Introduction of the UNIX International Performance Management Work Group

Henry Newman
Instrumental Inc.
4500 Park Glen Road, Suite 390
Minneapolis, MN 55044
hsn@instrumental.com

The Performance Management Work Group (PMWG) was first convened four years ago, and its work is now out for public review. Both OSF and USL are implementing this work as are a number of companies. XOPEN and POSIX 1003.7 have agreed to accept the work after the public review has been completed. The following White Paper is an overview of this work, and describes the group's motivations and requirements.

Performance Management Activities Within UNIX International

UNIX International
Waterview Corporate Center, 20 Waterview Boulevard
Parsippany, NJ 07054
Phone: +1 201-263-8400, Fax: +1 201-263-8401

1. Introduction

The primary output of the UNIX International Work Group on Performance Measurement is a set of requirements and recommendations to UNIX International and UNIX System Laboratories for the development of standard performance measurement interfaces to the UNIX System. Requirements will be based on the collective, non-vendor specific needs for a standard performance architecture. Currently the lack of this standard causes undue porting and kernel additions by each UNIX System vendor as well as a great variety of approaches to gain the same basic performance insight into the system. Building tools to monitor, display, model, or predict performance or its trends is a frustrating and currently single vendor enterprise. By providing standard data structures, types of performance data gathered, and a common kernel interface to collect this data, the whole UNIX system vendor community along with the UNIX software vendors can develop performance tools which last more than UNIX release and work on multiple UNIX platforms.

Some of the PMWG findings may be in the form of recommendations rather than requirements as a mechanism to stimulate the creation of a common base technology for performance measurement or reporting that is more tool oriented and provides a rallying point rather than a rigid standard imposed on the UNIX system performance measurement, end-user system tuning, capacity planning, and benchmarking areas.

In summary, the requirements and recommendations of the UNIX International Work Group on Performance Measurement can be a driving force behind the advancement of UNIX system performance technology allowing the end-users of UNIX systems to better understand and answer questions such as: what system to buy, how to tune the system, when to upgrade the system, and when to move to a faster system.

2. Organizational Statement of UI Performance Management Work Group

It is our desire that the Performance Management Work Group be composed of a balanced team of performance professionals representing the users prospective, as well as the development prospective in the area of Performance Management. We have invited a number of system management as well as development professionals from a number of systems data centers,

large systems manufactures, small systems manufactures, performance analysis organizations, and the US government users community to join the UI Performance Management Work Group. We are pleased to have in attendance at our Work Group meetings, a number of user and development professionals representing a broad cross section of the UNIX industry.

It has also proved to be quite valuable to have in attendance at our UI Work Group meetings, the performance professionals from other organizations outside of the UNIX International community. The experience they bring to the team in the performance management research areas, as well as their desire to develop and adhere to proposed performance management standards, makes the results of our efforts more acceptable throughout the industry.

With this prospective of having developers, users, and a broad representation of UNIX interested professionals attending our UI Performance Management Work Group meetings, the following document is a consensus of our views for making proposals to UNIX International to include Performance Management functions into the UNIX System V Roadmap.

3. Statement of UI Performance Management Work Group

The objective of this work group is to examine the area of performance management as it pertains to the UNIX Operating System and to make recommendations on performance management to UNIX International and to UNIX System Laboratories. In addition, this organization will also exchange information and ideas regarding performance management, with other related groups in the UNIX industry including, but not limited to, the IEEE Posix 1003.7 Committees, the Open Software Foundation, and X/Open. In particular, our results shall be made available to these organizations.

3.1 Scope

This Performance Management Work Group will be concerned with defining requirements and standards for the collection, presentation and distribution of performance data in large-scale distributed systems. Here, "performance data" is defined to include:

1. Interval or sampled data describing hardware and software resource usage or times, either globally or by some logical entity

2. Count data representing system or applications queue lengths, events, and system resource states

3. Data representing execution traces of processors

4. Data notifying of events occurring at a system, subsystem, or application levels

A layered model of function and interfaces for acquisition and use of such data is shown in Figure 1 to further assist in the delineation of the scope of concerns for this Performance Management Work Group.

