Amdahl versus Gustafson

TidalScale
Software-Defined Servers

Contents

1 Enabling Gustafson's Law 1

2 TidalScale Architecture 2

3 Revolutionary Customer Value 4

1 Enabling Gustafson's Law

Computer Science is obsessed with "negative results" and "limits." For instance, most people have heard of Amdahl's Law which seems to suggest that parallelism doesn't help computing beyond a certain point. Many are skeptical that there is a market for bigger single systems because Amdahl's law appears to prove that larger multicore systems won't solve today's problems any faster.

The reasoning behind Amdahl's Law depends on an assumption: that all the parts of the problem must interact such that the performance-constraining set of operations is nearly always sequential. It seems intuitively obvious that if programs are sequential and parallel programs are just a way to speed up a sequential program, then parallelism cannot help. But intuitively obvious things can be very wrong.

There is a countervailing law, Gustafson's Law, which states that computations involving arbitrarily large data sets in fact can be efficiently parallelized. Gustafson pointed out that Amdahl's Law focused only on problems that don't or never will need to scale to larger and larger datasets. Gustafson's Law instead proposes that programmers tend to set the size of problems to use the available equipment to solve problems within a practical fixed time. Therefore, if faster (more parallel) equipment is available, larger problems can be solved in the same time.

Many experts will quote Amdahl's Law as if it applies to every problem; relatively few are aware of Gustafson's Law (though it is well known in narrow systems architecture circles). It turns out that most people scale systems up not to make a particular problem run faster, but to handle larger and larger problems of the same sort.

Right-Sizing Servers with TidalScale

TidalScale is the first technology to fully implement Gustafson's law in the way it delivers scaled speedup for computations on larger and larger datasets and at low incremental cost. It does this by enabling users to configure existing commodity servers into Software-Defined Servers of virtually any size. TidalScale's Software-Defined Servers, called TidalPods, show that when you scale up a problem you **can** add more and more commodity processors to the TidalPod configuration of servers. Conversely, when you scale down a problem, you can easily remove servers from the TidalPod configuration to use fewer processors, less memory, or both.

TidalScale's Software-Defined Servers allow you to right-size a server on the fly to fit any data set. This flexibility counters Amdah's law against parallelization and provides a system where parallelization becomes a viable possibility that would allow for more efficient problem-solving for large data-intensive problems.

The key to right-sizing a server is TidalScale HyperKernel software. HyperKernel is a unique layer of software that sits underneath an operating system, such as Linux. HyperKernel is very low-level and very fast, and it helps users achieve the inverse of what traditional virtualization products do: instead of running multiple virtual machines on a single physical server, HyperKernel runs a single virtual machine on multiple physical servers.

This is possible today because of four things:

- CPU hardware acceleration for memory management operations: this was put in place by CPU vendors for virtualization (specifically, VT-x and VT-d),
- Low network latency: TidalScale uses raw Ethernet frames on a standard 10Gb Ethernet as its system resource bus,
- TidalScale's virtualization of all resource types: Memory pages, Processor State, Network I/O, & Storage I/O, and
- Denning's locality principle combined with TidalScale's use of patented machine learning technology to apply dynamic locality learning.

TidalScale's mobilization of all resource types enables them to flow around the system like the tides. TidalScale is unique in its ability to migrate interrupts and vCPUs to where they need to be delivered. By combining the hardware cost linearity of scale out with the software development ease of use of scale up, TidalScale delivers a radically improved price/performance that enables a range of new use cases. Now, customers can economically size their systems to fit the size of their computing problems because TidalScale:

- Redefines the economics of both scale up and scale out by seamlessly binding multiple commodity machines into a single virtual system that is an aggregation of the physical machines underneath.

- Brings new flexibility to on-premise and cloud data centers.
- Helps enterprises derive more value from existing datacenter resources.
- Provides automatic optimization for best performance at every instant.
- Is 100% binary compatible with existing OSs and applications.
- Is a 100% software product that always benefits from the latest hardware advances.

2 TidalScale Architecture

The focus of the TidalScale architecture is to deliver immediate customer productivity and performance to enterprise customers. Enterprises, for instance, are not interested in rewriting their software applications.

