Economics of Software-Defined Servers

TidalScale
Software-Defined Servers

Contents

1 Overview 1

2 Scaling to Solve Large Problems 1

3 TidalScale's Approach to Virtualization 1

4 Scaling Up for the Cost of Scaling Out 2

5 Turning 30 Servers Into 3 2

6 Turning 240 Servers Into 24 2

7 Running Big Data Solutions Without Modification 2

8 Summary: The New Economics of Computing at Scale 3

Neha Nagesh <neha.nagesh@tidalscale.com>

1 Overview

We introduce TidalScale Software-Defined Servers to answer the question, "What are the new economics of scaling to solve the problems posed by unpredictable workloads and big data?"

TidalScale's innovation is efficient, flexible, and future-proof. With TidalScale, users can now right-size their systems on the fly for the cost of commodity hardware, completely independent of the moving target of leading-edge technology. TidalScale Software-Defined Servers bring new levels of capability to modern data centers so organizations can flexibly address constantly shifting workloads, seasonal spikes in computational needs, and data that is growing in volume, velocity and variety. As an additional benefit, TidalScale Software-Defined Servers improve the performance of popular scale technologies, such as Hadoop and Spark, that run on clusters today.

Software-Defined Servers represent the missing piece of the software-defined data center. Unlike storage and networking which have been virtualized for years through software-defined solutions, servers have remained a fixed resource. This puts costly limitations on IT administrators and data scientists, leaving them to either over-provision with their next server purchase or spend weeks, even months rewriting code to run across clusters.

2 Scaling to Solve Large Problems

For many users, the focus is on some of the largest problems with datasets ranging from 500GB to 20TB or more. Graph analyses such as genomics, database queries, or simulations where it's best to have the entire dataset in memory are especially interesting to us. Given that DRAM is 1000x faster than fetching data from solid-state storage, memory-intensive applications can hit a memory cliff, suffering huge performance degradation or even failure when memory won't accommodate the dataset.

Traditionally, users approach these big data problems by scaling in one of two dimensions, up or out. Scaling up is easiest – buy a bigger computer. This solution has several drawbacks:

- The user must know in advance how big the largest problem will be — a challenge in an era of increasingly unpredictable workloads.
- The computer may be too expensive.
- There may be no machine sufficiently large, whether or not cost is a factor.
- Scale up is often not a practical route in the big data arena.

More commonly, users scale out by adding multiple servers to a cluster and building their solution on a layer of software such as Hadoop, Spark, or Cassandra that organizes the cluster in a specialized problem-solving engine. Scale out solutions can grow to many hundreds of servers, for the incremental cost of commodity hardware. However, they are subject to performance issues tied to the speed of server storage and network interconnects. Their biggest impact on users is the specific software environment they present and possible need to restructure, or shard, the dataset across server storage.

3 TidalScale's Approach to Virtualization

Until now, virtualization has meant dividing a machine into multiple independent virtual servers. TidalScale turns virtualization upside down by binding multiple physical servers into a single virtual server.

A Software-Defined Server arises from the TidalScale HyperKernel running on multiple underlying physical machines. The virtual machine, referred to as a TidalPod, is interconnected across an industry standard 10 Gigabit Ethernet. The Guest OS sees the TidalPod as a machine with a single, linear memory space that combines the memory of its components with a number of CPU cores that is the sum of the cores across its components.

4 Scaling Up for the Cost of Scaling Out

TidalScale introduces a new economic perspective to big data computing: the simplicity of Scale Up for the hardware cost of Scale Out. A TidalScale Software-Defined Server can host a range of operating systems, notably Ubuntu and Red Hat Linux, CentOS, and Windows Server, without modification. The HyperKernel starts on the physical servers, presenting the cluster as a single machine with a large number of CPU cores operating over a single linear memory space. The collective storage devices and network connections appear as a single system to the OS.

This is the essence of Scale Up: an ever-growing machine with many cores and much more memory. Because the growth comes with the addition of commodity servers, the underlying economics are those of Scale Out — that is, linear cost structure.

A TidalScale Software-Defined Server is flexible: hardware servers can be added or removed as needed for any given problem. It's also future-proof, because not only is the user free from requirements to define the size a problem upfront, but the underlying hardware isn't tied to rolling technology changes.

5 Turning 30 Servers Into 3

Consider a relatively big data ecosystem of 30 hardware servers. Whether they are 128GB or 768GB machines, a user employing TidalScale technology might bind them into three Software-Defined Servers of 10 nodes each. In the case of 128GB nodes the result would be three 1TB servers.