- *Measurement Application Layer*

The uppermost level of the model (layer 4) contains the application primitives and tools used to present currently captured and archival performance data to the end-user (or potentially, to an automated stand-in). These application implementations will be called Measurement Application Programs (MAPs).

- *Data Services Layer*

This level of the model (layer 3) is responsible for data simulation, archival data storage, management or services and resources required for distributed measurement access and

control, for measurement requesting, and for data transformations required for analysis and data recording.

- *Measurement Control Layer*

This layer of the model (layer 2) is responsible for managing the capture of data, including the synchronization, and for providing any necessary buffer or queue management for data assembled by the data capture mechanism. A portion of this layer and the next lower (data capture) layer may be functionally replicated in a subsystem or application for synchronized data collection from such entities.

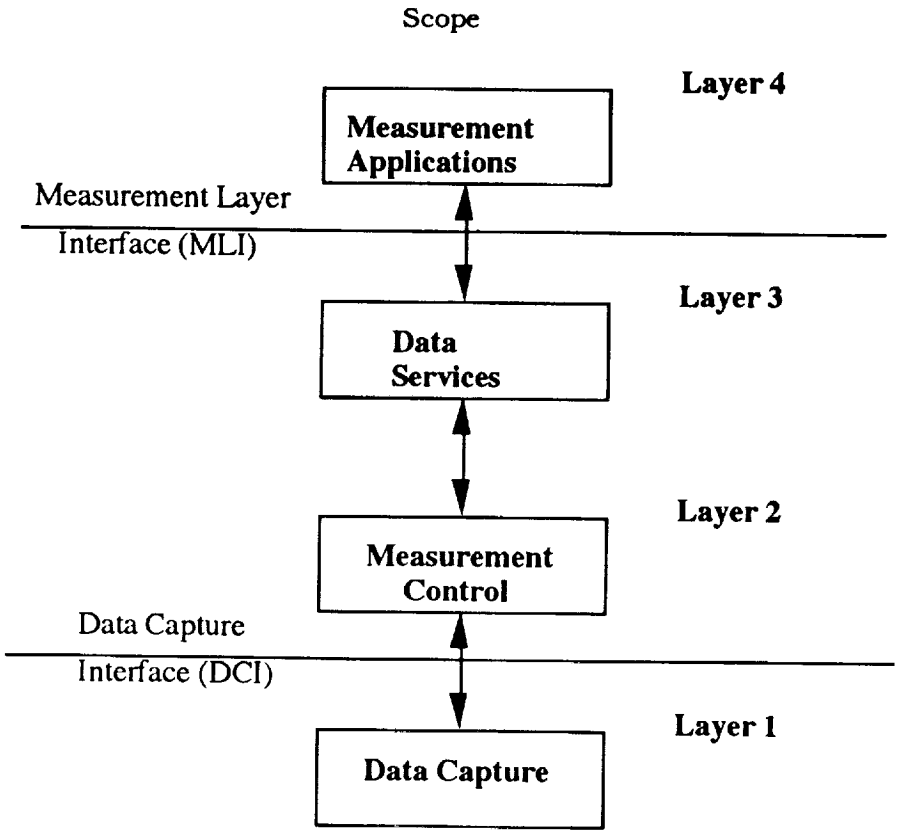

Figure 1. A Measurement Model for UNIX-Based Systems

- Data Capture Layer

This layer of the model (layer 1) is responsible for capture of data manifested in system or hardware counters or structures. Data is considered captured when it exists assembled into data structures of predefined class and type in storage controlled by services contained in the measurement model.

- Interfaces Defined by the Measurement Model

The interfaces between the layers are defined in a way that frees an upper layer from concern about how services are provided below it.

- The model provides a Measurement Layer Interface (MLI) for requesting measurement services. The MLI enables MAPs to be implemented without knowledge of the underlying measurement procedures.

- The model provides a Data Capture Interface (DCI) for access and initialization of low-level data sources such as the kernel data structures, hardware dependent data, and event generators.

