WHITE PAPER

HyperThreading Support in VMware ESX Server 2

vmware®

HyperThreading Support in VMware ESX Server 2.1

Summary

VMware ESX Server 2.1 now fully supports Intel's new Hyper-Threading Technology (HT). This paper explains the changes that an administrator can expect to see when running ESX Server on a HT system and provides details on the advanced algorithms and configuration options used to maximize performance of ESX Server on a Hyper-Threaded system.

Table of Contents

What is Hyper-Threading?

Intel recently developed Hyper-Threading Technology to enhance the performance of their Pentium IV and Xeon processor lines. The technology allows a single processor to execute two independent threads simultaneously. In Intel's terminology, the single chip is referred to as a package, while the hardware threads are called logical processors. While this feature does not provide the performance of a true dual processor system, it can improve the utilization of on-chip resources, leading to greater throughput for several important workload types.

To understand the performance implications of Hyper-Threading, it is important to understand that most processor resources are shared between the two executing threads. For instance, the L2 and L3 caches and all functional units (such as the floating point units and arithmetic/logical units) are flexibly shared between the two threads. So, if one thread is using very little of the cache, the other thread will be able to take advantage of all the unused cache space. However, if both threads demand large amounts of cache, they will compete for the limited capacity and likely slow each other down.

A full discussion of Hyper-Threading processor technology is beyond the scope of this paper. For more detail, readers should consult Intel's technical materials at http://www.intel.com/technology/hyperthread/.

Hyper-Threading on ESX Server: Basic Usage

By default, Hyper-Threading will be enabled during the ESX Server installation process on any hardware that supports the feature. A checkbox is also provided in the Management User Interface to enable or disable HT, although the system must be rebooted for the change to take effect.

Most systems with Intel Xeon MP processors or Intel Xeon processors with 512KB of cache support Hyper-Threading. However, in order for VMware ESX Server to enable HT, the server BIOS must be properly configured with HT enabled. Consult your system documentation or contact the system manufacturer for details on BIOS configuration.

CPU Count and Numbering

In general, an administrator should see few changes between an ESX Server system with HT enabled and one without it. Most obviously, the number of CPUs shown in the Management User Interface will double, and the list of available CPUs for the per-virtual machine "only use processors" setting (also known as CPU affinity) will double.

Processors are numbered so that logical CPUs on the same package receive adjacent processor numbers. That is, processors 0 and 1 are on the first package, while processors 2 and 3 are on the second package and so on. Administrators should note that this differs from the way Windows and Linux number processors in a Hyper-Threaded system. The numbering is especially important when using CPU affinity. Administrators should be careful not to bind two CPU-intensive virtual machines to the same package with affinity, as that may limit ESX Server's ability to take advantages of all packages in the system. Overall, VMware recommends that users avoid manual CPU affinity settings and allow the ESX Server scheduler to manage processor placement for optimal performance.

Processor Halting and Accounting

An operating system can place logical processors into a special halted state when they are idle. This state frees up hardware execution resources to the partner logical processor (the other logical processor on the same package), so that a thread running on the partner runs effectively like a thread on a non-Hyper-Threaded system. VMware ESX Server uses the halted state aggressively to guarantee full utilization of the system's processing power, even when there are not enough running tasks to occupy all logical processors.

ESX Server accounts for CPU time in terms of "package seconds," not logical processor seconds. A virtual machine running on a logical processor that shares a package with another busy logical processor will be charged for half as much as a virtual machine running on a logical processor with its partner halted. In other words, a virtual machine is only "half-charged" when it runs on only half of a package, but fully charged if it has the package to itself. Performance testing has shown this to be the most accurate and understandable way to quantify the impact of Hyper-Threading performance implications.

This style of accounting also makes it easier to compare performance between HT and non-HT systems, because "used seconds" are measured in the same units on both system types.

Performance expectations

Because the benefits of Hyper-Threading depend so heavily on the characteristics of the running workload, it is difficult to generalize about the performance impact of HT. Intel suggests that some applications may see performance improvements of up to 30%, but, in practice, these extreme improvements are rare. More typical applications see performance benefits in the 0-20% range. In some extreme cases, applications may also decrease slightly in performance when run on a Hyper-Threaded system.

When running SMP virtual machines on a system with two physical packages, however, the performance gains may be more substantial. VMware ESX Server coschedules both virtual CPUs in an SMP virtual machine. That is to say, of one virtual CPU in the virtual machine is running, they must both be running or idle. This can lead to a problem of "processor fragmentation" on 2-way systems. Consider the case where a uni-processor virtual machine is running and a 2-processor virtual machine is ready to run. One physical CPU will be idle, but ESX Server will not be able to run the SMP virtual machine, because it would need two available physical processors. Thus, a physical CPU may be left idle.

The above situation would not be a problem for a Hyper-Threaded system. For example, VMware ESX Server could dedicate one package (with two logical CPUs) to the SMP virtual machine and another package to the uni-processor virtual machine (running on one logical CPU, with the other halted), thus fully utilizing the system's resources. This increased utilization can lead to substantial performance benefits for realistic workloads with a mix of SMP and uni-processor virtual machines

Advanced features of ESX Server Hyper Threading Support

In addition to the basic features described above, which are similar to those provided by commodity operating systems for Hyper-Threaded hardware, VMware ESX Server 2.1 provides a number of cutting-edge enhancements and configuration options that advance the state of the art in HT performance and management.

