SELECTING THE RIGHT WORKSTATION

Alex Shows,
Performance Engineering
Dell, Inc.

This whitepaper discusses the elements and rationale behind selecting the right computing device to handle graphics or compute intensive workloads.

Selecting the Right Workstation

The most important factor in configuring and buying the right workstation is to know how it will be used. The intended purpose determines which components are critical to performance and those that are optional or even unnecessary. In addition, the more you know about how the workstation will be used, the more performance you'll be able to achieve per dollar spent. By first identifying the various modes of use, and then weighing the importance and frequency of those tasks, you can more effectively determine the right workstation for the job. Selecting the right system configuration and activating the new version of Dell Performance Optimizer 2.0 software will tune the hardware to ensure maximum workstation performance.

Computational vs. Interactive

The first step is to understand the type of work to be done on the workstation, sorting the major tasks into two categories: computational or interactive.

Computational

Computational tasks are typically large and complex jobs the user sets up and then runs to analyze a data model. In most cases, these involve little user interaction and are characterized by high utilization of all the system's available resources. Rendering frames of video, projecting call/put on options, integrated finite element analysis, folding proteins, motion simulation, and computing the down force of a new racecar spoiler design are all examples of computational workloads. Figure 1, below, is an example of heat flux simulation in SolidWorks Simulation, where the results of the computational workload are then available for interaction, enabling visual inspection of the results as they apply to the model.

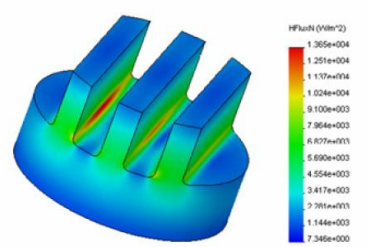

Figure 1 – Heat Flux Simulation

Interactive

Interactive workloads involve heavy user interaction and are characterized by sporadic peaks of high system component utilization separated by idle periods where the user is thinking about the next interaction. Examples of interactive workloads are viewing and rotating an engine model, annotating an HVAC path through a multi-story building and animating a complex, fully-rigged model in a 3D modeling program are all examples of interactive workloads. Figure 2, below, is an example of manipulating a complex mechanical model in CATIA, where all the parts of the assembly are available for editing and annotation in an interactive window.

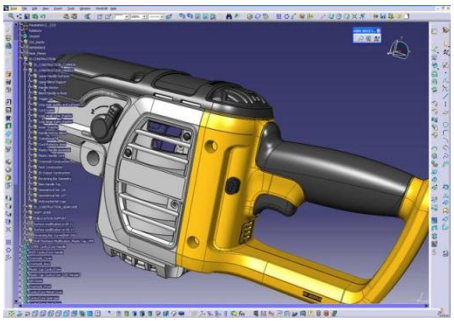

Figure 2 – Interactive Modeling

Component Selection

Splitting the usage model into computational and interactive buckets helps determine the optimal configuration of components such as the best processor and the capacity and number of memory channels populated, as well as the importance of the attributes of those components, such as peak possible CPU

frequency. For heavy computational workloads, it makes the most financial sense to spend money on the relevant components that undertake that work, like CPU and GPU. Also, multi-socket platforms can provide great performance improvements by reducing the amount of time a task requires to complete, so long as the software processing the work is able to scale in performance as processor count increases. If the application does not scale across the available processors, either due to architectural or licensing limitation, the additional cost and complexity of the second socket may not be justified.

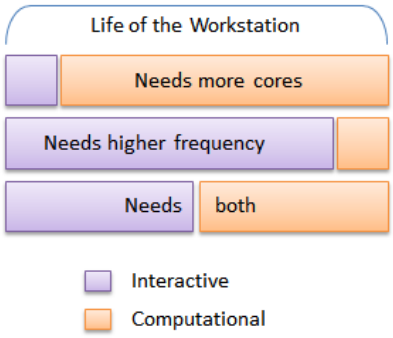

Figure 3 – Weighing Core Count vs. Frequency

Similar to the question of whether a second CPU socket is necessary, some computational workloads may scale in performance by using the Graphics Processing Unit (GPU) as a computational resource. It may help to think of the GPU as a dragster. Given a set of data (fuel), and a straight track (predictable, repeated instructions), the GPU is incredibly fast in a straight line. The CPU, on the other hand, is like a rally car. The navigator inside the rally car is like the CPU's branch prediction algorithm, providing hints to the driver about what turns are coming up and how best to navigate them, while the driver is adept at quickly responding to road conditions around a highly complex track. Similarly, many computationally intensive applications are adept at finding ways to improve their performance through the use of the GPU. Thus, it's important to determine if your application can make use of the GPU, and what type of GPU might be required.