This work group will focus on supporting the development of appropriate tools and services in the Measurement Application Layer (layer 4) through the establishment of requirements and standards for:

1. General services contained in the Data Capture, Measurement Control and Data Services Layers (layers 1-3) of the measurement model
2. Interface characteristics for Measurement Layer and Data Capture interfaces (MLI and DCI)
3. Measurement data classes, subclasses and, where appropriate, specific data item content of captured data

3.2. Relationship to Distributed Systems Management Frameworks

The measurement model just described fits naturally into USL's Distributed Management Framework (DMF) or other CORBA-based[1] environments such as OSF's DME. Thinking of objects in the measurement model's Measurement Application Layer as clients and thinking of objects in the model's Data Services Layer as service providers to measurement applications, we can note a number of administrative and functional benefits provided by these frameworks. These include:

1. The ability to configure the functionality of measurement service providers according to the requirements of their executing platform type and role as an entity in a measurement scenario (functional extensibility). Thus, for example, on one node that functions as a measurement data server, we might wish to configure a measurement server instance that includes the method for writing to a historical archive or performance data, but not do so at another node that functions only as a business application entity, that is, an object that provides performance metrics about itself.

2. The ability to identify and authenticate a measurement application program and its invoker, and to authorize access to appropriate measurement data services (security). For example, some measurement applications and users might have authorization to write to some specific database within a measurement data archive but not to others, or some users might have the authorization to see performance data concerning some business application(s) but not others, etc.

3. The ability to transparently operate distributed measurement applications, server objects, and Data Capture Layer collector objects across locations in a network. This implies, for instance, that a measurement application at one network location (a manager system) may request and receive data from a measurement server or managed system at another network location without having to directly establish contact with the remote provider. The measurement model formulates such access as a peer-to-peer communications between objects in the Data Services Layer.

4. Providing a repository for well-defined interfaces. For the measurement model these would be the Measurement Layer Interface (MLI) and the Data Capture.

Relationship to Distributed Systems Management Frameworks
Interface (DCI).

1 Common Object Broker Architecture, an approach for supporting distributed object-oriented applications formulated by OMG, the "Object Management Group".

4. Overview of Current Tools/Utilities

This section presents a brief overview of the current performance measurement tools available under the UNIX operating system. The two major implementations of the UNIX OS (UNIX System V, BSD) have different sets of tools but they provide similar types of data.

In UNIX System V, the performance tools are sar, sadc, timex, and login/process accounting (lastlogin, acctcom, acctcms, etc.). Data may be collected automatically and summarized with these tools. The BSD version of UNIX from U.C. Berkeley contains the following performance tools: vmstat, iostat, netstat, systat (in 4.3 BSD), and process/login accounting (sa, ac, lastcomm, last).

Login accounting provides information on when users logged into and out of the system. Process accounting is based on per process records that are written at process termination. It provides information on the duration of the process and the resources it used.

The system performance tools (sar, sadc, vmstat, etc.) sample counters and system state kept by the operating system to provide information on general system activity. The interval between samples may be specified, typically with a granularity of one second.

These tools have been useful in observing general system behavior and have been used in tuning existing systems as well as understanding the behavior of systems under development. However, they are not adequate for investigating/solving a variety of performance problems from diagnosing a current anomaly to system wide capacity planning. The following are the problems with the current implementations of performance data gathering utilities:

- Correlation: It is difficult to correlate data from different sources. For example, it is difficult/impossible to correlate process activity with any aspects of system behavior.
- Granularity: The granularity of data is often not correct. For example, process accounting records are only written at process termination. There is no mechanism to determine how the process behaved during its lifetime.
- Inextensibility: It is not possible to extend the data collected with these tools without the availability of source code. The non-standard methods for extending these tools is quite difficult.
- Data presentation: Typically inflexible and awkward to view.
- Lack of standard analysis tools

5. Background/Definitions

Most production (proprietary) operating systems (IBM/MVS, DEC/VMS etc.) provide an extensive array of tools for performance data gathering and management tools. UNIX Operating Systems currently provides a very limited set of tools and utilities for performance management. Performance information is widely used by developers (software, hardware, operating systems and applications), sales and marketing organizations. However, very limited performance data is available for commercial UNIX applications and production type UNIX computer (data) centers.