Here are some observations:

- Regardless of application context, a user sees only three, rather than 30, servers. This is a vastly more tenable system management prospect.
- When applied together, the three software-defined servers guarantee availability in the face of a single hardware failure.
- Widely used clustering technologies such Hadoop, Spark, and Cassandra run better on TidalScale software-defined servers.
- A user may configure multiple virtual nodes within each TidalScale Software-Defined Server without regard to the underlying hardware organization.

6 Turning 240 Servers Into 24

The example of the last section can also be applied to even larger ecosystems. A more ambitious collection of 240 hardware servers condenses down to just 24 virtual servers. The 10:1 ratio is just an example, and there is much room for flexibility. Choosing 5:1 yields 48 virtual servers and greater granularity, while choosing 20:1 condenses the 240 servers down to just 12 virtual servers to address particularly demanding in-memory computing problems or core-hungry applications.

7 Running Big Data Solutions Without Modification

A TidalScale Software-Defined Server essentially acts as a monster computer with terabytes of RAM in a linear address space and dozens of processor cores for parallel execution. This simple model frees users from having to pre-configure their problems for a more traditional clustering technology. Users with a multi-terabyte dataset, whether derived from a database, a simple table, or a dynamically-generated graph, can now run their application of choice, without modification, operating entirely in memory. This freedom to run huge R, Python, database, or simulation programs promotes experimentation at scale and expands the search for the best approach to a computation destined to be run repeatedly.

One dimension of application experimentation is the changing scale of solutions. TidalScale makes it easy to reallocate hardware nodes across multiple Software-Defined Servers. The configuration is fluid while the high-level view remains simple: a virtual server of X gigabytes of RAM and Y processor cores, where X may range well into the thousands, for terabytes of available memory.

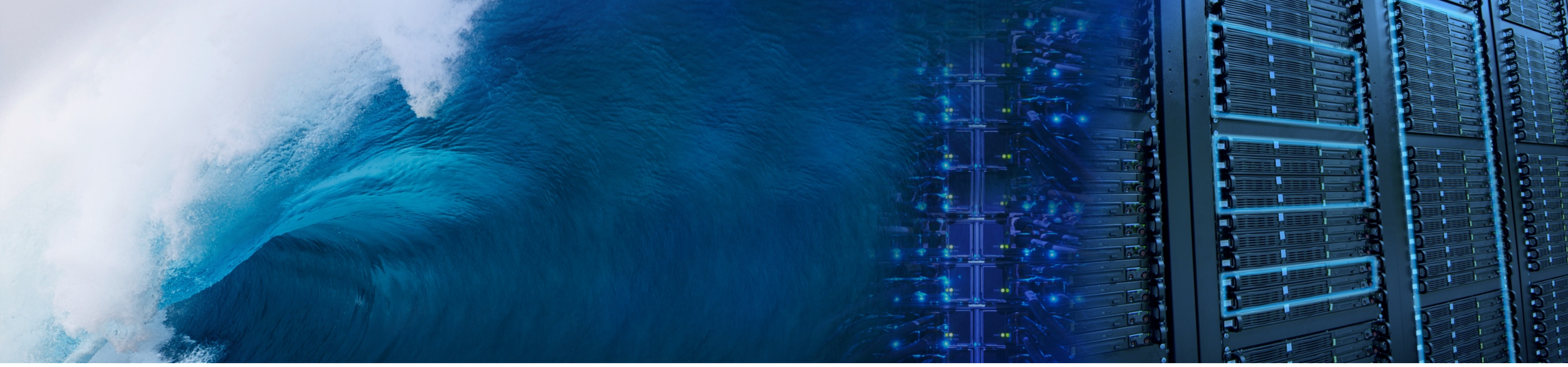
8 Summary: The New Economics of Computing at Scale

At TidalScale, we have tackled a what is widely considered to be a difficult problem: virtualizing across multiple hardware components to satisfy a simple request – "give me a bigger computer to attack my problems." In the process, Software-Defined Servers bring much-needed flexibility to organizations seeking to make more efficient use of existing resources, whether this is to solve big data problems or handle constantly changing workloads.

Above all, the resulting simplicity opens the door to any large-scale application that would benefit from in-memory operation:

- Scaling Up at the cost of Scaling Out, namely, adding hardware incrementally, as needed.
- Dealing with 1/5 or 1/10 the number of systems at the application level.
- Reducing the risk of outgrowing the system.
- Running today's clustering technologies more efficiently.
- Decoupling from the arc of the most capable leading edge (and most expensive) servers.
- Extending the practical life of modern servers already deployed.

The most enduring solutions are sophisticated answers to deceptively simple questions, and TidalScale has taken that challenge in the huge arena of big data analysis and data center flexbility, where the benefits of simplicity scale with the problems.



TidalScale

Software-Defined Servers