NHT-1 I/O Benchmarks

Russell Carter[1], Bob Ciotti[1],
Sam Fineberg[2], Bill Nitzberg[2]

Report RND-92-016 November 1992

NAS Systems Development Branch
NAS Systems Division
NASA Ames Research Center
Mail Stop 258-6
Moffett Field, CA 94035-1000

Abstract.
The NHT-1 benchmarks are a set of three scalable I/O benchmarks suitable for evaluating the I/O subsystems of high performance distributed memory computer systems. The benchmarks test application I/O, maximum sustained disk I/O, and maximum sustained network I/O. Sample codes are available which implement the benchmarks.

[1] The author is a member of the NAS Systems Development Branch.

[2] The author is an employee of Computer Sciences Corporation.

This work was supported through NASA contract NAS 2-12961.

1.0 Introduction

High throughput network and disk I/O systems are critically important components of high performance highly parallel systems capable of supporting the complex scientific workloads found at national supercomputing centers such as NASA Ames Research Center's Numerical Aerodynamic Simulation (NAS) Facility. These workloads are composed of large scale computational fluid dynamics applications which require significant amounts of I/O. Maintaining acceptable system throughput while processing such a workload has required the development of sophisticated operating system features: virtual memory, checkpointing, copy-on-write techniques, etc. Such systems require large amounts of I/O capability that must be available to both user applications and system processes in order to maintain sufficient I/O throughput. This document describes three simple, scalable I/O tests that are applicable to high performance highly parallel system architectures. Large scale distributed memory computer systems are the target architectures; the design is intended to allow easy adaptability to shared memory architectures as well. These tests provide metrics for user application I/O, peak disk I/O performance, and network I/O.

The non-volatile secondary mass storage devices that are intended to be tested have historically been based on rotating magnetic storage devices (disks). This may not be true in future highly parallel computer systems. Hence, in the following, the term "disk" is intended to denote any local, non-volatile, read-many, write-many, secondary storage device that is supplied as a standard subsystem to the highly parallel computer system on which these benchmarks are to be run.

1.1 Application I/O Benchmark

Many unsteady Navier-Stokes computer codes commonly in use at NASA Ames write solution files to disk at regular predetermined intervals. The first part of the parallel I/O benchmark is intended to test the ability of the system to simultaneously support significant computation and I/O.

1.2 Disk I/O Peak Performance Benchmark

This benchmark is intended to measure the maximum sustained disk I/O transfer rate available to the system.

1.3 Network I/O Benchmark

This benchmark is intended to measure the maximum sustained network I/O transfer rate available to the system.

2.0 Application I/O Benchmark Specification

2.1 Background

I/O is a necessary component of the numerical simulation of unsteady flows. Typically, unsteady flow solvers iterate for a predetermined number of steps. Due to the large amount of data in a solution set at each step, the solution files are written intermittently to reduce I/O bandwidth requirements for the initial storage as well as for future post-processing. This benchmark simulates the I/O required by a pseudo-time stepping flow solver.

The *Application I/O Benchmark* implements the approximate factorization algorithm (hereafter denoted the *Approximate Factorization Bench-*

mark) precisely as described in Section 4.7.1 of The NAS Parallel Benchmarks [1], with additions described below. The specification is intended to conform to the "paper and pencil" format promulgated in the NAS Parallel Benchmarks, and in particular the Benchmark Rules as described in Section 1.2 of [1].

The choice of the Approximate Factorization Benchmark as the basis for the Application I/O Benchmark was determined by the following considerations:

1. The layout of the data to be written to disk must be roughly similar to the layout required for highest performance of a user calculation on a highly parallel system, otherwise the relevant data must be rearranged. Significant data layout rearrangements on a distributed memory highly parallel system typically require substantial amounts of wall clock time relative to the computation time. This in practice affects the overall observed I/O transfer rate, and degrades the performance of the application as a whole. The I/O transfer rate obtained from periodically writing the solution vector of the Approximate Factorization Benchmark represents an estimate of the practical transfer rate that may obtained by a NAS user code optimized for floating point performance. If significant data rearrangements are required to effect adequate I/O transfer rates, the additional overhead may result in increased runtime when compared to the runtime of the Approximate Factorization Benchmark.