In this paper we define the types of performance data that are required and useful for analysis and management of computer systems. Architectures and implementations of performance data gathering tools, which are available in other proprietary operating systems are described. Finally, recommendations and requirements for implementation of tools in UNIX systems are proposed.

5.1 Performance Management Systems - Technology

5.1.1 Technology Overview - Large System Facilities

Currently, the most developed performance and accounting data management facilities for large-scale systems are to be found in proprietary operating systems such as IBM's MVS and DEC's VMS on its VAX computers.

In general, the modes of capturing data for either presentation as reports or subsequent use by other tools includes:

- Sampled Data - Data which is measured by repetitive capture (at the sampling rate) and presumably accumulated in a counter.
- Interval Data - Data which represents the *incremental* activity within a certain time interval.
- Event Data - Data which provides notification of the occurrence of a particular state within a subsystem.
- Trace Data - Data which captures a succession of subsystem states, usually in substantial detail.

IBM's MVS provides selectable recording of accounting and performance data through SMF (System Management FAcilities) extended by high resolution performance data through RMF (Resource Management Facility). Other MVS facilities provide for the acquisition of trace data. Since these sources have well-defined data contents and formats, third parties have created management tools (especially for SMF/RMF data) that provide extensive reporting capabilities for accounting, security functions, and performance analysis (e.g. MICS, JARS, TSO/MON). Some modeling tools, such as BEST/I and CMF/MODEL make direct use of these same data sources for model definition and validation. Lastly, data manipulation and statistical analysis packages such as SAS have provided a basis for both "home-grown" and vendor-supplied tools, again based on these same data sources.

Performance Management Systems - Technology

DEC provides a set of tools for VAX/VMS, each using its own data collection mechanism and maintaining separate logs for each VAXcluster node. These DEC products include:

MONITOR: This tool provides on-line reporting of system-wide information for a running system. Allows viewing of combined usage from VAXclusters on a single terminal.

ACCOUNTING: As part of VMS, provides basic accounting information and optional information on user jobs or processes, on images or programs executed, and on batch and print jobs. An included utility produces reports.

SPM: The SOFTWARE PERFORMANCE MONITOR provides more extensive data collection and reporting and includes an Event Trace Facility which permits the triggering of custom written trace code capturing data from both the OS, the Record Management Services, device drivers, or applications. SPM can maintain a historical database of information over multiple nodes. Both system-wide and per-process statistics are supported. SPM software does not provide synchronization among nodes of a VAXcluster.

VPA: VAXPerformance Advisor - Collects and analyzes system-wide performance data using a knowledge base of rules and thresholds. VPA synchronizes clocks among nodes in a VAXcluster (to within 0.5 sec).

It is important to recognize the benefits that these and similar facilities offer, however, it is not our intention to replicate either the specific methods or data content.

5.1.2. Accessing Performance Data in A Vendor-Independent Way

This Performance Management Work Group believes that accessing of performance and accounting data through well-defined, standard and non-proprietary interfaces is essential for the creation and wide availability of a toolset that is suitable for large-scale UNIX-based systems management. Such interfaces and their related functions will promote:

- Mutual insulation of client measurement applications from implementation details in the measurement provider or its sources. This facilitates version independence and ease of measurement application maintenance which benefits system vendors, software creators, and ultimately, the system owner.

- Portability of tools. Applications built to a standard, vendor-independent interface can function on various implementations. Well-designed performance and accounting applications can include awareness of both common data and that specialized to a particular architecture.

- Data capture efficiency. Requesting of measurements through a common measurement interface makes it possible to service requests for the same data from multiple measurement applications by distributing data obtained from a single data capture.

- Extensibility of instrumentation. A standard interface for data capture make it possible to add instrumentation in a well-defined and thus more easily maintained way.

