A VIRTUAL MACHINE EMULATOR FOR PERFORMANCE EVALUATION (SUMMARY)

M. D. Canon, D. H. Fritz,
J. H. Howard, T. D. Howell,
M. F. Mitoma, J. Rodriguez-Rosell

IBM RESEARCH LABORATORY
SAN JOSE CALIFORNIA 95193

Virtual machines have long been used for functional testing of operating systems and to obtain the services of multiple operating systems from a single machine. They have not been used for performance evaluation however, because the timing observed by a program executing in a virtual machine is unpredictable and dependent on such factors as system load, real operating system overhead, real scheduling, etc. System level performance evaluation of hardware and operating systems has typically been done by hardware prototyping and dedicated machine benchmarking.

This paper introduces the notion of virtual hardware prototyping for system performance evaluation. The approach makes use of the virtual machine concept by adding timing simulation to virtual machine support. The resulting virtual machines reproduce both machine timing and machine architecture. They are then useful for system performance evaluation.

This approach has been implemented in the Virtual Machine Emulator, an experimental version of IBM's VM/370 Control Program. The Emulator extends the System/370 virtual machine environment to include timing simulation. This allows observation of the behavior and performance of 370 prototypes running real workloads without requiring the physical construction of the hardware.

With virtual hardware, changes to the hardware timing and configuration are easily made, and functional extensions to the hardware (e.g. new instructions) are also possible. Performance is evaluated by executing and observing a workload, including a target operating system and user programs. Because real software is used, many of the simplifying approximations of traditional modeling are not necessary. The objective is to obtain greater accuracy than analytic modeling and simulation, but with an approach that is more economic than hardware prototyping.

To implement virtual machine timing simulation, it is necessary to control or simulate both the apparent passage of time and the apparent times at which asynchronous events occur. Emulator virtual machines do not observe real time. Instead a virtual time is observed which passes only when the virtual processor is executing or is waiting for a virtual event. Virtual asynchronous events, such as I/O or timer interrupts, are scheduled to occur in virtual time by the Emulator. Thus timing interference from other users and from Emulator overhead is not apparent.

Under the Emulator, the user selects not only the memory, channel and device configuration of the virtual machine, but also the timing characteristics of the virtual processor and the virtual disk and tape devices. The Emulator controls the virtual processor speed by controlling the rate at which virtual time passes relative to instruction processing. Virtual device timings are controlled by schedluing I/O operation completions to occur at the appropriate virtual times. All virtual clocks and timers operate in terms of virtual time.

The Emulator is capable of supporting sets of virtual machines with synchronized environments. Synchronized virtual machines observe the same virtual time of day within a user-specified tolerance, and consequently advance through virtual time "in synch". This allows evaluation of sets of communicating machines, for example a network of machines interconnected by communication lines or shared disks. This synchronization is accomplished by special scheduling and dispatching of synchronized virtual machines by the Emulator.

The Emulator has successfully undergone three separate validations. They were done by executing a multi-programmed workload (user programs and operating system) on a real machine, and then executing the workload on a virtual machine that precisely matched the configuration and timing of the real machine. All major performance parameters measured on the virtual machine matched those measured on the real machine within 5%. The degradation due to Emulator overhead was measured at 2.5 to 1.

Application areas for the Emulator include hardware development, installation management and software stress testing. Virtual prototyping enables the hardware designer to economically create and evaluate hardware with new timing characteristics or functions. It enables the installation manager to evaluate alternative system configurations without actually implementing them. It allows software designers to subject code to a wide range of timing environments to verify program and algorithm performance. The Emulator supports these activities without requiring dedicated hardware because the Emulator also supports the standard VM/370 virtual machine interface.