Addressing several constraints of the modern data center,TidalScale is focused on the need that enterprise applications have for more flexibility in creating servers of precisely the right size to run any single program or tackle any problem, large or small. The speed of running programs in memory is undeniably better than that of almost any other option, including AWS or SSD to name a few, as supplemental storage. The benefit of TidalScale is that it is both cost-effective and very simple and to add or remove commodity hardware servers to the single software-defined server through HyperKernel's point-and-click control panel.

TidalScale is not aiming to win the competition for the fastest supercomputer on standard HPC benchmarks but is instead focused on enabling enterprises to flexibly address unpredictable workloads and to solve data-intensive problems as they grow larger. Most of the differences with ScaleMP are driven by the differing needs of the enterprise market versus the HPC market.

TidalScale delivers several main improvements over traditional solutions:

- **Virtualization of all resource types**: Memory pages, Processor State, Network I/O, Storage I/O. TidalScale mobilizes all resource types which enables them to flow around the system. TidalScale is unique in its ability to migrate interrupts and vCPUs to where they need to be delivered.
- **Dynamic locality learning**: Rather than make software developers write to processor physical location, TidalScale dynamically learns better ways to map virtual resources onto physical resources as the application is running. Depending on the pattern of stalls (what Intel calls VMEXITs), TidalScale intelligently adjusts physical placement of resources to maximize locality effects automatically. For instance, by dynamically tracking working sets TidalScale can avoid busting them up when it would be better off migrating a virtual processor to a node with resources that are working well together. If a virtual processor needs to perform a PCI operation on device located on a different node, TidalScale will migrate the vCPU and then try not to subsequently migrate it away too early. Because TidalScale's virtual to physical processor mappings are hierarchical, it is able to be very cognizant of the cache affinity at all levels: hyperthreads, cores, sockets, boards, and system. It has been shown by systems architecture research that there is considerable headroom for performance gains that results from coalescing the cpus, dataset memory, and I/O so those running a particular application are "close together." Hundreds of times per second, TidalScale observes all application reference patterns (resource working sets) and remaps the physical to virtual configuration to deliver maximum system performance automatically. Physical-to-Virtual mapping evaluation happens at each stall, so learning and convergence tend to be rapid in practice. This is novel intellectual property.
- **Software Compatibility**: TidalScale requires no changes to the OS or to any applications. TidalScale is completely 100% binary compatible with Operating System and Application software. TidalScale works with unmodified versions of a range of operating systems including Linux-based distributions (Red Hat Enterprise, Ubuntu, and CentOS) and FreeBSD. TidalScale has tackled and solved the more general SSI problem at a lower level that does not require perfect knowledge of the location of all virtualized memory pages in the system. Each node maintains its own view of the rest of the system, and the system recovers if that view turns out to be outdated.
- **Hardware Compatibility**: TidalScale can support a wider variety of hardware devices. This comes from TidalScale's systematic virtualization of all resource types which means the system is uniquely able to support hardware timing requirements by migrating vCPUs to the nodes containing the device physical hardware. For some devices, vCPU migration is the only way they can be supported.

- **Scalability**: A TidalScale system is more scalable because it has no single bottleneck. For instance, system timing is implemented on a distributed set of clocks implemented transparently to the guest OS and applications. TidalScale systems are loosely coupled and thus avoid the need for system-wide synchronization.
- **Industry standard system management infrastructure**: TidalScale's top-level architecture is based on a small "control computer" that manages the key "glue" that holds the system together, the internal buses, power control, boot control, etc. This means that installation, setup, and operationally, the TidalScale system behaves like one large computer, and can be managed through software that runs in its administration interface within a datacenter. It also creates a clear security boundary. The management node is not a constraining element in the performance of the core system.
- **Price/Performance**: The TidalScale architecture adds up to revolutionary price/performance because customers can assemble systems using commodity hardware, commodity networks and unmodified Operating Systems and software applications.

