

SYSTEM ANALYSIS AND PERFORMANCE IMPROVEMENT ACTIVITY -- OVERALL GUIDELINES1. Introduction -- The Need for a New Activity

By the end of 1967, the MULTICS system had reached a sufficient level of completion that performance had become an important issue. The scope of the project was widening, as concepts were firmed and coordinated, to include the requirements of a productive system. It was discovered that unless system initialization times were significantly reduced, further integration of system modules would be limited to an unacceptably slow rate of progress. Performance was significantly improved and thus the MULTICS project entered a new phase in which performance became an increasingly pertinent and significant factor.

Subsequently, it has become increasingly apparent that an organized, orderly approach to the problem of performance improvement is needed. In response to this need, the System Analysis and Performance Improvement Activity has been established. Among the reasons for concentrating this work into a single Activity are:

- .The concentration of performance-related knowledge and experience in one group reduces the dilution of knowledge and duplication of effort characteristic of uncoordinated, ad hoc performance improvement efforts.
- .The centralization of knowledge regarding system performance factors and their interrelationships permits decisions about system modification to be made in an appropriately broad context.
- .The possibility of generalized system measurement, data reduction and performance improvement techniques being developed as a

result of localized exposure to numerous specialized instances should result in improved efficiency of the people involved and improved response time for needed system changes.

.There will be a decrease in requirements for the time of other MULTICS personnel best employed in other activities.

In summary, the urgency of a continuing performance improvement effort increases as the emphasis on MULTICS shifts from R and D of concepts toward system productivity. Also, the reasons for centralizing this effort into a single activity, the System Analysis and Performance Improvement Activity, are compelling.

2. Prerequisite for Effective Performance Improvement

The problem of MULTICS performance improvement is unique in at least ^{three} two ways:

.Because of the high content of new concepts and combinations of concepts in MULTICS, many areas of the system have a distinct R and D flavor with generality being introduced in order to facilitate later tuning as knowledge was gained rather than prematurely freezing the design. As a result of this generality, performance has suffered seriously.

.Although performance is becoming important event in currently running versions of the system, MULTICS is far from completely implemented and in some cases design remains to be done. Introduction of these as-yet unintegrated functions will without exception decrease system performance below the current level, in many cases drastically so, unless performance improvements take place on a continuing basis.

.MULTICS is coded in a high-level language. This language itself is new and as a result the language processor generates code, the efficiency of which ranges from fair to extremely poor. This is reflected as poor MULTICS performance.

Because of these unusual conditions, performance improvement cannot be expected to occur at a maximum rate until two constraints are satisfied.

.The credibility of the hard core system design and implementation must be established. [The ad hoc addition of system super-structure upon a basic system, the current performance capabilities of which are extremely poor, is at best unconvincing and at worst potentially disastrous in terms of eventual redesign and re-implementation implications.] It is therefore imperative that a MULTICS performance specification be prepared to establish that minimum level of capability and performance which, in the opinion of all organizations involved, will demonstrate design and potential performance credibility.

[A major utilization of this² document will be to restrict the integration of new functions into the system until a defined basic subset of the system behaves at a sufficiently promising level of performance.] A mutually agreed-upon MULTICS performance specification is indispensable to well-directed performance improvement.

.Not only must direct system improvements be implemented, but also a means for effecting significant improvements in the code used to implement MULTICS must be possible. This certainly includes supervised, selective modifications to the current EPL compiler and perhaps will even include the selective recoding of MULTICS procedures in assembly language.

Performance goal may be reasonable idea, rigid adherence is not

Assuming that these conditions can be met, a rational, useful approach to performance improvement is feasible.

3. General Plan of Attack

Assuming that the above mentioned constraints are satisfied, then three broad policy guidelines suggest themselves:

- .The scope of the System Analysis and Performance Improvement Activity should not extend beyond the basic MULTICS system described in the performance specification until minimum performance requirements have been exceeded. *def? that multics?*
- .The order in which system modules will be examined and improved should in general be from the "inside" outward to the extent that this is meaningful. *meaning?*
- .A continuing effort to improve EPL code generation, with input from the System Analysis and Performance Improvement Activity, is vital to achieving acceptable system performance.

Within these policy guidelines, the following generic steps appear appropriate to the general analysis-improvement sequence for each area being analyzed.

- .Design the appropriate controlled experiments within a reproducible environment to permit the measurement of the system performance factors of interest.
- .Design and implement the software required to perform the experiment and process the results.
- .Perform the experiments in the Bootload and/or MULTICS environment.
- .Interpret (and distribute) the results.
- .Select specific modules or functions for improvement.
- .Recommend system modifications to improve performance to a specific predicted level.

expand what is in help input.

- .Upon receiving approval for modification from the Project Director(?), schedule the (re)design, (re)implementation and integration required for the improvement.
- .Assist as appropriate in implementing the system modification.
The degree to which the System Analysis and Performance Improvement Activity should participate will be determined on a per-improvement basis.
- .Measure performance of improved system to validate predicted performance increase.
- .Recommend, if appropriate, integration of improvements into the standard system.

Individual instances of these tasks as applied against specific functions or modules will be documented, scheduled and staffed using the MULTICS Task Report.

4. Personnel Requirements

The System Analysis and Performance Activity is to be as nearly self-supporting as possible. This implies that a fairly wide diversity of talents and interests in the Activity. Among the types of activity expected are:

- .Running of bootloads and dumps.
- .Design and coding of MULTICS-compatible measurement procedures.
- .Design and coding of data reduction and analysis packages.
- .Recommendations at the design, implementation, coding and EPL code production level regarding specific system modifications.
- .In certain cases, reimplementations or tentative test modifications of MULTICS modules.

Thus personnel requirements seem to range from overall MULTICS understanding to EPLBSA coding and bootloading.

But mostly done by other activity as a consultant to this activity

Initially it seems that four to six people will be adequate to form the nucleus for this activity.

5. Documentation

Documentation from the Activity will include:

- .Task Report entries for specific tasks performed by this activity.
- .MSPM sections if appropriate.
- .Performance and analysis reports.
- .Recommendations for system modifications if documentation is necessary.

Record of how well you are doing - History of improvements, or log.

C.T. CLINGEN