PDR RID Report

Date Last Modified 11/14/95

Originator Richard Rood

Organization GSFC

E Mail Address rood@dao.gsfc.nasa.gov

Document

Phone No 301-286-8203

RID ID **PDR** 394
Review SDPS
Originator Ref
Priority 2

Section **Page** **Figure Table**

Category Name System-level **Actionee** HAIS

Sub Category

Subject Benchmarks

Description of Problem or Suggestion:

It has been stated many times that various small parts of the subsystems have been benchmarked. A common characteristic of benchmarks is they test CPU capabilities of single components, but do not test interactions of either hardware and software systems. These interactions often become the rate-limiting step, and severely compromise the ability of hardware to meet their advertised performance. Another common characteristic of benchmarks is that they are somebody's favorite problem, and hence, are often not representative of generalized, cross-system problems. What effort has been made to assure the benchmarks are representative? How do we gain confidence in their appropriateness?

Originator's Recommendation

Brief descriptions of benchmarks and what they are designed to address.
Development of end-to-end benchmarks to understand integration and communications flow.
Use of standard benchmarks that are widely accepted as representative, and, where there is experience across a wide hardware and algorithm parameters.

GSFC Response by: **GSFC Response Date**

HAIS Response by: Eisenstein **HAIS Schedule** 5/22/95

HAIS R. E. Privor **HAIS Response Date** 6/28/95

Within ECS, the term "benchmark" is used to refer to performance measurement of specific hardware and software configurations. Performance analysis occurs in several stages, which are described below as hardware benchmarks, subsystem/functional benchmarks, and end-to-end system performance tests.

Hardware Benchmarks: Basic hardware benchmarking measures hardware components, include processing and I/O. Processor benchmarks include SpecInt92 (Integer), SpecFp92 (Floating Point), and Linpack 1000 (Linear Equations). Benchmarks of hardware performance using industry standard operations, and without actual developed software, are necessary but inadequate to properly size specific platforms. This sort of benchmarking is generally performed by hardware vendors and is not generally repeated by ECS.

Subsystem/Functional Benchmarks: Within ECS subsystems, engineers benchmark actual software components (COTS or developed code) to understand the true requirements for running specific applications on target hardware architectures. However, this sort of benchmark still tests specific, isolated functions within a subsystem, and is most often limited to a single platform. Subsystem benchmarks are typically developed by running candidate software in specific contexts that are appropriate to the overall end-system (e.g., using scenarios) and then measuring the resulting performance (processing, memory utilization, and I/O). The following is a representative list of areas in which we have analyzed subsystem performance relevant to Release A (either via measurement in the EDF, or by analyzing the results of performance tests done on other systems):

- FDDI switch/router performance, using various numbers and types of filters
- Throughput over NSI and V0 networks (various specific conditions considered)
- TCP and UDP (concurrent input and output streams) on various platforms over FDDI
- DCE / OODCE client and server performance (RPCs, message throughput and delays for IPC implementations, directory and security services)
- MSS COTS performance (HP Openview, Tivoli, DDTS, Remedy, Clearcase)
- Production Management COTS performance (Autosys)
- Examination of Pathfinder, AVHRR/Land processing using DCE on distributed workstation clusters
- Preliminary results of instrument team algorithm development, yielding data on RAM and disk utilization
- FSMS
- RAID
- Network attached storage
- Sybase, Illustra spatial searches
- Robots
- Sybase performance for gateway, interoperability prototypes

• Network attached storage
• Sybase, Illustra spatial searches
• Robots
• Sybase performance for gateway, interoperability prototypes

Measurement of performance in subsystem benchmarks may include CPU and RAM utilization, memory-to-disk I/O, memory-to-memory I/O, and transactions per second (e.g., for standard database functions). Stress testing (e.g., use of simultaneous sessions using layered software) is performed at this level as well. Early Sybase benchmarks examined the impact of multiple simultaneous queries to a representative database. MSS performance analyses address the performance of the MSS platform while multiple operator sessions are in process, using the layered COTS products to be placed on the platform. Representative benchmarks for sub-system functions are compared with published results, when possible (e.g, for DCE functions), though care must be taken to ensure that the test environments are comparable.

End to End System Performance Analysis and Measurement: The term "end to end" has no consistent definition, but generally refers to testing across multiple subsystem functions or components. Early in system development (e.g., prior to CDR), end to end -- or cross-subsystem -- prototypes are typically developed in order to confirm or test the viability of interface functionality. Examples of such prototypes relevant to Release A have included .the Prototype Workshop 1 (demonstrated data management's advertising and search capabilities), the Data Server "core" prototype (demonstrated Gateway to Data Server interface), and the later EPs (EP5 and EP6). Although these prototypes can be measured while in operation, they do not provide a complete understanding of "end to end" system performance since they typically are only prototypes. That is, they (1) may not include the full functionality of the components being tested; and (2) may use non-representative components for some parts of the system in order to achieve a given end-to-end product or demonstration. Nevertheless, these prototypes can provide insights into potential performance bottlenecks as well as functional interface issues.

Modeling -- both static and dynamic -- is also used to understand cross-subsystem performance, especially with respect to multiple processes running on a single platform (or cluster), as well as contention for network resources. Modeling of processing clusters, processing and archive storage, and processing to archive I/O (network) requirements has been performed using the BoNES simulation tool. Modeling of other subsytems has been performed using static (e.g., spreadsheet-based) analysis. Although static analysis may not show all system interactions, we have developed scenarios to look at worst-case performance. For networks, this includes analysis of push and pull traffic, including data and queries, and communications infrastructure traffic, including DCE and management (SNMP polls, transfer of log files) traffic. An "end to end" modeling approach (briefed at Release B IDR) is being used to develop a means of examining the total system under a number of operational circumstances. This approach involves developing a spreadsheet that captures benchmark and analytical results of key lower level operations on each host, and manipulating these by varying system loads in response to various scenarios.

In the post-CDR timeframe, I&T's build-thread testing provides insight to the performance of complete software components on target hardware platforms, as well as the interaction of components using the precise target infrastructure (network and DCE) configuration. The use of an actual DAAC configuration within the EDF provides assurance that test results are representative of expected results at the installed sites. Build-threads are bottom-up in the sense that they address infrastructure, common components, and subsystems before examining higher level system interactions.

In large government-procured systems, hardware specification and procurement generally precedes integration and test, introducing an element of risk related to incomplete performance testing of the overall system. One of the foremost issues is the development of a scalable hardware and network architecture that enables reconfiguration and expansion with the least possible impact to M&O. In general, ECS has used scaleable configurations (e.g., SMPs) to minimize the impact of increased processing needs, and hub-based networks (using multi-mode fiber) to minimize the impact of network reconfigurations and evolution to higher speed services (e.g., ATM).

Status **Closed** **Date Closed** **11/14/95** **Sponsor** **Herring**

****** **Attachment if any** ******