



Multi-Core Processing Advancements via Optimized System Resource Allocation and Capacity Management

Benoit Marchand
Founder, eXludus Technologies, Inc.

White Paper
January 2008

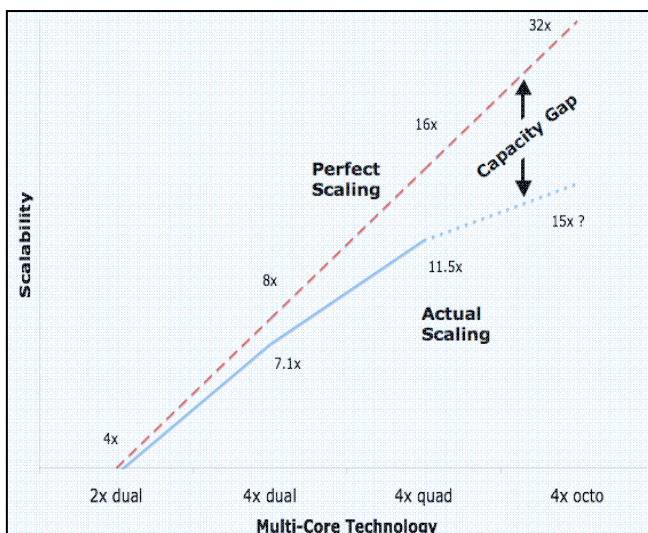
Introduction

Multi-core processors represent potentially the most significant information technology (IT) advancement in decades. The capability this advancement could provide include: enhanced server consolidation efforts; reduced administrative burdens; increased application processing throughput; and decreased energy consumption growth rates and related expenses, a major ongoing IT burden. Realizing these attractive benefits, though, requires new solutions that harness multi-core system potential.

Application throughput increases are essential to the growing class of performance-demanding commercial applications, a fast growing segment dubbed “Extreme Computing” by IDC, that encompasses application areas such as finance, chip design, seismic processing, digital content creation, life science, etc. However, in practice the industry has yet to deploy an efficient and practical way to harness multi-core processing capacity in a scalable way so that the desired application throughput increases may be realized.

Harnessing and scaling multi-core processing is limited by the inability of existing technologies to efficiently allocate system resources. When a single system resource - in this case processing power - scales disproportionately in relation to other system resources, system imbalance may result. Such imbalance creates conditions where applications can exhaust resources or interfere with one another, an exponentially increasing likelihood as the number of processor cores per system increases, unless appropriate measures are taken to avoid these conditions. Failure to implement such measures leads to sub-optimal resource utilization and slower user application performance.

The graph below illustrates this resource imbalance effect. This benchmark (VMware’s VMmark) clearly demonstrates that a single 16 core system yields about 30% less aggregate processing capacity



than four 4 core systems running the same workload. It can be predicted that next generation multi-core processors, with 8 cores each, might yield only half of their full processing capacity on such benchmarks. Given processor roadmaps with plans for 128 cores (and more) per processor, this decreasing marginal contribution per core must be addressed if the potential promise of multicore processing is to be realized.

eXludus’ Multi-Core Optimizer is the first node-level capacity management technology developed specifically for multi-core processors, and has been designed to properly allocate system resources and prevent job interference. Implementing such measures enables multicore systems to deliver their full processing capacity and thereby significantly improve user application performance.

eXludus’ Multi-Core Optimizer is the first node-level capacity management technology developed specifically for multi-core processors, and has been designed to properly allocate system resources and prevent job interference. Implementing such measures enables multicore systems to deliver their full processing capacity and thereby significantly improve user application performance.

