system which is summarized in the following table. Actually, the data intensity will partially change: the major existing applications double their storage intensity by decreasing the archival interval from 12 to 6 hours simulation time. Other uncertainties result from new applications, e.g., atmospheric chemistry.

	95	96	97	98	99	2000
CPU performance [GFlops	9	9	20	20	35	50
data generation rate [GByte/day]	40	80	150	150	300	400
data archival rate [TByte/year]	10	20	40	40	80	100
required data archival capacity [TByte]	20	40	80	120	200	300
average data moved [GByte/day]	160	320	600	600	1200	1600
average transfer rate [MByte/s]	1.6	3.2	6	6	12	16
required peak transfer rate [MByte/s]	16	32	60	60	120	160

The immediate conclusions are that current technology supports both the required robotic capacities and transfer rates for disk and tape devices. On the other hand, it has to be noted that current technology fails for required network performance and in particular for the aggregate performance of the existing migration software.

Detailed Statistics

Reasonable estimates for future data management system requirements can be made from some global parameters and the rules described if the applications are well known and do not change considerably over time. These facts are insufficient, though, both to define the detailed structure needed for future systems and to identify possible bottlenecks in existing ones. More details are needed for these purposes.

Central file service at DKRZ is based on the well known UniTree product running on a Convex C3800 hardware platform. The built-in statistics and logging features of standard UniTree are relatively poorly developed, so some additional instrumenting code has been added to generate timestamped event-driven messages to analyze the dynamic behavior of the system. About 8 MByte of raw data thus generated per day is preprocessed to combine the various event-driven messages into a single transaction record resulting in 2 additional MByte per day of compressed transaction information.

The logging and analysis concentrates on these three key areas of the system:

- client interface via the network
- disk cache
- tape devices and robotic systems

Client interface

The client interface for normal data access is restricted to **ftp** since both **rcp** and **nfs** (besides the well known performance and security deficiencies of **nfs**) lack the ability to adequately control the access to the Unitree families which is required to choose the proper tape technology typically as a function of file size and possibly other file attributes. So it was decided to modify **uftpd** to log the major transaction events along with the relevant parameters:

- start of transfer and possibly cache hit
- stage to disk if necessary
- end of transfer

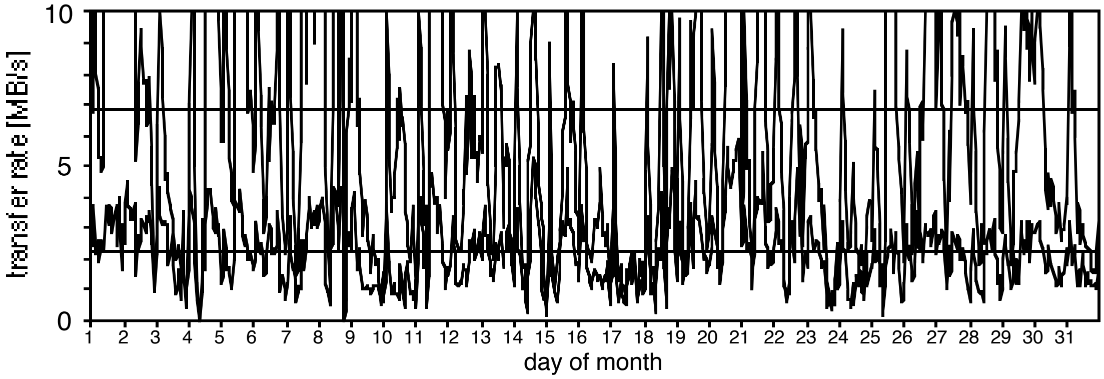
The relevant parameters additional to the time stamp are

- ftp operation code
- path name, capability, user and group id
- family, number of copies, host and network
- file size and transfer times (disk and possibly tape)

This collected information allows a variety of different parametric statistics like number of requests and data transferred per given time interval with associated variances. An example is the following diagram which shows the **ftp** access pattern with the number of parallel transfers and the effective transfer rate averaged over one-hour intervals.

Aggregate ftp transfer rate [MB/s]

(May 1996)



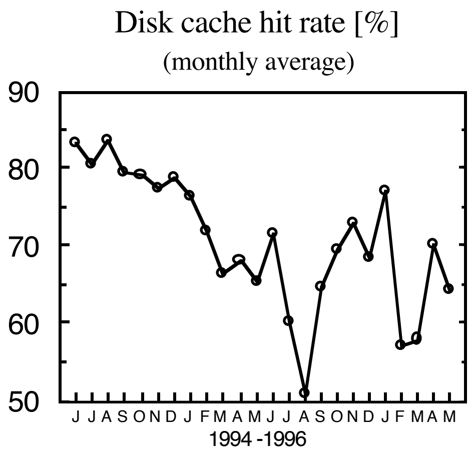
The 2 straight lines are the averages per month with 6.8 parallel accesses and a total of 5.7 TB transferred by 116K requests. If the time interval is shortened to ever smaller values it can be expected that due to the bursty nature of the traffic particularly in prime times the observed peak rates will reach a maximum either defined by the maximum load or the performance limit of the system. The current hardware platform consists of a Convex C3840 with 1 GByte central memory, 12 disk channels (IPI and SCSI), 14 tape channels