CPU

When choosing a CPU, first think about how much time is spent in computational workloads, where all available cores will be driven for long durations at high utilization. The more time spent in these usage types, the more of the workstation budget should be spent on maximizing core count. Begin by maximizing core count in a single socket, while considering budgetary requirements for other components. Then, if additional computational performance is desired, move to a platform with dual CPU sockets to further increase computational performance.

It is important to avoid maximizing computational performance by moving first to a dual CPU socket platform. While these platforms will provide the best performance, there is a slight penalty due to the nature of multi-socket architectures. This penalty will impact interactive usage models by slightly reducing frame rates possible from the graphics card. Figure 4, below, illustrates the effects of Xeon CPU architectures on graphics performance. The Xeon E3-class CPUs typically include the latest microarchitecture and higher frequencies. The Xeon E5-class CPUs are usually either one microarchitecture revision or one die shrink behind the E3-class of the same point in time. Lastly, the dual-socket Xeon E5-class incurs a slight performance penalty in single-threaded workloads, such as interactive applications feeding the GPU. See below for more on graphics performance.

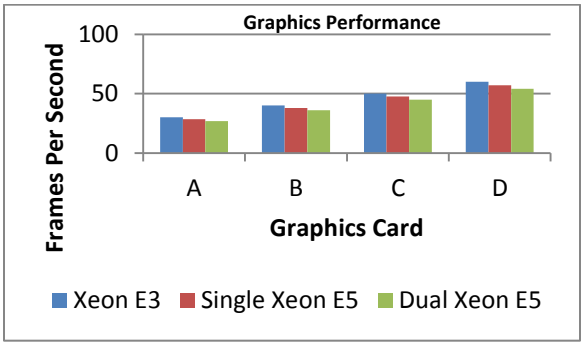

Figure 4 – CPU Impact on Graphics Performance

While it is best to maximize core counts for computational workloads, interactive usage models provide the best performance with the highest CPU frequency. This is because interactivity (as measured by frames per second) is often limited by the efficiency of a single core to feed the GPU with instructions and data. Most modern graphics programming interfaces today can only feed data and instructions to the GPU using a single thread, despite the GPU driver being multi-threaded. As a result, performance benefits with increasing core count are negligible beyond four cores. The more time spent in interactive usage models, the more of the workstation budget should be used to increase the maximum CPU frequency.

Most Intel CPUs available in Precision workstations support a featured called "turbo." Turbo mode is when a CPU adjusts its frequency based on the workload distributed across its cores. When fewer cores are busy, the CPU runs at a higher frequency. The highest Turbo frequencies are possible when only a single core is active. The lowest Turbo frequencies are used when many, or even all, cores are active. This dynamic clocking allows interactive workloads to operate at peak turbo frequencies, while computational workloads still operate above the nominal frequency of the CPU. This is important because comparing the nominal frequency of two CPUs (or their "rated frequency," commonly quoted alongside the model name) isn't always representative of the frequency they will be operating the majority of the time.
To more precisely compare CPUs, one should compare the Low Frequency Mode (LFM), High Frequency Mode (HFM), minimum Turbo frequency (all cores loaded) and maximum Turbo frequency (one core loaded). LFM is important to compare if you want more power efficiency at idle. If the CPU isn't doing any work, how important is it that the CPU consumes as little power as possible? HFM is important to compare if the CPU doesn't support Turbo. Minimum Turbo frequency is important to compare if the CPU will spend most of its time running computational workloads. And finally, the maximum Turbo frequency is important to compare if the CPU will spend most of its time running interactive or otherwise single-threaded workloads.

