Solution Brief

AI Specifiers and Edge Network Engineering Specifiers
Edge-to-cloud AI Service

Zeblok Automates Scaling of Hub-to-Edge AI with Low Overhead

To support AI-enabled devices at the network edge, Zeblok's Ai-MicroCloud®, running on Intel-powered Supermicro servers, can provision 1,000 edge servers efficiently with full life-cycle management using minimal compute overhead

As edge-based computing continues to gain traction, deployment of artificial intelligence (AI) and machine learning to automate deployment and ongoing management of servers at the edge is a natural result.

Worldwide spending on edge computing is expected to be $208 billion in 2023, an increase of 13.1% over 2022, according to the International Data Corporation (IDC) Worldwide Edge Spending Guide. The analyst firm also predicted that enterprise and service provider spending on hardware, software, and services for edge solutions is forecast to sustain this pace of growth through 2026 when spending will reach nearly $317 billion.[1] The Edge artificial intelligence market specifically is expected to hit $66.47 billion by 2030.[2]

At the network edge, AI computing occurs on physical devices close to the user—and data sources—rather than in a centrally located cloud or data center. Network edge locations can range from retail point-of-sale locations to machines in factories. When data can be processed close to where it is generated, AI-enabled edge devices can be more efficient in terms of cost, latency, throughput, privacy, and resiliency.

As the growth of AI at the edge compounds, so does the challenge of large-scale deployment of AI-enabled devices. Humans can handle replicating a manual business process from one site to another. But scaling the provisioning of that process to thousands or tens of thousands of locations or endpoints is simply not feasible in a manual fashion.

One way to address this is with infrastructure provisioning using infrastructure-as-code. This approach offers automation, versioning consistency, and lighter storage loads.

Developers don't need to manually provision operating systems, storage, and other infrastructure components each time they stand-up a new edge location. Codifying the infrastructure provides a template to follow for provisioning. Although this can still be accomplished manually, an automation tool can help scale more quickly.

Building and running AI or machine learning applications at the edge of the network is a complex exercise in distributed systems engineering. And the challenge is often one of operations and scale. Intel® Network Builders ecosystem member Zeblok provides a solution to help customers scale infrastructure deployments at the edge in an automated fashion while employing AI capabilities.

Automated Edge Server Deployment

Zeblok's Ai-MicroCloud® provides an end-to-end solution to scale and deploy AI microservices on edge servers in large numbers. The software allows a network engineer to scale AI services on edge servers in just a few clicks using very little memory and compute overhead as will be presented later in this paper. Industries such as retail, fast food, Industry 4.0, telecom, energy, oil and gas, in fact, almost every industry can deploy AI services at the edge with ease.

Ai-MicroCloud® provides a single, cohesive, turnkey, cloud-native AI environment for development, testing, training, optimization and deployment of an end-to-end AI or machine learning solution to production from the cloud to the edge. Many enterprises use an industry benchmark of two days for a single engineer to create a cluster (three hubs and three edge servers). Using Ai-MicroCloud®, the typical enterprise might be able to deploy thousands of edge servers in a comparable timeframe (see Table 1).

Ai-MicroCloud® introduces multi-tenancy on edge servers by offering workflows for machine learning (ML) operations teams to deliver AI solutions from independent software vendors (ISV) into separate namespaces while offering full lifecycle management capabilities. This architecture introduces flexibility and vertical scaling within the edge servers at each edge location. Hence, AI ISV assets such as models, algorithms, and applications from multiple parties can be delivered on a single edge server. Ai-MicroCloud® has low overhead that allows for expanded density at the edge for value-generating workloads.

Ai-MicroCloud® works for both hub and edge servers, that can be either physical or virtual. The recommended resources to run Ai-MicroCloud® are 4 CPU cores, 4 GB RAM, and 100 GB disk space for each hub server. Three servers are recommended in hub locations to provide redundancy for high availability. The recommended Ai-MicroCloud® resources for an edge server are 4 CPU cores, 4 GB RAM, and 20 GB disk space just for the Ai-MicroCloud® platform. Both the hub and edge servers can have higher resources as per workload requirement.

To show the performance of Ai-MicroCloud®, Zeblok worked with Intel® on three tests of the solution to show a full range of performance characteristics.

