INTRODUCTION TO WEB SCALE IT

Web scale IT is a relative newcomer to IT infrastructure and architecture, but it changes everything we know about scale, agility, and flexibility for companies.

In this article, I'll tackle web scale IT, including:

- Concepts & components
- Web scale vs hyper-converged
- How & why to switch to web scale IT

What is web scale?

Web scale IT, or web scale infrastructure, is the technology of converged architectures with very flexible, scalable, fault-tolerant software that can run on standard x86 hardware. This combination of characteristics allows you to integrate various infrastructure components—computation, storage, virtualization, and networking—into one platform or appliance. By aggregating resources and centralizing management, you will:

- Increase efficiency and flexibility
- Minimize maintenance

Global enterprises working across large scales have implemented (essentially created) web scale IT to change the way their IT teams work and maintain speed, agility, and scalability.

The best examples are web giants like Google, Facebook, and Amazon, but smaller players are starting to get in on the action, too. Gartner first introduced the term in 2013 with high hopes for its rapid growth within five years. Although the uptake has been slower than anticipated, its steady growth will likely continue into the future.

Web scale characteristics

Web scale doesn't refer to one single technology; rather, it represents an infrastructure formed by a set of technologies and capabilities that large companies have successfully implemented at global scales.

The methodology and approach of web scale infrastructure are unique in that they primarily only work in large environments, fostering different ways of thinking compared to more traditional approaches. Driving the methodology are properties and components that focus on:

- Convergence
- Distribution
- Automation
- Error recovery and prevention

Methodology & approach

The web scale architectural approach focuses on designing, building, and managing data center and software infrastructure in a way that tailors systems to global computing and large environments.

The nature of the internet and global computing means modern architectures tend to grow at very fast rates. This, in turn, creates bottlenecking. To avoid scale limitations, web scale methods prioritize the efficient scalability of your infrastructure for these key components:

- Speed and agility
- Consistency
- Versioning
- Tolerance

Tolerance, in particular, is key to helping operators identify and fix issues more quickly, thus preventing bottlenecking and enabling faster deployment. Web scale's open environment fosters tolerance by facilitating:

- Standardizing protocols
- Identifying issues
- Creating a unified stack for efficient communication

Making improvements in these areas often requires some customization to help suit specific business requirements. Doing so allows you to:

- Tailor your system to your needs and budget
- Prevent getting locked-in with vendors

Properties & components

Web scale infrastructures are software-based, meaning that everything is on software running on standard x86 hardware without any accompanying specialized hardware performing single tasks.

This pairs well with the need for being able to expand while functioning as a cohesive unit—rather than relying on several deployments of multiple units that are not individually scalable.

That said, web scale cannot upgrade all components in one go due to the immense size of the system. That means that certain features are crucial for web scale systems:

- Self-defining and versioned objects
- Self-describing and version-aware services

This allows the encoding and serialization of structured data as well as communication between the various parts of the distributed system, all without the expectation of upgrading all components at once.

The huge size of web scale architecture also leads to the human-to-machine ratio heavily skewing in the robots' favor. Such disparities mean you should implement analytics and automation software to reduce human responsibilities and interactions.

To aid this automation, web scale architecture should include programmatic interfaces working off HTTP-based services. These interfaces should use latency, loss-tolerant protocols, and asynchronous request responses to grant complete control and automation.

And finally, to protect against single points of failure and bottlenecks, the architecture should include failure tolerance considerations that can address problems as quickly as possible. Some techniques for accomplishing this goal include:

- Consensus algorithms
- Rate limiting
- Multiple replicas
- Two-phase commit

Hyper-converged vs web scale

One word you're likely to hear to describe web scale IT is hyper-converged. Although web scale infrastructures are converged, they still have significant differences from hyper-converged systems.

The first difference is that hyper-converged architectures replicate between machines across systems to provide reliable hardware abstractions. As web scale architectures are software-based, they use custom software abstractions provided by applications to build reliability. These software abstractions are more ideal for scalability than hyper-converged hardware abstractions that have strong consistency requirements.

The hardware that web scale does use is also frequently customized—the opposite of mass-produced commodity hardware that hyper-converged systems tend to use. The companies employing web scale are so large that they can afford to pay the higher upfront costs of these customizations in order to reap the rewards of performance gains (and eventual cost savings in the form of lower maintenance) in the future.

The relationship between the software and any existing hardware is also quite different in the two models:

- Hyper-converged architectures typically separate the two.
- Web scale fits them together in a tightly coupled data center design.

This fits into the customization theme, where such coupling helps the infrastructure, hardware, and applications to work optimally in a specific environment.

Within the software, hyper-converged architectures also co-locate storage and compute services rather than separating them into different services, as web scale typically does. Although web scale can place compute and storage together, it still divides the applications into microservices that are separated by the network.

Harnessing web scale IT

So why has the uptake of web scale infrastructure been slower than expected? Part of the problem is that it's intimidating!

Making this change requires IT teams alter their entire way of working. If an IT group hasn't fully embraced the idea of making the switch, they might not make the necessary effort when it comes to web scale implementation. While incremental changes can be helpful in some situations, web scale implementation is better suited to big leaps, leaving all old ways behind.

Some of the biggest changes will include a heavier focus on architecture and design rather than maintenance. IT teams will need to constantly think about and work on improving:

- Compute
- Storage
- Growth horizons
- Failure recovery
- Automation

Automation in particular is a crucial aspect of web scale, so the IT teams should be comfortable with implementing artificial intelligence and machine learning in appropriate contexts. You'll end up spending more time on automation—and less time on network architecture or operations.

This situation might be the opposite of what most network operators are used to, so the change could feel particularly drastic for them. But with automation in place, the team can then:

- Focus more on managing network devices
- Reduce time spent on debugging.
- Minimize or eliminate human error in the ever-growing system

Getting started with web scale

Think you're ready for web scale?

Ease your IT team into web scale by building the infrastructure on the side. Then, progressively push new applications onto it.

To encourage full implementation, communicate thoroughly, reaching the entire organization and emphasizing the importance of the switch. Highlight the benefits the change will bring in the long term—and that it's likely all or nothing. In the long run, benefits of switching to web scale will include improved performance, efficiency, resiliency, flexibility, and cost savings.

The actual switch is daunting and will require concerted effort all around, but it's the way of the future. Keeping up with the big players—and getting ahead of them—hinges on embracing web scale architecture.

Related reading

- BMC IT Operations Blog
- Converged vs Hyper-Converged Infrastructure
- Monolithic vs Microservices Architecture (MSA)
- State of the Mainframe in 2020
- Lewin's 3 Stage Model of Change Explained