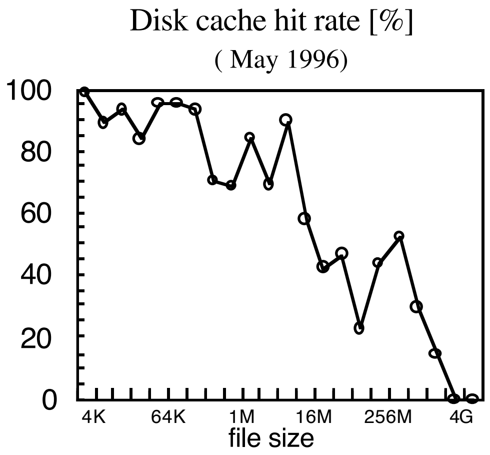
(BMX and SCSI) on the server side. HiPPI-switch connections to the major clients, and typically no disk bottlenecks on the client side. In this hardware environment the highest burst rate ever noted was around 16 MB/s which gives rise to the suspicion that the current overall bottleneck in terms of performance is induced by software overhead. This approach can generally be nicely used for benchmarks during the running system without shutting out user operations totally, e.g., to validate hardware or software upgrades and reconfigurations.

It should be noted though that averages over larger time intervals are only of limited value since the frequency distributions are far from normal. It is typical for mass storage systems that the number of transactions is dominated by small files, but the bulk of the data flowing is caused by a relatively small number of large transfers. The specific distribution of course is application dependent. Additional to plotting system parameters by file size it is worthwhile to also study other distributions, e.g. by host and network.

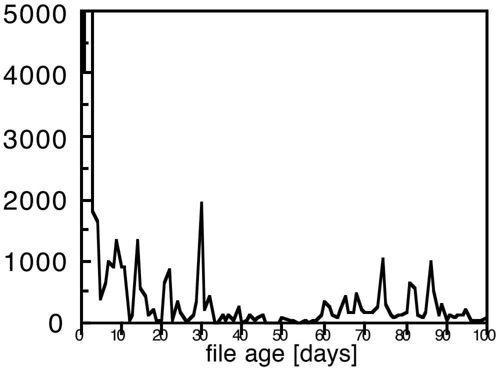
Disk cache

Probably the most important factor to performance is the disk cache since the cache hit pattern dominates the turnaround time as seen by client access requests. The overall disk cache hit rate development over time for the last 2 years is plotted in the following diagram. Due to the increasing load on the system this global cache hit rate has been decreasing almost monotonically from 85% down to 50% over the course of a year which tendency could be reversed by a major disk cache upgrade in mid 1995. In two steps 48 Elite-9 SCSI disk drives with 8 SCSI channels have been added to the existing IPI devices resulting in a total disk capacity of 550 GByte. It is interesting to note that this hardware upgrade did not take immediate effect but took several months instead. The reason is that it is easy to fill up the cache immediately, but it takes very long to put the files on disk which have the most positive influence on the overall hit rate, i.e. as many small active files as possible. This is also the explanation for the backward developments e.g. in February 1996: due to drive failures the contents of complete partitions have been lost with a resulting loss of a large number of small file copies.

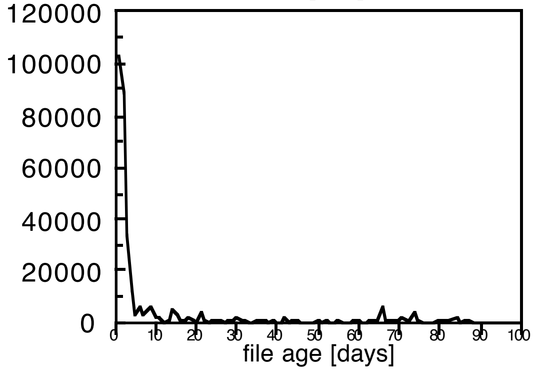




Although the global hit rate has decreased to 65% the situation is still much better for small files as the hit rate versus file size distribution shows. The next 2 diagrams show a snapshot of the cache file distribution (both number of files and aggregate size as function of age). Although the Unitree disk cache purging algorithm is not very versatile it turns out that the old files do not add considerably to the total size since they tend to be small.



Disk cache distribution
number of files



Disk cache distribution
size [MB]

Another valuable information to derive from the transactions logged with the capability identifying single files is the working set which denotes the locality of the user load.