Test I - Horizontal Scaling

Horizontal scaling tests will show that the hub server has significant CPU and memory head room. As the number of the attached edge servers grow, the hub server must retain capacity for other applications, such as running data lake components, bringing back inference results, driving datasets for continuous model training, understanding model drifts, and running other AI ISV solutions that require a level of processing that is typically not available in edge servers.

Horizontal Scaling Test Setup and Methodology

To demonstrate the scalability of Ai-MicroCloud®, the company conducted testing that showed the hub server's ability to deploy 1,000 edge servers. The test setup consisted of 1,000 virtual machine edge servers connected to three physical hub servers.

Out of the three hub servers, one is active and the other two are on standby. When the active server is disabled, the next standby takes over, ensuring redundancy and availability. Additionally, these servers host Zeblok's application manager, data lake, and other supporting components. The hub servers can also host other software for model training and heavy data processing at the hub.

Three hub servers powered by Intel® processors and Ethernet network adapters, were configured as follows:

- **Hub Server (zs47)** runs with an Intel® Xeon® Platinum 8260M CPU @ 2.40GHz, 384 GB of RAM, dual socket, 24 cores and a two port Intel® Ethernet Connection X722 for 10Gb network card.
- **Hub Server (zs16)** runs with an Intel® Xeon® Platinum 8280M CPU @ 2.70GHz, 192 GB of RAM, dual socket, 28 cores and a two port Intel Ethernet Connection X722 for 10Gb network card along with eight port Ethernet Controller XXV710 for 25GbE SFP28 network card.
- **Hub Server (zis13)** runs with a single Intel® Xeon® Gold 6338N CPU @ 2.20GHz, 512 GB of RAM, dual socket, 32 cores and a two port Intel Ethernet Connection X722 for 10Gb network card along with four port Ethernet Controller XXV710 for 25GbE SFP28 network card.

The 1,000 edge servers were created as virtual machines on servers running VMware ESXi and having hostnames zss07 and zss08 (aside from generating the edge servers, these machines did not have an impact on the performance tests). The decision to test 1,000 servers was determined by the test bed configuration and is not the upper limit of Ai-MicroCloud® capacity.

Each virtual machine was configured with 4 GB of RAM, 2 CPU cores, and 20 GB of disk space. All the servers were running Ubuntu 20.04.4 operating system on the 5.4.0-124-generic kernel.

In addition to the three hub servers and 1,000 virtual edge servers, two jump servers were used to trigger edge installation automation scripts and performance benchmarking tools.

To conduct the testing, the three hubs and 1,000 edge servers were connected through manually run test scripts. The tests used 150 Kubernetes pods of JupyterLab, a microservice testing app, spawned on different edge servers.

Testing also demonstrated simulation of multi-tenancy on a per edge node by running containerized applications such as MLflow and Nginx pods, which are placeholders and can be replaced by containerized workloads from ISVs. Ai-MicroCloud® delivers this software to the edge servers as microservices or industry standard APIs.

Significant Time Saving

Table 1 shows the time required for each major part of the configuration with hub and new edge server deployments taking 40 minutes and 25 minutes to add additional servers to a cluster. The installations tested for this paper were done in parallel.

Type of Set Up	Component	Time
Cluster (hub and edge)	Operating system	20 min
	Orchestration Manager	15 min
	Ai-MicroCloud® Manager	5 min
Adding edge nodes to existing cluster	Operating system and edge cloud	25 min

Table 1. Estimated time for each aspect of hub and edge server deployment.

As will be seen from the test results, Ai-MicroCloud® can potentially save thousands of employee hours in preparing the edge servers. The physical servers can be staged and provisioned and then sent to their destination or drop shipped from the server vendor and updated over the Internet automatically.

Furthermore, 10 MLflow microservices were spawned on different edge servers through Ai-MicroCloud®. Load testing was completed with the URLs of five MLflow microservices spawned. Additionally, 10 NGINX pods were spawned on different edge servers. Load testing was completed with URLs of five NGINX pods spawned by applying load/request on those pods.

The edge servers were connected in incremental order to the hub servers and resources were monitored on the hub nodes. As the three nodes in the hub work in active standby mode, the readings were only taken from the active node. Cluster performance was observed, and benchmarking results noted.

Test Results

Overall, testing demonstrated the low overhead that Ai-MicroCloud® generates on Supermicro configurations leaving room to deploy additional workloads.