When weighing whether to maximize CPU core count, or maximum CPU frequency, or some blend between them, one should always seek the latest CPU microarchitecture and generation. Newer CPU generations typically come with either a process shrink (smaller transistors), or a new architecture. Newer architectures often bring greater performance at the same frequency, and this extends beyond just CPU performance. Because many applications spend time waiting on a single core to feed the GPU instructions and data, as the CPU's integer performance increases, so does the graphics performance. What this means is that the same frequency CPU on newer generation architectures can provide higher frames per second with the *same* graphics card!

Graphics

In general, when it comes to graphics cards, the more you spend the more speed you can buy. Speed in graphics is most commonly associated with real-time rendering performance, as measured in "frames per second". The higher your frames per second in an application, the more fluid your interactions with the data model, and the more productive you can be. Computational capabilities aside, finding the right graphics solution for a workstation depends on the desired frames per second in the applications of most interest.

A good rule of thumb for graphics performance is to look for a card that is capable of delivering more than 30 frames per second in the most important applications, using data models and rendering modes most like those in your day-to-day use. While the persistence of vision

phenomenon suggests that 25 frames per second is the minimum required to maintain the illusion of smooth animation, more is always better. If a particular graphics card is able to deliver more than 100 frames per second in a particular rendering method using a specific model size and type, it is reasonable to assume that you can increase the complexity and/or size of the model and still be able to interact with that model without observable stuttering.

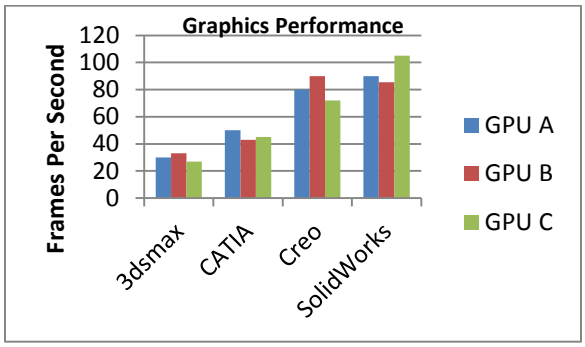

Figure 5 – Graphics Performance Depends on Many Factors

Figure 5, above, illustrates that graphics performance can vary based on many factors. To select the best choice for graphics, it is best to evaluate performance within the particular application or applications used on the workstation. While graphics performance generally scales well moving up to higher class graphics cards (higher core clocks, faster memory, more GPU cores, etc.), one cannot look at aggregate performance across many applications and decide based on this factor alone. This is because two graphics cards in the same class can provide very different performance levels with the same application. In fact, graphics cards in the same class can perform *quite* differently in the same application simply by changing the complexity of the data set or changing the rendering mode. The recommended solution is to identify the class of GPU targeted for the workstation, and then measure each of the cards in that class to determine which provides the best performance under your specific usage. For most, however, this is impractical, so it's important to use some other standard method of measurement as a proxy.

SPECviewperf (available at SPEC.org) is an excellent benchmark for comparing different workstation graphics cards because it measures the frames per second of several varied workloads using rendering methods that mirror that of the many popular workstation applications. With this benchmark, anyone can view the detailed frames per second measurements of several different methods of rendering and compare graphics card performance based on published results. The benchmark also provides representative screen captures of the image quality of these methods. If one were a Creo user, one could use this data to compare how one card performs versus another, not just in Creo, but specifically with a data model and rendering mode that most closely represents their particular use of Creo.

When considering which graphics card is best for your workstation, weigh the amount of time in a typical day that the workstation will spend in either highly interactive work, or in computational work which utilizes the GPU. The more time spent in these usage types, the more of the workstation budget should be spent on graphics. Conversely, the less time spent in these usage types, the more of the workstation budget should be spent on other components such as the CPU, memory, storage, etc.