- Distributed control of measurement and access to measurement data, even across heterogeneous hardware architectures. Such distributed control and access facilities should also provide the means for achieving a level of coordination and synchronization between dispersed measurements sufficient to make possible a coherent logical view of the data.

- Increased third party applications development. Portability of tools encourages third party interest due to the increased size of the potential market.

5.2. Performance Management Tools

Performance management covers a wide area of related activities and can be grouped into the following three task categories:

- The first category of tasks is related to capacity planning and quality of services as specified in the Service Level Agreements (SLA).

- The second category embraces maintenance, tuning and elimination of bottle-necks, and deals with planning on a weekly or a monthly scale.

- The third category consists of ad hoc operations in order to keep the systems alive and to solve user problems.

The performance management tools provide for configuration planning , capacity planning, on-line performance measurement/monitoring, and expert systems to analyze, interpret and to predict computer systems performance. It is important to note that these performance tools require accurate data in terms of resource and system utilization and this white paper deals with descriptions for the performance data gathering facilities. An example use of the performance management tools in traditional data processing is illustrated in the figures 2 and 3. Figure 2 shows the expected and actual usage in a specified peak period (e.g. 9:00 A.M. -

11:00 A.M.) of application packages in a given production data (computer) center. Figure 3 shows the detailed usage of DBMS commands. Based on the information presented in Figures 2 and 3, the data (computer) center management can easily identify the top running applications and users and, adjust the computer systems resources (CPUs, Memory, Disks, Tuning etc.)