2. The Approximate Factorization Benchmark has been implemented on a number of highly parallel systems, with comparatively high performance results [2]. Thus it is clear that the algorithm as specified in Section 4.7.1 of [1] is an appropriate numerical algorithm to implement on highly parallel systems.

2.2 Instructions

Section 5.2 of [1] specifies that N_S iterations of the Approximate Factorization Algorithm be performed. The Application I/O Benchmark is to be performed with precisely the same specifications as the Approximate Factorization Benchmark, with the additional requirement that every I_w iterations, the solution vector U must be written to disk file(s). Relevant parameters for the Application I/O Benchmark, including I_w, are provided in Table 1 on page 4. I/O may be performed either synchronously or asynchronously with the computations. Performance on the Application I/O Benchmark is to be reported as three quantities: the elapsed wall clock time T_T, the computed I/O transfer rate R_{IO}, and the I/O overhead ζ. These quantities are described in detail below.

The specification of the Application I/O Benchmark is intended to facilitate the evaluation of the I/O subsystems as integrated with the processing elements. Hence no requirement is made for initial data layout, or method or order of transfer. In particular, it is permissible to sacrifice floating point performance for I/O performance. It is important to note, however, that the computation-only performance T_C will be taken to be the best verified time of the Approximate Factorization Benchmark.

2.3 File Format

The solution vector for each time step cannot be spread across multiple files. If multiple solution vectors are contained in a single file, they must be sequentially concatenated in the order they were generated. The file(s) must be standard sequential file(s) containing solution vectors stored in the canonical order for the language used. For Fortran, the array elements of U should be in the order specified by the ISO Fortran standard [3], as if U was written in a single Fortran write statement. For the C language, the array elements of U must be in standard C order. Depending on the language, U will be stored in either row major or column major order. Further, it must be possible to transfer the solution file(s) over the network to a uniprocessor system and post-process them sequentially without significant processing overhead for data rearrangement. Evidence should be supplied that indicates that the file(s) are in the specified format.

2.4 Reported Quantities

2.4.1 Elapsed Time

The elapsed wall clock time T_T is to be measured from the identical timing start point as specified for the Approximate Factorization Benchmark, to the larger of the time required to complete the file transfers, or the time to complete the computations. The time required to verify the accuracy of the output files generated is not to be included in the time reported for the Application I/O Benchmark. The elapsed time T_T is to be reported in seconds.

2.4.2 Computed I/O Transfer Rate

The computed I/O transfer rate R_{IO} is to be calculated from the following formula:

$$R_{IO} = \frac{(5w) \times (N_{\xi} N_{\eta} N_{\zeta}) \times N_S}{I_w T_T}$$

Here N_{ξ}, N_{η}, and N_{ζ} are the grid size dimensions and w is the word size of a data element in bytes, e.g., 4 or 8, and the remaining quantities were defined previously. The units of R_{IO} are bytes per second.

2.4.3 I/O Overhead

I/O overhead ζ is to be computed as follows:

$$\zeta = \frac{T_T}{T_C} - 1$$

The quantity T_C is the best verified run time in seconds for the Approximate Factorization Benchmark, for an identically sized benchmark run on an identically configured system.

2.5 Application I/O Benchmark Parameters

The following parameters are to be used when performing the Application I/O Benchmark:

TABLE 1. Application I/O Benchmark Parameters

N_S	N_{ξ}	N_{η}	N_{ζ}	I_w
200	102	102	102	5

2.6 Verification

The integrity of the data stored on disk is to be verified by the execution of a post-processing program that sequentially reads the file(s) described in Section 2.3 and writes an ASCII formatted file of floating point numbers on the standard UNIX output stream *stdout*. These numbers are the 5 elements of each entry on the main diagonal of the three-dimensional, $(N_{\xi} \times N_{\eta} \times N_{\zeta})$ -entry solution vector. These numbers are to be written in order from lowest array index to highest array index. The numbers are to be formatted equivalent to the FORTRAN 77 FORMAT specification of "F15.10"; one number per line of output. The verification program is to be executed on precisely one node of the system, or another appropriate system; no multiprocessing is allowed. The vendor should supply the file containing the

$$5 \times N_{\zeta} \times \frac{N_S}{I_W}$$

numbers, and the program source code used to generate the output directed into the file. An example implementation of the verification program is provided in the sample programs discussed in Section 5.