Memory

It has been said that you can never have too much random-access memory (RAM). While that adage may be true for modern multicore systems running massively multithreaded applications, it is still very important to weigh other factors when considering which type of memory to include in the workstation. For computational workloads, you'll almost always want to maximize the amount of memory bandwidth available to the processing cores. Thus if given the choice about whether to

populate eight DIMMs of an 8 GB capacity each, or four DIMMs of a 16 GB capacity each, choose the option that populates more DIMM slots. The increase in available memory bandwidth will reduce the likelihood that memory bandwidth is the bottleneck to computational workloads, shifting the computational burden back to the CPU cores, frequency, and cache.

Choosing the right frequency is also important, and varies depending on the workload. In applications requiring maximum memory bandwidth, populating all available DIMM slots with the highest-frequency memory is important.

However, some applications require the lowest latency possible, irrespective of available bandwidth, and in that case you would want to populate all available DIMM slots with the lower-frequency memory. An example of this is in random accesses of memory that is small enough to fit in the CPU cache but there is no way for the CPU to predict what memory location to access next.

While memory bandwidth remains important, the lower latency of the slower memory speed can provide benefits to these random reads and writes. For example, in financial markets such as high speed trading transactions, the applications that monitor the current prices of investments like stocks and commodities require the lowest latency so they react as quickly to market changes as possible. Because great fortunes can be gained and lost in fractions of a second, lower latency of slower frequency memory can provide advantages, even if overall memory bandwidth is lower.

Figure 6, below, illustrates this concept using SPECwpc computational workloads. A Precision T7610 platform with dual Xeon E5 CPUs produces significantly higher scores as the number of DIMMs increase, all else being equal. What this translates to in the real world is lower times to complete jobs like rendering, finite element method (FEM) analysis, computational fluid dynamics and similar. It should be noted that the benefits are specific to computational workloads. However, the benefits of populating more DIMM slots for interactive workloads are difficult to measure, as most graphics workloads fit into graphics memory and most graphics cards have dedicated memory on the card.

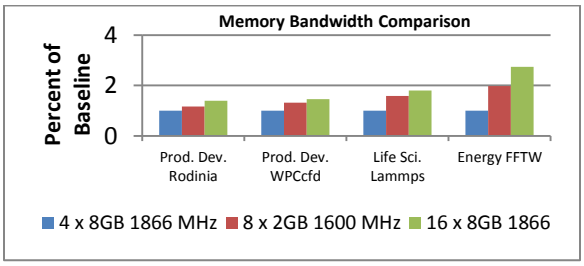

Figure 6 – SPECwpc Improvements with increased DIMM slot population

Lastly, when the integrity of data used in individual computations is paramount to the end result, Error Checking & Correction (ECC) memory should be used. For example, when iterating across a large dataset where the outputs of computations are continually provided as inputs into another sequence of computations, one mistake missed in early computations can have a dramatic impact on the final outcome.

One unique feature of Dell Precision Workstations is the incorporation of Dell's exclusive, patented Reliable Memory Technology (RMT). RMT uses the Error Correcting Code (ECC) data to identify specific locations in the DIMM where errors are occurring, if they occur. RMT can identify an error at a particular memory bit location, and automatically mask that location in memory to ensure that a healthy location is used for subsequent reads and writes. The net effect of this is that, rather than replacing an entire DIMM as ECC errors become overwhelming, RMT masks only the small regions of concern and allows the DIMM to operating normally, increasing system availability and reliability, and extending the usable life of the memory by allowing the user to continue to work using a

DIMM that would require immediate replacement on other systems.

Storage

There are a wide variety of storage performance considerations that depend completely on the usage model. For instance, is the data on the network or stored locally? If so, how frequently are updates committed to the network resource? If not, how much capacity is required locally? Is redundancy required on the local storage? All of these factors are important to determining the right storage components for the workstation, and due to this complexity the subject deserves much greater attention than given here.