As shown in Figure 1, a linear relationship exists between the CPUs needed and the edge servers deployed. The chart shows a one percent increase in CPU utilization was found when the number of nodes were increased from 200 to 300. No major changes were found in memory usage as seen in Figure 2.

As the tests reached 1,000 connected edge servers, CPU utilization only increased moderately from just over two cores to eight cores.

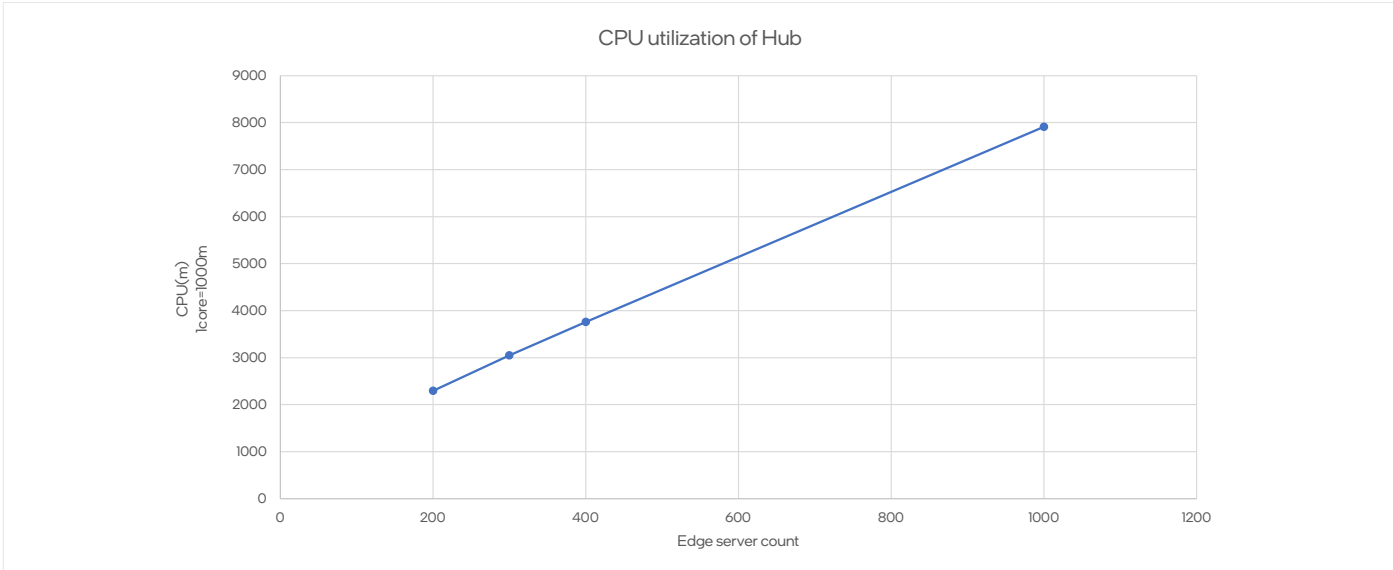


Figure 1. Number of virtual edge servers deployed and CPU utilization of hub servers (higher is more resource efficient).

Scaling the edge servers up to 1,000 or more occurs with very little CPU utilization on the hub, allowing hub servers to simultaneously run other applications and handle data processing. The Ai-MicroCloud® solution does not take up headroom and has a small footprint at the edge. Two to three percent of the CPU cores can utilize other applications.

Figure 2 depicts the number of servers on the X axis and the amount of memory in GB utilized on the Y axis. At 200 edge servers, memory usage is just below 12.05 GB. When the server count increases five-fold to 1,000 edge servers, memory usage increases only a few megabytes to just over 12.3 GB.