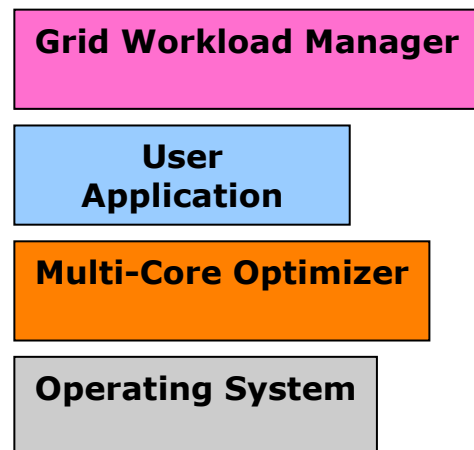
A Capacity Management Problem

Resource imbalance / distribution is a recurring phenomenon in IT's short but prolific history and several methods have been used over the years to solve such imbalances. For instance, in the late 1980's parallelization techniques were widely used to overcome a sudden increase of processing capacity (RISC SMP) on what were essentially high-end workstations. And in the mid 1990's Grid workload managers were introduced to overcome the emerging distributed resource allocation problem brought about by cluster deployments. However, these prior methods are incapable of solving the resource allocation problem introduced by multi-core processors: parallelization is neither practical (too many applications on the market today) nor sufficient (Amdahl's Law of Serialization details the scalability limits of large core count systems, but in brief, even in cases where an application can be parallelized to a high degree it would likely be unable to take advantage of more than 10 cores); grid workload manager resource sampling techniques just can't keep up with the – too – rapid pace of resource consumption changes, nor can actually prevent performance limiting job interference. [Note: Additional information that details these limitations further can be found in a separate eXludus white paper, “*Server Consolidation and the Combinatorial Effects of Multi-core Processing, Virtualization, and Grid Technologies*”.]

eXludus' Multi-Core Optimizer is based on a breakthrough in resource allocation technology. Using advanced math techniques and operational research principles, Multi-Core Optimizer dynamically creates an optimal processing schedule for applications such that resources are used to their fullest extent, while simultaneously eliminating any risk of applications interfering with one another. The principle is quite similar to systems implemented to identify the ideal routing schedules to quickly move the most parcels from senders to recipients given available resources (workers, trucks, planes). In other words, what is the best resource allocation scheme given a fixed set of system resources and application resource needs that optimizes processing capacity? The major difference between the parcel delivery scenario and the eXludus solution is that, where the parcel routes are rather static once initially defined, Multi-Core Optimizer creates dynamic, real-time schedules as each new job enters the system. Thus, Multi-Core Optimizer capacity management is based on a powerful resource allocation mechanism that adapts instantly to changing scenarios.

This capacity management function, however, should not be mistaken with the form of capacity management performed by grid workload managers. Whereas grid workload managers solve the problem of distributing processing capacity, they do not control in any way how and when resources are allocated within the individual processing nodes (other than by limiting the number of jobs dispatched). Multi-Core Optimizer, to the contrary, has no control over the distribution of jobs to processing nodes, but does control how resources are allocated and consumed by the running jobs once they have been dispatched. So, grid level (existing workload manager solutions) and node level (MultiCore Optimizer) capacity management solutions complement one another.

This diagram to the right illustrates the software stack and location of all components.



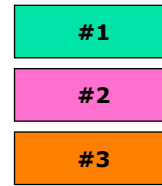
Resource Allocation and Optimization

In a multi-tasking operating system all resource allocation demands are granted by the kernel on the basis that processes run individually while sharing resources. For instance, given three applications that require simultaneous access to the same sharable resource, each application will take roughly three times as long to execute as compared to if a single application had been alone on the system (multi-tasking effect).



1 time unit

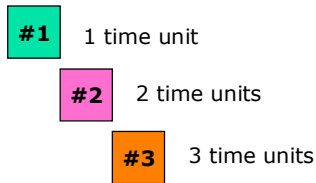
Each application, if running alone, consumes a resource for 1 time unit.



3 time units

But when running concurrently all three applications complete in 3 time units since they must share the resource.

With Multi-Core Optimizer applications collaborate – transparently - with one another in using resources. In the example below the second and third applications are blocked until the first application releases the resource, then the first application signals the second application it may proceed, while the third application remains blocked until the second application releases the resource. Thus the average execution time is now shorter (1.5x shorter than without cooperative behavior).