For simplicity, we'll assume the data is stored locally on the workstation and not concern ourselves with network bandwidth, frequency of updates, and check-in/check-out procedures. A single user of the workstation will have a blend of three common local storage use cases:

- **"Office Productivity"** – reading and writing small files with occasional large file transfers
- **"Interactive Workstation"** – opening and saving a wide variety of file sizes
- **"Computational Workstation"** – iterating across very large sets of data, often generating large temporary files

Optimizing for the Office Productivity use case is usually as simple as weighing anticipated capacity needs with the highest-performing drive class within the budget. While rotational drives have traditionally dominated this segment, in recent years the decreasing cost of MLC (multilevel cell) memory and controllers has brought the more favorable solid-state drives (SSDs) and hybrid drives within reach of more users. In general, hybrid hard drives provide the best price/performance for this use-case, while SSDs provide the best outright performance. Hybrid hard drives function by storing the most commonly used data in cache, because it is faster to access than from the rotating media in the drive. As long as the files in use are relatively small, data is kept on the flash memory resulting in faster performance.

Figure 7, below, is an illustration of the typical scaling one might expect across the various usage models and drive types. SATA and serial-attached-SCSI (SAS) drives are architecturally very similar, so their scaling is uniform across the workloads and depend primarily on disk interface type, rotational speed, and on-board memory. Hybrid drive performance can vary greatly depending on the workload. The more deterministic and repetitive the workload, the better the hybrid performs. However, hybrids are limited by the size of their flash cache, thus for computational workloads that iterate across a large dataset in a non-deterministic way, hybrids provide less benefit than office productivity usages. SATA and PCIe SSD provide significant benefits over rotational drives, most notably in random reads. SATA SSDs are limited by their interface type, so for maximum throughput in interactive and computational workloads a PCIe SSD provides significant improvements.

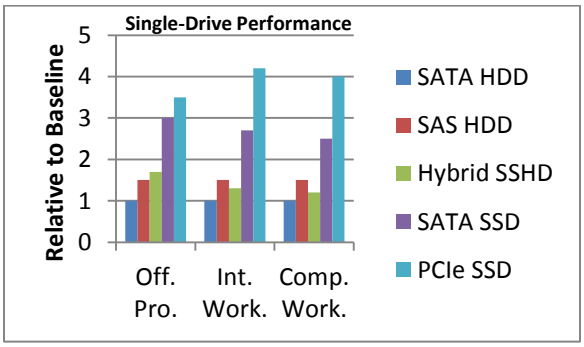

Figure 7 – Single-Drive Performance Comparison

An "interactive workstation" usage model requires greater performance, and this is where SSDs, SAS drives, and RAID arrays begin to play a more important role. If a single SSD provides the capacity needs of both your office productivity and interactive workstation usages,

this option will be the best performing short of a multi-drive RAID 0.

RAID arrays enable the creation of a large virtual drive that spans one or more physical (or logical) drives. Depending on the RAID type, new features such as redundancy (having more than one copy of the data simultaneously) and greater performance are possible. If redundancy is more than or equally as important as performance, having a RAID array such as a RAID 1, 10 or 5 would be a better choice. Then, it becomes a decision between available (matching) drives to build the array. Moving to a RAID can increase storage costs considerably, making it prohibitive to include high performing drives in the array. One way to mitigate this cost while maintaining high performance is to use an SSD boot drive with the operating system and applications on it, while building a RAID array out of lower-cost rotational disk drives to store the larger datasets.

For the computational workstation wherein significantly large datasets are used, the only option for this type of usage may be a RAID array composed of large drives. By combining the smaller capacity drives into a single large volume, the application can use all of this capacity as if it were a single large drive. Multiple drives in RAID 0 will maximize performance *and* capacity, but this provides no redundancy. Multiple drives in RAID 1 provide redundancy but don't maximize performance *or* capacity. RAID 10 increases performance, capacity *and* adds redundancy, but is the most costly in terms of the number of drives required.

Between RAID 0 and RAID 10 is RAID 5, which provides increased performance, capacity *and* adds redundancy with *fewer* drives than a RAID 10, but requires more overhead to manage the array due to the computation of parity data that is then distributed across the array. When considering whether to add a fourth drive to an integrated storage controller and creating a RAID 10, consider the option to upgrade to a discrete RAID controller with on-board memory and moving to RAID 5. You're likely to see higher capacity and the addition of a discrete RAID controller may mean higher performance in office productivity and interactive workstation usages, not to mention benefits to computational workstation usage types.