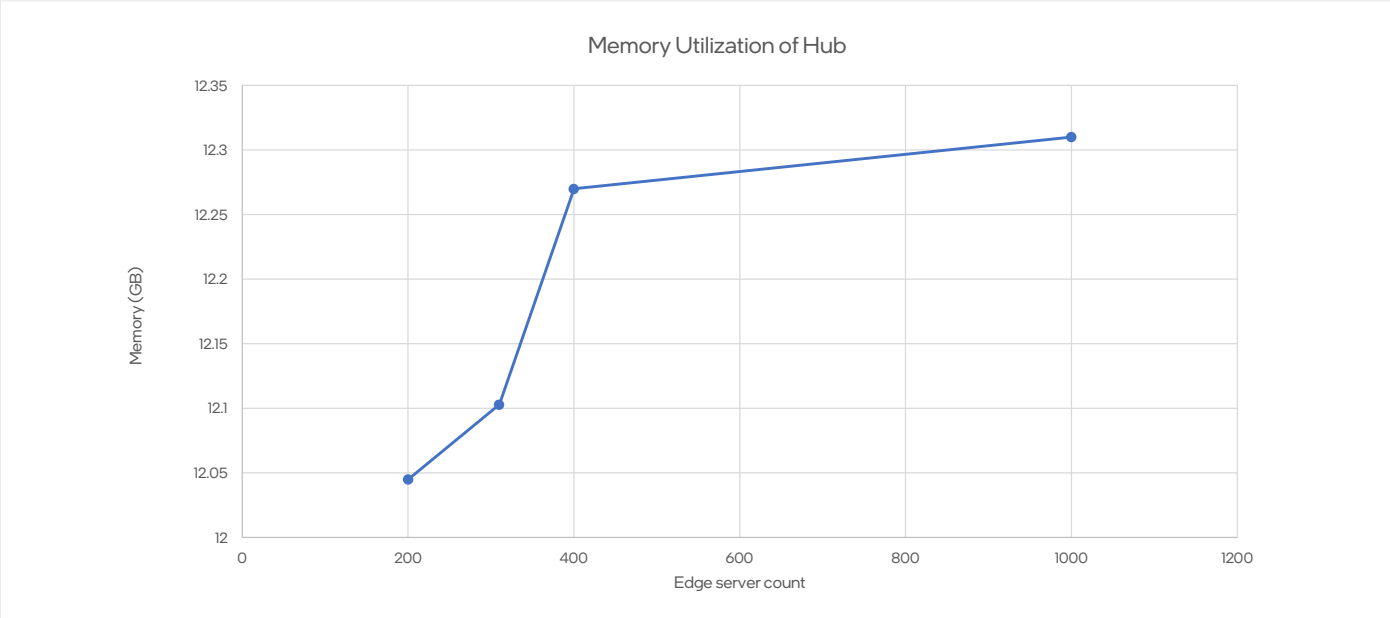


Figure 2. Number of virtual edge servers deployed and memory utilization of hub servers (higher is more resource efficient).

For simulation, Nginx pods were deployed on random edge servers and http traffic was generated on the edge server's physical IP address. NGINX simulated web or content driven workloads on the edge servers. Figure 3 shows that, as the number of http requests per second by the client increased, the CPU utilization of the NGINX pod increased.