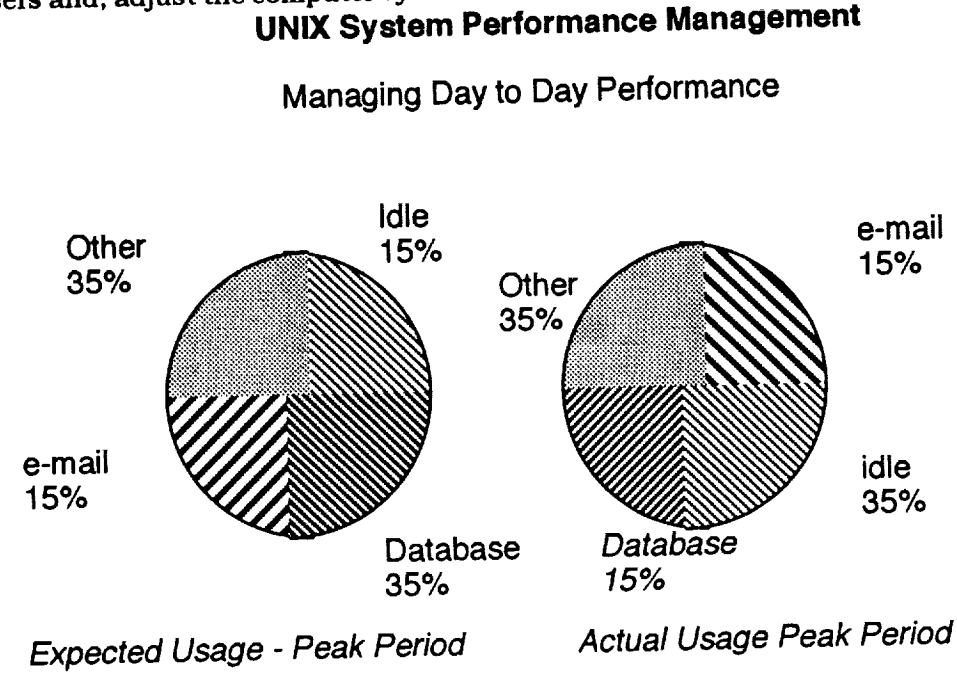

Figure 2. UNIX Performance Management Tools/CPU Usage Comparison

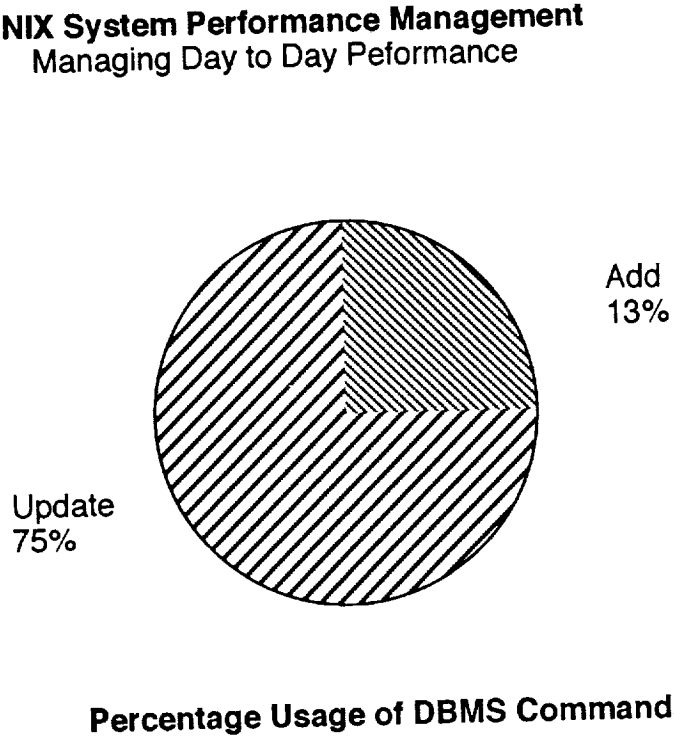

Figure 3. UNIX Performance Management Tools/DBMS Commands Usage

6. Performance of systems: A users perspective

6.1. Overview

Performance methodologies have evolved considerably over the last two decades from an analysis of system utilization, to a degradation analysis of manageable subcomponents of end-user response time (or batch process compete time). The primary focus of the performance analyst has shifted from the resource to the workload. This is sometimes called workload analysis. After workload analysis has been completed, and the critical resource(s) have been identified, the performance analysts secondary focus shifts to dividing the time spent at the resource(s) into subcomponents.

A critical requirement for subcomponents analysis of end-user response time is an architected definition of what constitutes the beginning and ending of a transaction. In the UNIX environment this is not the beginning and ending of a process but must be defined from an end-user perspective. For management reporting and Service Level Agreements it is imperative that response distribution buckets be maintained so percentiles may be reported. This is because response times do not fall into statistically 'normal' distributions making average times difficult to understand.

6.2 Granularity

The required granularity of the subcomponents of response time is dependent upon the level of analysis being done.