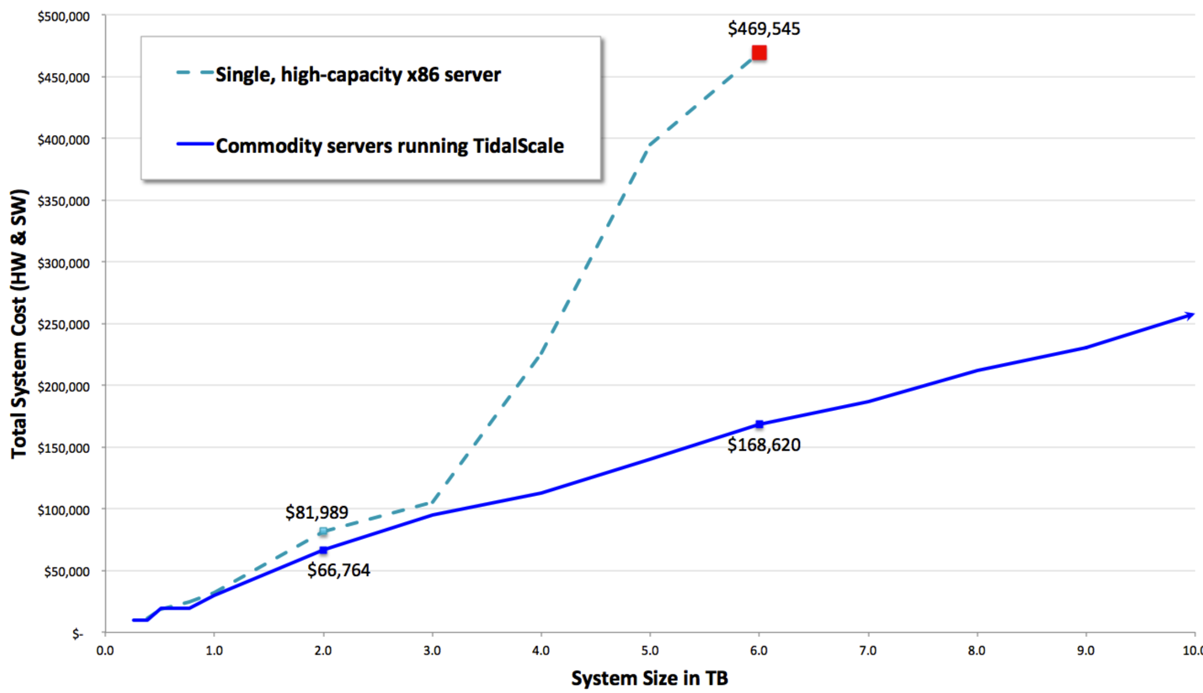


Figure 1: TidalScale Delivers Linear Hardware Costs compared to Traditional Systems