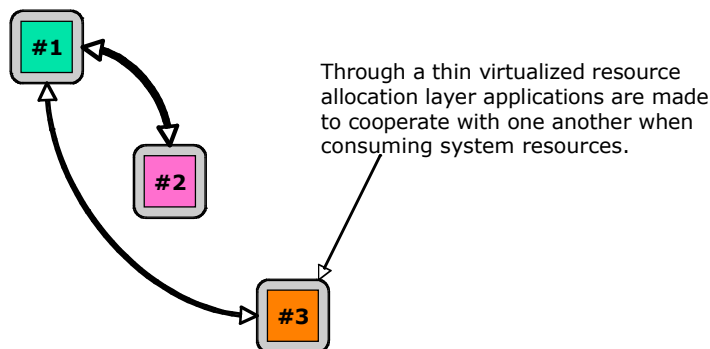
When applications synchronize their resource consumption on average they finish in 2 time units.

Simultaneously to the applications cooperating on resource consumption, Multi-Core Optimizer recognizes that, while the second and third applications are blocked, the resources they have been consuming are temporarily released and fills more work into that available capacity. Throughput has been increased without affecting execution times!



Meantime applications are waiting for their turn to use a resource, they release the other resources they had been using, allowing other applications to progress forward.

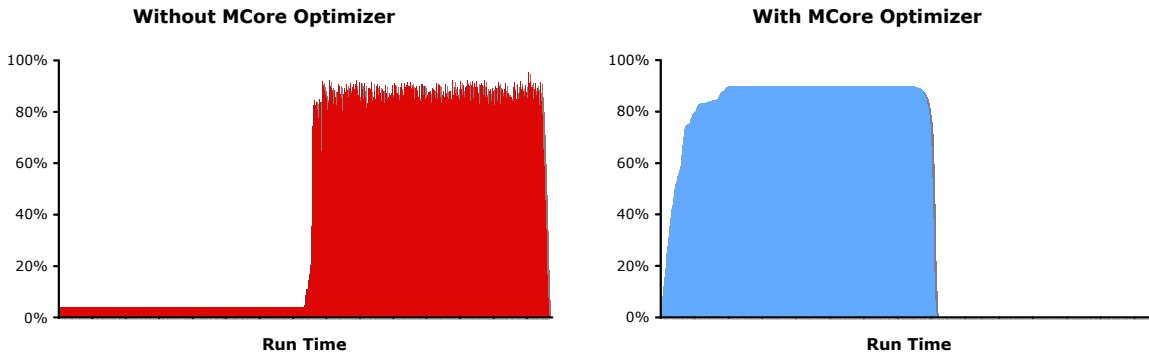
Applications are forced to cooperate with one another through a virtualized resource allocation system than encapsulates all applications and implements the peer-to-peer mechanisms without requiring modifications to applications.



