AMPERE®

Web Services on Ampere® Altra®

Real world solutions using popular Cloud-Native applications on

Ampere® Altra®

Ampere® Empowering the Future

Web services are built using cloud-native applications working together to deliver content over the Internet. They are increasingly built using microservices and can be easily deployed, managed, and scaled using a containerized environment such as Kubernetes. Cloud-native processors such as the Ampere Altra Max allow web service workloads to scale out using multi-node clusters and run in a predictable manner with minimal variance under increasing loads while delivering exceptional energy efficiency.

Cloud-Native Advantage

Cloud-native is a modern approach to building and running software applications that makes use of the flexibility, scalability, and resilience of cloud computing. More and more developers are embracing cloud-native microservices-based architecture to develop and deploy applications such as web services to the cloud.

The web services used here simulate real-world services using many of the popular cloud-native applications such as NGINX, Redis, Memcached, and MongoDB. These applications run as micro-services using a Kubernetes cluster with Docker containers.

Key Benefits

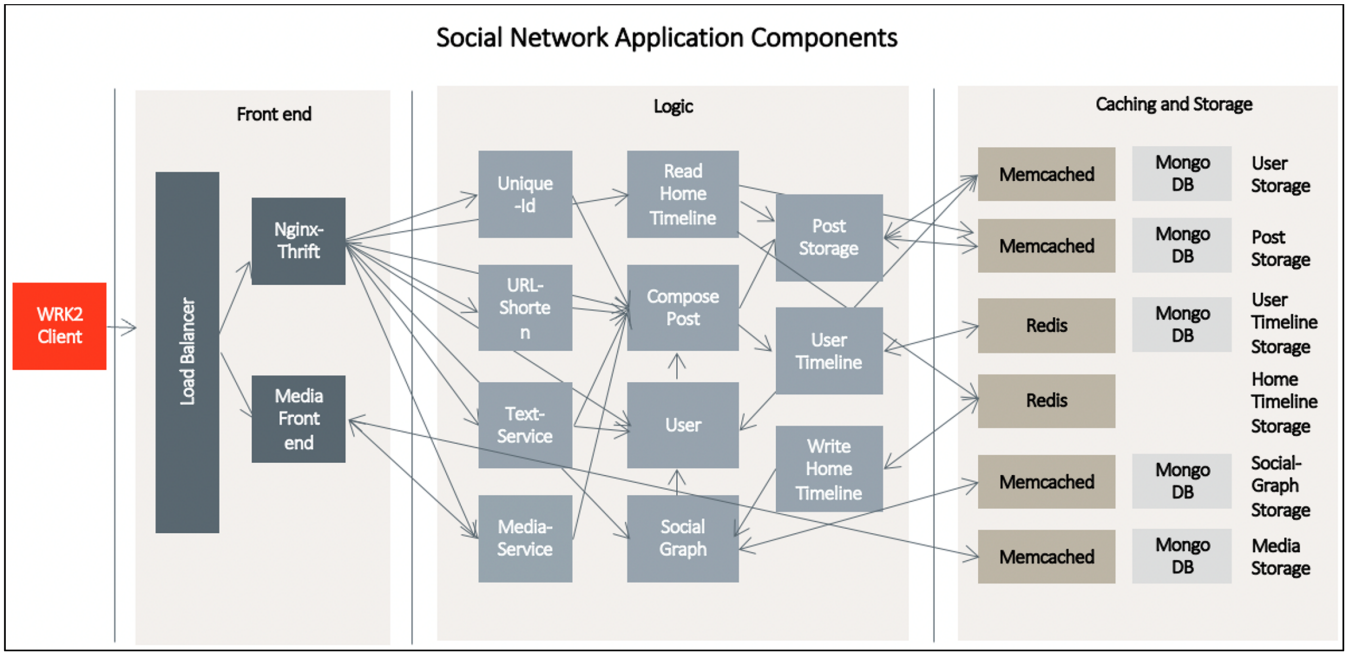
- **Cloud-Native**: Ideal for web services that use common cloud-native applications and are deployed in a containerized environment.
- **Scalable**: Predictable performance for web services even under high utilization. Real world web services simulated on Ampere systems show better throughput and lower latencies as compared to legacy x86 platforms.
- **Developer Friendly**: Robust ecosystem of applications supported on Ampere processors and ease of application portability to aarch64.
- **Power Efficient**: Competitive levels of raw performance while consuming less than half the power compared to the competition.

