The Advantages of Intel® Itanium™ Architecture for Directory Services

Information for IT Managers
and System Integrators

White Paper

The IA-64 architecture and its first microprocessor implementation, the Intel® Itanium™ processor, provide capabilities that enhance the performance and scalability of directory services. The 64-bit addressing enabled by the Intel Itanium architecture will help overcome the scalability barriers and awkward, maintenance-intensive partitioning directory schemes of current directory services on 32-bit platforms. In addition, a number of the Intel Itanium architecture's performance-enhancing features add to the enhancement of directory services, including increased instruction parallelism, speculation, predication, multi-way branching and support for a large number of registers.

Growing Importance of Directory Services

Directory services provide a central repository, which stores a sophisticated set of information about users, systems, devices and other information. A directory service enables a cross-application infrastructure for naming, describing, finding and managing relationships among internal and external corporate resources. Current directory service applications include UNIX's* NIS and NIS+, Novell's* NDS, the Internet's Domain Name Service and Microsoft's* upcoming Active Directory.

Directory services are increasingly becoming critical components of IT infrastructure. In addition, Microsoft's well-publicized development of Active Directory and the increased attention on Novell's NDS on new platforms has made the value of directory services more broadly apparent.

Business interests that drive these comprehensive directories include the efforts of centralized IT departments to drive down TCO and the need for a single-authentication security infrastructure.

The earliest directory services focused on managing user names and passwords or on managing the names of all machines on a network. However, newer approaches (including Novell's NDS, Microsoft's Active Directory, and UNIX LDAP implementations) enable directory services to become repositories for a much broader range of corporate information such as network printers, desktop configuration information and even software objects (e.g., DCOM components). When directory services were just used for logon, performance did not typically pose a critical bottleneck. Performance has emerged as an issue, as more applications have started to depend on newer, general-purpose directory services.

The Intel Itanium architecture provides a rich set of functionality that should enable larger, more powerful directory services. In general terms, benefits enabled by the IA-64 technology in the Intel Itanium processor include:

- Large 64-bit address space: Traditional 32-bit architectures only support up to 4 GB of physical memory; often, 2 GB of that is typically reserved for the system kernel. The Intel Itanium architecture supports a large, flat address space and physical memory addressability that will enable application performance for directory services.

  Scalable high-end directory service applications typically require a fast response time while holding a large or fast-growing amount of data. Placing the entire directory in physical memory accelerates the response time by avoiding the time-consuming performance hit of accessing disk.

- Increased instruction parallelism: Intel Itanium compilers package instructions in bundles (three instructions per bundle) that can be executed in parallel; the Intel Itanium processor provides multiple execution units to execute these instructions concurrently. Increased instructions per clock will benefit a broad range of software routines, particularly code that contains few operations which depend on data processed in the preceding few instructions.

  For directory service applications, many routines should benefit from parallel execution of instructions, especially when doing searches of the directory to retrieve answers to queries.

- Very large set of registers: For about a decade, the IA-32 architecture has relied on eight registers (temporarily holding places for variables used by an application). Code with heavy data manipulation routines often require more than eight short-term holding places to avoid excessive and unproductive loads and stores. This overhead in moves and stores between registers and memory increases the likelihood for processor stalls and reduces performance.

For directory service applications, indexing and searching routines required for directories will be the most likely portion of data-intensive code to benefit from the Intel Itanium architecture's 128 integer, 128 floating-point, and eight branch register.

- **Speculation:** Microprocessor clock rates have historically grown much faster than the speed and latency of system memory, leading to a variety of branch prediction and caching techniques. The Intel Itanium architecture's speculative loads take this one step further, allowing the compiler to schedule pre-fetching speculative loads from memory well in advance of the need for the data, thus removing the latency of the load operations and reducing processor stalls. Many competing RISC microprocessors can pre-fetch only after the last code branch, but the Intel Itanium architecture does not impose this limitation. This additional speculation capability pays off for code with heavy branching (due to "if" statements or procedure calls) and memory access. The Intel Itanium architecture also supports multi-way branching, in which up to three conditional branch instructions are evaluated at the same time. The result is reduced memory latency and faster application processing.

  For directory service applications, speculation means faster traversing and searching for information within the directory service, since the processor can pre-fetch intermediate results as the code traverses a binary tree or hash data structure.

- **Predication:** Predication is the conditional execution of instructions, which allows code to avoid using branch or "if" statements. Instead, the chip executes both paths of the branch at the same time as the "if" statement is being run and then discards the unwanted path once the results of the "if" statement have been determined. Like speculation, predication pays off for code with heavy branching, but unlike speculation, predication focuses on the process of executing parallel code paths rather than on speeding memory access.

  For directory service applications, prediction increases the processing of conditional code for handling TCP/IP packets, directory service requests, authentication requests and (as with speculation) the conditional code for traversing and searching for information within the directory service.

The table below breaks down the unique Intel Itanium architectural features for specific tasks performed by directory services. The isolation of specific areas of likely speedup in the chart does not imply that any routine needs to be optimized by hand for the Intel Itanium processor. Instead, the particular routines will likely employ algorithms that compilers can optimize effectively using a particular feature of the Intel Itanium architecture.

Table 1: Intel Itanium Processor Benefits for Directory Service Tasks

Predication (cond exec)	Speculation (spec load)	Increased # of Registers	Parallelism (incr. IPC)	Task Performed By Application (in chronological order)
–	–	–	X	OS receives a network interrupt, takes bits from network adapter and puts them in a buffer
X	X	X	X	OS processes TCP/IP packet, routes it to directory services applications
X	X	–	X	Directory services applications receives, interprets incoming query or command
X	X	X	X	Directory services applications checks whether the directory operation has been authorized to perform operations on the directory
–	–	–	X	Directory services applications logs the operation request
				Directory services applications performs one or more of the following directory operations:
X	X	X	X	▪ Search/traverse directory
–	–	–	X	▪ Add/update directory entry or hierarchy
X	X	–	X	▪ Retrieve specific directory entry
X	X	–	X	▪ Authenticates user based on an entry
X	X	X	X	▪ Indexes directory
–	–	–	X	Directory services applications updates log, if appropriate
–	–	–	X	Directory services applications sends results or acknowledgment to the network adapter
X	X	–	X	Directory services applications synchronizes updates with remote directories, if appropriate

X = a task that will likely benefit from faster performance due to a specific Intel Itanium optimization feature
– = areas that are unlikely to contain code that will optimize with a particular Intel Itanium feature

Summary

As directory services rapidly grow in size and importance over the next five years, the Intel Itanium architecture provides a high performance 64-bit platform that can scale extremely well. Advanced Intel Itanium architecture features such as predication, speculation, multi-way branching and a large register set can accelerate the execution of core directory service algorithms.

Printed in USA
order number: 283582-001
0101/CMD/JH/1K