3.0 Disk I/O Peak Performance Benchmark

3.1 Background

Not only is application I/O performance important, but it is also vital that a system be able to deliver high I/O performance for system functions such as loading programs, swapping, paging, or user memory management, in a multiuser, multitasking environment. Required aggregate system I/O performance is significantly greater than that of an individual application. The Peak I/O benchmark is intended to measure the performance of this type of optimized I/O.

3.2 Instructions

There are two parts to the benchmark: a write test and a read test. Each test must read or write at least 80% of the system's random access memory. Before each test begins, this memory must be filled with random data. The write test must transfer the contents of this memory to disk, while the read test must transfer the data stored during the write test from disk to this memory.

For the write test, the elapsed time is measured from the beginning of the write test (before any files have been created or I/O initiated) until all 80% of the memory has been completely transferred to disk. At this point, it should be possible to turn off the system without affecting the integrity of the stored data—the stored data must fully reside on disk.

For the read test, the individual nodes must transfer precisely the same data that was stored during the write test (in the same order) from disk to the memory. Before the read test begins, the only copy of the stored data must be on disk. (It may be necessary to reboot the system to ensure that none of the data is cached or buffered on the system.) The elapsed time is measured from the beginning of the read test (before any files have been opened or I/O initiated) until all the data has been transferred to the memory.

3.3 Reported Quantities

The peak I/O rate R_{PEAK} is reported in bytes per second and will be calculated as follows:

$$R_{PEAK} = \frac{2\,(0.8 S_{MEM})}{T_{PREAD} + T_{PWRITE}}$$

In this equation, S_{MEM} is the total size of the system's random access memory in bytes, and T_{PREAD} is the wall clock time in seconds required for the peak read benchmark. This time consists of not only the time to read the data, but also the time to open and close any files in which the data is stored. Similarly T_{PWRITE} is the wall clock time required for the peak write benchmark in seconds, including the time to open and close any files.

4.0 Network I/O Benchmark

4.1 Background

The Network I/O Benchmark is intended to measure the performance of the external network I/O subsystems. This type of I/O is required to support user data transfers to remote storage, data communications typically required of heterogenous distributed applications, and data transfers required by distributed visualization applications.

4.2 Instructions

For this benchmark, the data written to disk as required in the Disk I/O Peak Performance Benchmark (Section 3 above) must be combined into a single system readable file. The local system's *ftp* (or *rcp*) command shall be used to transfer the single file containing the Disk I/O Peak Performance Benchmark data from the system's local mass storage system over the network through a loopback connection to a different file on the original local mass storage system. Evidence should be supplied that indicates that the transfer occurred over the loopback connection and that the file was transferred correctly.

4.3 Reported Quantity

The network I/O rate $R_{THROUGHPUT}$ is reported in bytes per second and will be calculated as follows:

$$R_{THROUGHPUT} = \frac{S_{FILE}}{T_{TRANSFER}}$$

In this equation, S_{FILE} is the total size of the file in bytes, and $T_{TRANSFER}$ is the wall clock time in seconds required to complete the file transfer. This time consists of the time to transfer the data over the loopback connection and the time to open, close, read, and write the files in which the data is stored.

5.0 Sample Programs

Sample programs implementing simple versions of the benchmarks, and the verification test, are available by mailing a request to NHT-1 Benchmarks, NAS Systems Development Branch, MS 258-5, Moffett Field, CA, 94035-1000. Email requests may be sent to rcarter@nas.nasa.gov.

6.0 Acknowledgments

The authors wish to thank Ian Stockdale, Toby Harness, and Eric Barszcz of the NAS Systems Division for their helpful comments and suggestions regarding this document.

7.0 References

[1] D. Bailey, J Barton, T. Lasinski, and H. Simon, eds. "The NAS Parallel Benchmarks, Revision 2." Technical Report RNR-91-002, NASA Ames Research Center, Moffett Field, CA 94035-1000, July 2, 1991.

[2] D. Bailey, E, Barszcz, L. Dagum, and H. Simon. "The NAS Parallel Benchmarks Results." Technical Report RNR-92-002, NASA Ames Research Center, Moffett Field, CA 94035-1000, August 31, 1992.

[3] International Standard Programming Language Fortran, ISO/IEC 1539: 1991 (E), ISO/IEC Copyright Office, Case Postale 56, CH-1211 Geneve 20, Switzerland, sections 6.2.2.2 and 9.4.2.