Replicating Real-World Web Services

Web Services showcased here are deployed on a 3-node cluster of Ampere Altra Max servers. The load generator is a client workload that simulates multiple simultaneous connections. The demo uses a Kubernetes deployment with the applications running as a collection of multiple Docker containers. The number of replicas of each pod, CPU and memory allocation for the pods is tuned to achieve the lowest P99 latency and highest throughput. At the end of the test, the load generator results indicate the average and 99th percentile of latency or response times for all client connections as well as the throughput measured in terms of requests per second.

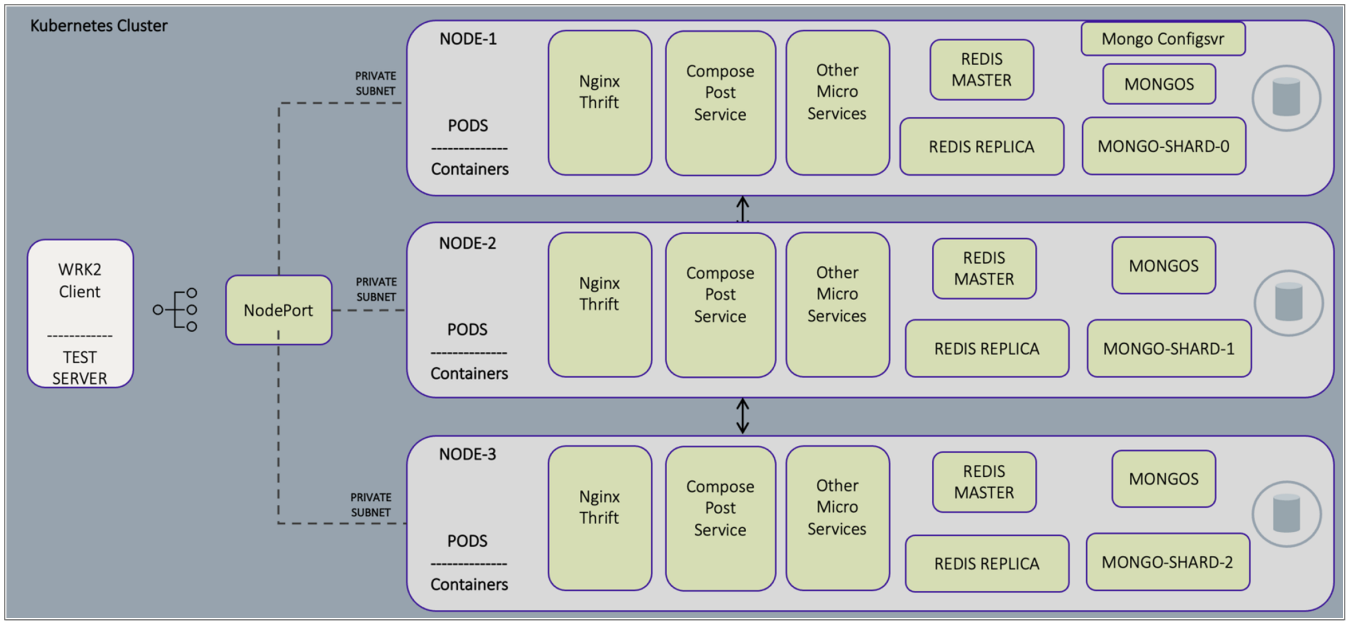
DeathStarBench Social Network Application

The DeathStarBench Social network application running Ampere Altra Max servers simulates a Twitter-like application running at scale with thousands of users connecting to the front-end using http, composing posts, tagging other users, adding media or URLs to the posts, and saving to the user and home timelines.



Scalable Deployment Using Kubernetes

The Social network application can be easily deployed on a Kubernetes cluster using helm charts. This allows the application to scale multiple servers as the load on the service increases. The front-end and application layers of DeathStarBench/socialNetwork use micro-services and are run as Pods on a Kubernetes cluster allowing multiple replicas of the Pods to be deployed. Database scaling for the back-end caching layers (Redis, Memcached) and database (MongoDB) is implemented using clustering and sharding. The goal is to benchmark the performance of a real-world social network website and determine the peak performance using a load testing service WRK2 which emulates a large numbers of users visiting the website and evaluating how well the service responds under these stressful conditions.



For additional information, visit the Ampere Solutions Portal.

Ampere Computing® / 4655 Great America Parkway, Suite 601 / Santa Clara, CA 95054 / www.amperecomputing.com