Figure 8, below, illustrates the scaling of performance across various usage models with different RAID types. The scaling here assumes a full hardware RAID controller card, such that parity computation and the full RAID protocol stack is offloaded from the CPU. All other variables of system configuration are kept constant. Each bar is also a mean across a number of different measurements that include reads, writes, random, sequential and variations of these. As you can see, the two-drive RAID 0 performs well, but offers no redundancy. RAID 1 offers redundancy and the presence of two drives does improve read performance, but there is a slight penalty in write operations whenever the writes must be committed to the disks. RAID 5 sees great benefit with three drives, and the hardware RAID controller unburdens the CPU from parity calculation, lessening the impact to write performance of those operations. It's important to avoid RAID 5 if using a software RAID controller, because the CPU will be taxed by parity computation and performance will be much lower in many cases than other RAID types. RAID 10 offers an excellent blend of performance and redundancy, though it requires four drives to build.

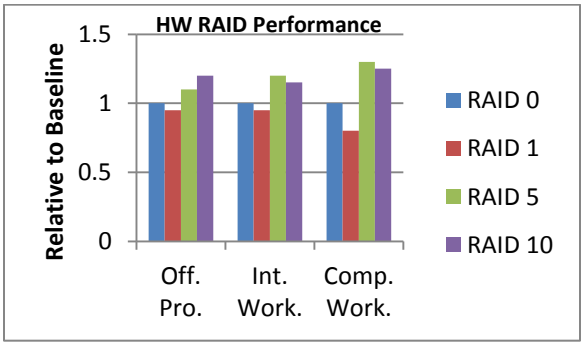

Figure 8 – Hardware RAID Performance

Application Certification

One of the key differentiators of professional workstations, as compared with conventional PCs, is the platform certifications to run specific professional workstation applications. Considering the complexity of the software environment, one can imagine the incredible number of variations that might exist in operating system versions, application versions, and hardware, firmware, and driver versions. All of these variables can have an effect on application stability and performance. Workstation certification addresses this complexity by carefully documenting which configurations of a system were determined to be compatible with the specific application version of interest. The latest certification list is available on Dell.com/Workstations. This reduces the risk to the user when considering a new workstation or an upgrade to an existing workstation, as they can be confident before purchasing that the specific workstation has been certified by the software vendor of their desired application.

Conclusion

To find the right workstation configuration, first identify the primary and any secondary or tertiary usage models.

For **interactive usage models**, focus on maximizing CPU frequency, followed by the class of graphics. For individual CPU models, choose the latest architecture and look primarily at the peak Turbo frequency. Of the available CPU models, determine the best frequency for the price. Then look to graphics and compare frames per second using industry-standard benchmarks such as SPECviewperf, focusing on the applications and/or rendering modes that are most important to your usage. Judge which of the available GPU models offers the best frames per second for the price. Then look to memory and maximize memory bandwidth at the capacity desired. And finally, look to storage, where a single SSD might address all the interactive usage model needs, unless capacity or redundancy requires a RAID array, or spending limits dictate a single rotational disk drive.

For **computational usage models**, focus on maximizing core count, followed by CPU frequency. For individual CPU models, choose the latest architecture and look primarily at the lowest Turbo frequency (which reflects the lowest frequency the CPU will Turbo up to under heavy load). Look for the best core count per dollar, and if the workstation will spend more than half of its life in computational work, consider upgrading to a dual-socket workstation. If the application supports GPU compute, consider upgrading the GPU to models with more compute cores, as the performance per dollar in GPU upgrades will often be higher than the CPU (here again based on the percentage increase in core count). Upgrade memory by populating as many slots as possible first; except for a limited set of applications which are highly sensitive to latency, it is always best to upgrade to the fastest memory speed for the maximum possible computational throughput. Finally, consider the storage requirements primarily in terms of capacity and bandwidth required by the application, which is often much larger and higher than with other usage models.

Optimal performance for a particular usage model can be achieved by identifying the factors that are most important to that usage: those having the highest impact on performance. Combining those selections in a workstation certified for us with the key applications of that usage model will ensure that user has the best possible experience at any price.