Archival storage

The core of the mass storage system is the archival storage typically consisting of tape devices integrated into large robotic capacities. Tape technologies differ considerably in their technological parameters and cost, and furthermore the tape handling software in question may or may not take advantage of the various features they provide. It is thus not sufficient to compare different tape technologies just by their published features or even by comparative tests under the native operating system, since mass storage systems often handle tape devices independently and in different fashion. In particular it is not sufficient to compare streaming transfer rates because for realistic file distributions transfer is not the only fraction of the operation contributing to elapsed time, and it may not even be the dominating one.

Currently there are 2 robotic systems and 3 tape technologies in production use at the DKRZ: STK 4400 libraries with both 3490 and D3/Redwood drives as well as Metrum RSS-600 with VHS drives. In order to evaluate this technology logging information from **tapesrvr** and **pdmsrvr** define the various events constituting a complete tape transaction:

- receipt of request from **disksrvr**, possibly wait time in queue
- mount request issued to **pdmsrvr**
- tape positioned to requested block offset
- transfer complete, possibly „lazy wait“ on drive
- dismount request issued to **pdmsrvr**
- tape dismounted

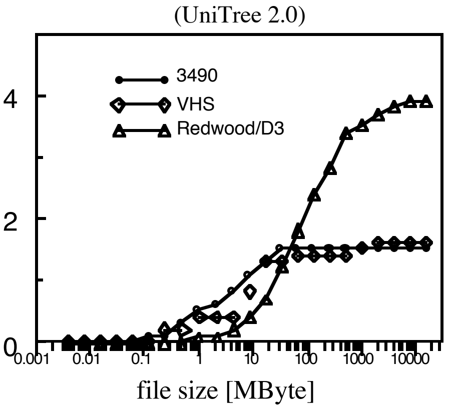
The parameters additional to the timing information of the various events listed above include:

- stage or mig request
- client or repacker request
- capability
- family, location, copy, and possibly fragment
- VSN, tapetype, physical unit
- block offset
- file size

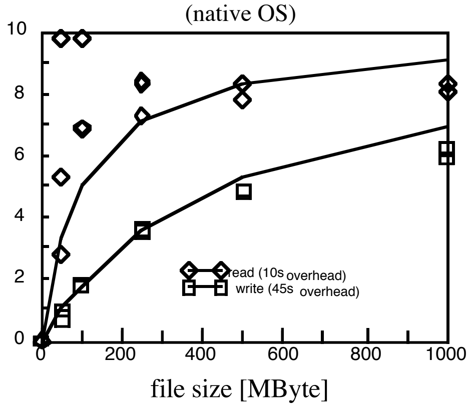
The analysis of this device oriented information enables a wide range of useful statistics, like the impact of internal reorganization (repacking) on overall system performance for client initiated requests, device specific performance for mount/dismount, positioning and transfer operations, queuing states for available devices, effects of system parameters like the „lazy wait“ feature on mount/dismount overhead, and many more.

An example is the following diagram which shows tape transfer performance as measured in the running Unitree system for 3490, VHS, and Redwood/D3 tape technologies as a function of request size. It turns out that VHS is relatively performing best by reaching 80% of the streaming rate already for moderately sized files of 30-40 MByte, whereas 3490 does not exceed about 50% of the peak rate. Relatively disappointing is D3 performance which saturates at about 40% of the streaming rate needing even larger files in the GByte range. It is assumed that the reason for this poor performance again is caused by the inefficient server-server communication scheme since similar tests under the native operating system show that the D3 streaming rate of almost 11 MByte/s can indeed be reached for files of 1 GByte and more.

Device transfer rates [MByte/s]



D3 transfer rates



The different read/write behavior is caused by the overhead of tapemark processing which is higher for writing files. The measured write results follow the theoretical curve very closely, the read results for small files are perturbed by the read-ahead capability of the D3 device, a feature which cannot be typically made use in a mass storage system.

Conclusions

The UniTree product as it is released emits a rather large amount of logging information into various log files, but this data typically is meant for operator information in difficult operational situations or directly for debugging purposes. It is strongly advised that the existing logging functionality in the standard release is advanced by adding messages with relevant parameters for every major event in the course of executing file or device oriented requests as described.

This upgrade would be relatively easy in terms of implementation effort. The benefit is a complete sequence of transaction descriptions generated by external user requests or by internal administration commands. This collection of transaction records can be used for intensive statistics and performance evaluations. It is furthermore a perfect source to drive realistic system simulations to study the effects of possible hardware or software changes.

Acknowledgments

This project has been initiated and in most parts designed by DKRZ. The implementation of actual changes in the UniTree server code has been performed by Ingolf Sessler, a consultant to the Hamburg office of Convex Computer. We appreciate the open and ongoing cooperation.