CPU Scheduling enhancements

VMware ESX Server's CPU resource controls have been tightly integrated with HT accounting. Virtual machines still receive CPU time proportional to their share allocation, but capped by user-specified min and max values. While shares allow relative allocation of resources (so that an administrator can specify one virtual machine should receive twice the resources of another virtual machine, for instance), min and max are absolute guarantees, measured as a percentage of a package's resources.. That is, a virtual machine with a min of "75%" and a max of "90%" is guaranteed to get at least 75% of a package's time, but never more than 90%, even if extra idle time is available in the system.

To achieve this level of fairness, ESX Server will dynamically expand a high-priority virtual machine to use a full package by halting its partner logical processor, even if other virtual machines are currently runnable in the system. This does not waste resources but simply redirects them to the high priority virtual machine, so that it can receive up to a full physical package (or two full physical packages for an SMP virtual machine), depending on the administrator-specified configuration.

This feature differentiates ESX Server from some commodity operating systems, which attempt to keep all logical CPUs busy, even if doing so hurts the progress of a high-priority task. Expansion and contraction are fully dynamic and transparent to the administrator.

HT Sharing configuration

While Hyper-Threading Technology can provide a useful performance boost for many workloads, it also increases the possibility of performance interference between two applications running simultaneously. For instance, as discussed earlier, an application with extremely poor cache performance may lead to performance problems for another application running on the same physical package.

On some commodity operating systems, when an application is observed to interact poorly with Hyper-Threading Technology, the administrator has little choice but to disable Hyper-Threading on the entire machine. ESX server, however, provides an additional level of control for administrators to manage package-sharing settings at the level of the individual virtual machine.

Users can select from three choices (called "HT-sharing" settings) for each virtual machine: any sharing, no sharing, or internal sharing only. The default setting, "any," permits ESX server to schedule virtual CPUs from this virtual machine on the same package with any other virtual CPU. This allows the system to exploit Hyper-Threading Technology to its fullest, and it is the best choice for the majority of applications.

The "internal" setting applies only to SMP virtual machines. It specifies that the two virtual CPUs from the virtual machine in question can share a package together, but not with virtual CPUs from any other virtual machine. This contains any Hyper-Threading performance issues within the virtual machine, so it can neither affect the performance of other virtual machines nor be affected by them. ESX Server can still dedicate a full package to each virtual CPU in the virtual machine, if resource constraints and the system activity load permit it. For applications that are quite sensitive to performance variations (such as streaming media servers), this setting may provide the best balance between Hyper-Threading utilization and performance isolation.

Finally, the "no sharing" setting guarantees that each virtual CPU will always run on a full package, with the partner logical CPU halted. This setting can be chosen to maximize the virtual machine's isolation, and it is particularly appropriate for virtual machines running applications that are known to perform poorly on Hyper-Threaded systems.

Selecting the "Isolate from Hyper-Threading" option in the MUI will apply the "internal" settings to an SMP virtual machine, or the "no sharing" settings to uniprocessor virtual machines. The hyper-threading man page describes how to configure these settings via config file and proc node interface.

Quarantine

Among the applications that interact badly with Hyper-Threading, most suffer performance degradations of less than five percent. However, recent research has shown that a malicious application could degrade the performance of another workload running on the same physical package by as much as 90% through, for example, the use of self-modifying code in a tight loop (see "Microarchitectural denial of service: insuring microarchitectural fairness," by Grunwald and Ghiasi, Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture).

Although such an attack has not yet been observed in the field, ESX Server includes special optimizations to ensure that a rogue thread in one virtual machine cannot severely degrade the performance of another virtual machine. ESX Server uses low-level hardware counters to observe the frequency of certain harmful events, such as pipeline flushes and instances of self modifying code. If the number of harmful events observed in a given time period for a certain virtual machine is too high, the system will automatically "quarantine" that virtual machine by placing it into the "no sharing" state, as described in the previous section. This setting protects other virtual machines from the potential denial of service attack, but does not excessively degrade performance for the misbehaving virtual machine, as it loses only the added benefit of Hyper-Threading. If the level of dangerous operations eventually drops below a specified threshold, the virtual machine will be released from the quarantined state. Quarantining is entirely transparent to the administrator and requires no additional configuration.

Interrupt routing enhancements

Particularly for network intensive workloads, context switches due to interrupts can be a major source of overhead. To address this problem, VMware ESX Server 2.1 has tightly integrated the interrupt steering code with the Hyper-Threading aware scheduler. ESX Server minimizes unnecessary context switches by preferentially directing interrupts to halted logical processors, which are already waiting in kernel mode. Similarly, when the scheduler has to decide which logical processor of a package should begin running a task, it will preferentially choose the logical processor with the lower interrupt load.

Further reading and references

Marr, D.; Binns, F.; Hill, D.; Hinton, G.; Koufaty, D.; Miller, J.; Upton, M. "Hyper-Threading Technology Architecture and Microarchitecture: A Hypertext History." Intel Technology Journal. http://developer.intel.com/technology/itj/2002/volume06issue01/ (Feb 2002).

Dirk Grunwald; Soraya Ghiasi. Microarchitectural denial of service: insuring microarchitectural fairness. Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture: 409-418

hyperthreading (8) man page. VMware ESX Server 2.1 man page (2004).

VMware ESX Server 2.1 Administration Guide. VMware, Inc. technical manual (2004).

V00014-20001205

VMware, Inc. 3145 Porter Drive Palo Alto CA 94304 USA Tel 650-475-5000 Fax 650-475-5001 www.vmware.com