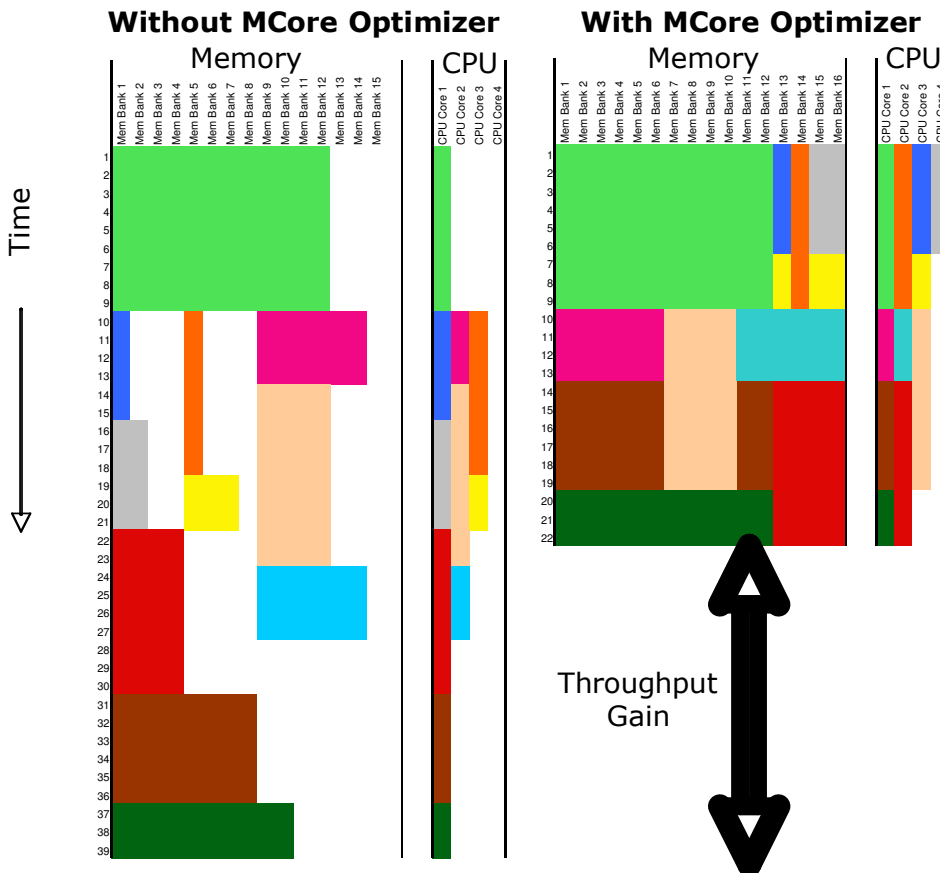
In essence Multi-Core Optimizer serves as a resource broker between the jobs / applications running on a node and the operating system.

Benefits

In the following example Multi-Core Optimizer determined the optimal resource allocation schedule and accelerated the application by hiding all I/O latencies (0.2TB total) that were previously limiting multi-core processing efficiency (N.B. the graphs are generated by Multi-Core Optimizer). The result was that job run time was almost halved! The application used is BLAST on the 6GB NT (human genome) data set submitted as a job array.



In this next diagram we illustrate how eleven independent jobs of varying size are typically dispatched by workload managers to nodes, and how Multi-Core Optimizer increases job throughput through resource allocation optimization.



The white space indicates unused memory or processor resources. As can be seen, with Multi-Core Optimizer resources are more heavily used and all jobs terminate in less time, even though each job's run time remains the same. By jobs spending less time queued, the time-to-response as measure by users is significantly reduced.

Conclusions

Although multi-core processors create a resource imbalance condition that can be harmful to applications and reduce system efficiency yield, eXludus' Multi-Core Optimizer compensates for the imbalance through capacity management and optimized resource allocation. With the ability to make full use of all system capacity, Multi-Core Optimizer paves the way to efficiently use the rapidly increasing number of processing cores, and the much larger multi-core processor systems anticipated.

The benefits of this technology are many, and profound:

- Application performance is improved as a direct result of making full use of the available system capacity. Better application performance leads to faster results and shorter decision times.
- Consolidation efforts can be enhanced as more work can be completed within the same number of – or fewer – systems.
- Administrative overhead is reduced as system administrators no longer need to analyze specific application resource requirements in order to segment and isolate workloads.
- Energy efficiency is improved, as powered cores are not left un- or under- utilized.

eXludus believes technology is only helpful if users can easily make use of it! Multi-Core Optimizer, though a complex technology, has been designed to hide this complexity so that end users can easily take advantage of the technology's benefits. There are no application changes required, no operating system modifications needed, no tuning parameters to master, nor extensive configurations to implement. Applications run as is. Installation takes seconds. Users need not change anything in their everyday work.

In this paper we have focused on the core technology and omitted the benefits of Multi-Core Optimizer's numerous other advanced features, such as automatic job parallelization, asynchronous data staging, feedback mechanisms, on-the-fly virtualization provisioning, etc. Readers are encouraged to discover more about eXludus technology and the company's innovations by contacting its team (info@exludus.com) or visiting its web site (www.exludus.com).