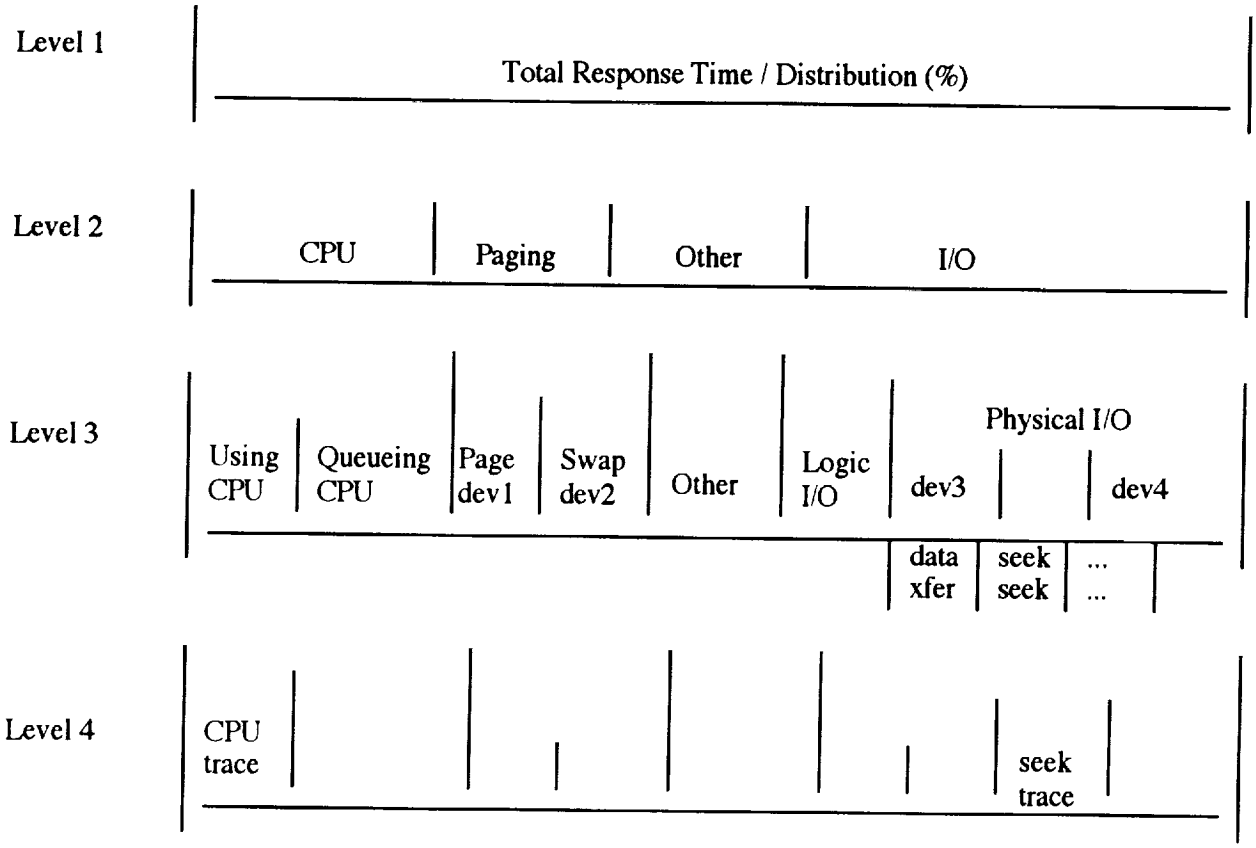

Figure 4. Workload Analysis By Level

At the highest level (what we will call level 1) the total response time or response distribution is all that is required to determine if further analysis is necessary. This information is best gathered by event driven mechanisms.

At the next level (level 2) it may be sufficient to see the delays for CPU, I/O, Paging and 'Other' divided out. These times could include both using and queuing times for each resource. This information is best gathered by high priority state sampling techniques.

At the next level (level 3) each component can then be subdivided into its component parts. For example I/O can be divided into logical and physical. Physical I/O can then be split, by device, into it's measurable subcomponents. This information may be gathered by either high priority state sampling techniques and/or event driven mechanisms.

At the lowest level (level 4) detailed traces can be used to further divide a subcomponent into smaller manageable parts.

Measurement controls should be flexible enough to allow monitoring of individual end-users and groups of end-users by transaction type. Information should be available for both real-time and historical analysis.

7. Summary

In this paper we presented the planned direction of the UNIX International Performance Management Work Group. This group consists of concerned system developers and users who have organized to synthesize recommendations for standard UNIX performance management subsystem interfaces and architectures. The purpose of these recommendations is to provide a core set of performance management functions and these functions can be used to build tools by hardware system developers, vertical application software developers, and performance application software developers.

Published by:

UNIX International
Waterview Corporate Center
20 Waterview Boulevard
Parsippany, NJ 07054

for further information, contact:
Vice President of Marketing

Phone: +1 201-263-8400
Fax: +1 201-263-8401

International Offices:

UNIX International
Asian/Pacific Office
Shinei Bldg. IF
Kameido
Koto-ku, Tokyo 136
JAPAN

Phone: +81 3-3636-1122
Fax: +81 3-3636-1121

UNIX International
European Office
25, Avenue de Beaulieu
1160 Brussels
BELGIUM

Phone: +32 2-672-3700
Fax: +32 2-672-4415

UNIX International
Pacific Basin Office
Cintech II
75 Science Park Drive
Singapore Science Park
Singapore 0511
SINGAPORE

Phone: +65 776-0313
Fax: +65 776-0421