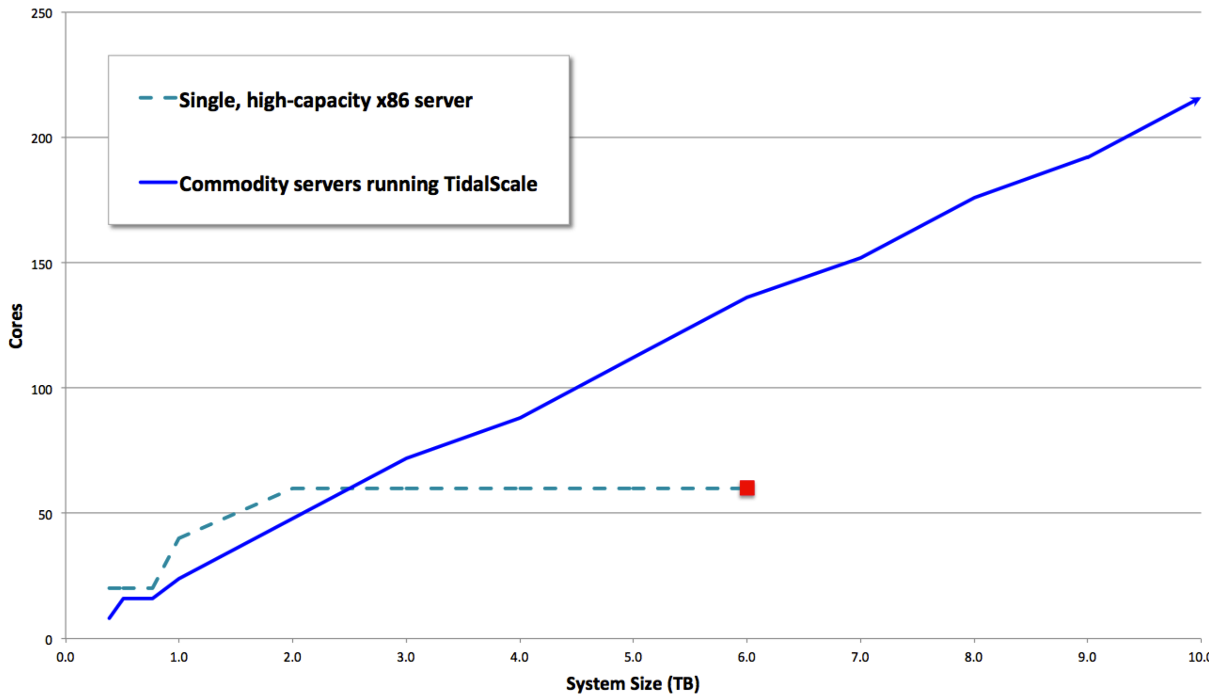


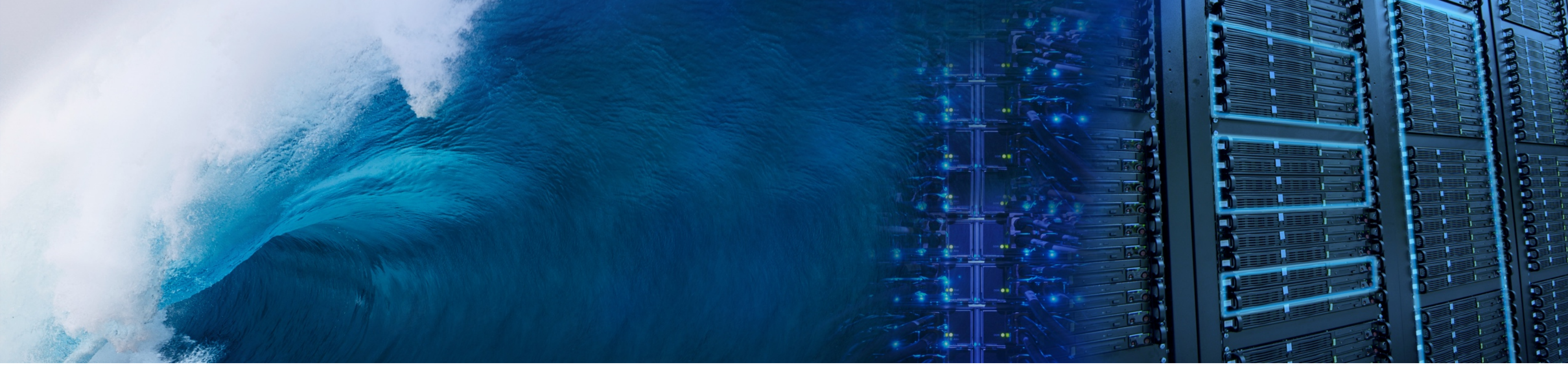
Figure 2: TidalScale Delivers Linear Performance Scaling compared to Traditional Systems

The combination of linear hardware costs, automatic locality learning and a standards-based deployment process make TidalScale ideal for deployment in enterprise data centers.

3 Revolutionary Customer Value

TidalScale uniquely combines the hardware cost linearity of scale out with the software development ease of use of scale up and thereby delivers a radically improved price/performance level that enables a range of new use cases.

TidalScale delivers the first technology to fully implement Gustafson's law in the way it delivers scaled speedup for computations on larger and larger datasets. The TidalScale architecture offers the revolutionary flexibility to scale your processors to your problem, and to right-size servers on the fly to fit any data set. In accordance with Gustafson's Law, TidalScale allows programmers to adjust equipment to their software, allowing for productive parallelization of large enough problems.



TidalScale

Software-Defined Servers