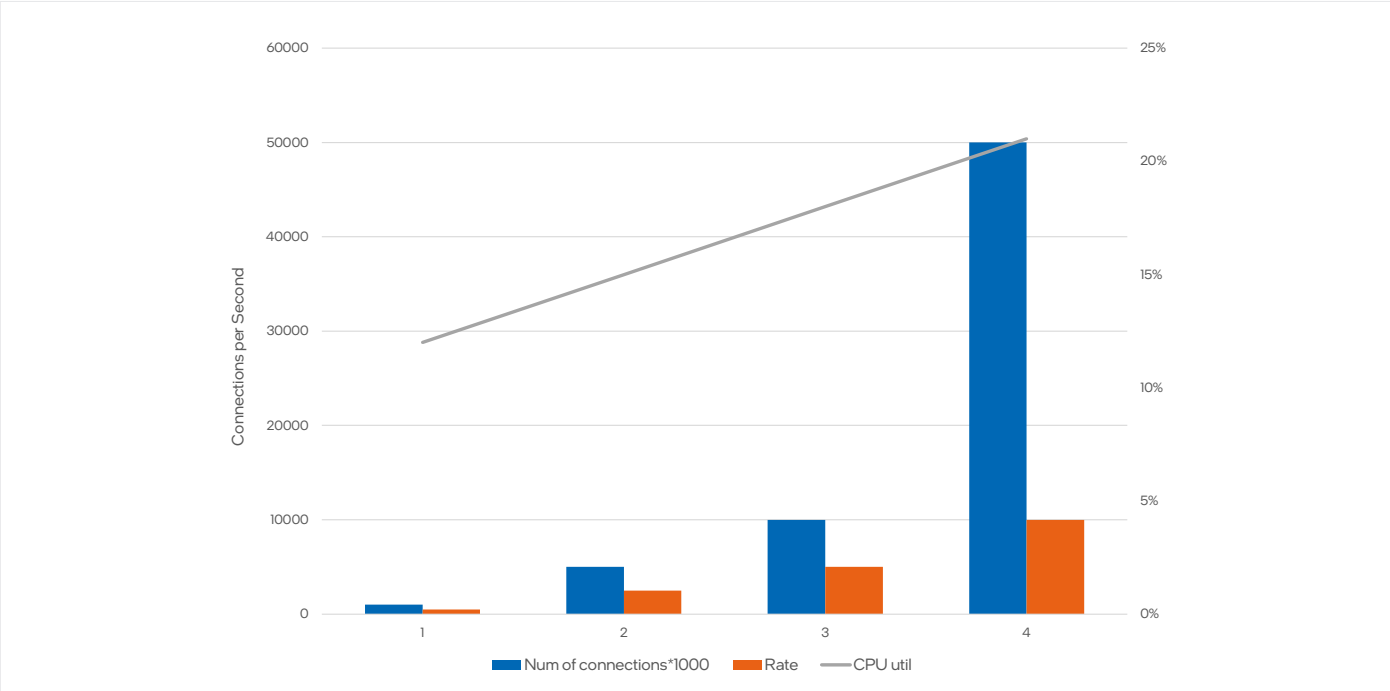
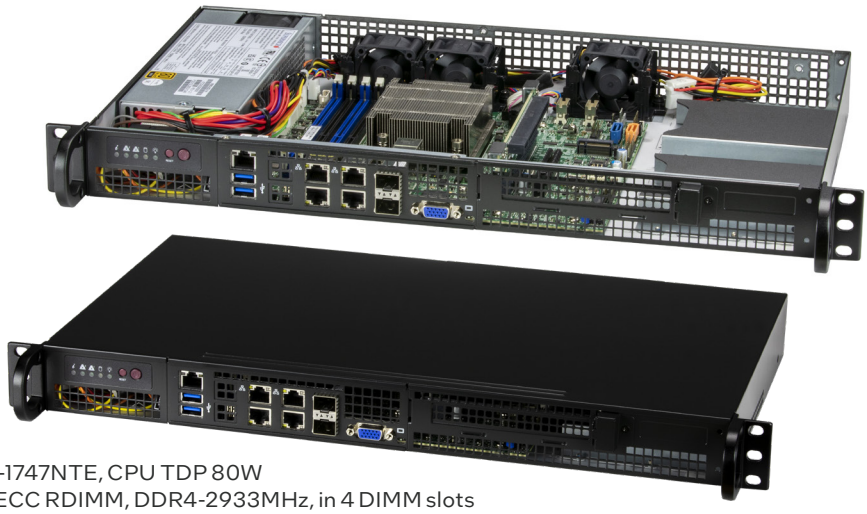


Figure 3. Number of edge connections and client http requests per second and CPU utilization.



Key Features

1. Intel® Xeon® Processor D-1747NTE, CPU TDP 80W
2. Up to 256GB Registered ECC RDIMM, DDR4-2933MHz, in 4 DIMM slots
3. 4x GbE and 2x 25G SFP28
4. 1x Internal 3.5'' or 4x Internal 2.5'' drive bays
   1x M.2 M-Key 2280 (SATA/PCIe 3.0 x4) 1x M.2 B-Key 2280/3042/3052 (USB/SATA/PCIe 3.0 x2) 1x M.2 E-Key 2230 (PCIe 3.0 x1)
5. 3x 40x28mm 4-PIN PWM fans
6. 200W Low-noise AC-DC power supply
7. 1x VGA, 2 USB 3.0

Figure 4. Supermicro IoT SuperServer SYS-510D-10C-FN6P that was used as an edge in this testing.

Test II - Vertical Scaling

These tests demonstrate that there is significant CPU and memory head room in the edge servers.

The low resource utilization of Ai-MicroCloud® at the edge provides compute head room for AI workloads. The edge hardware resources can run either compute-heavy workloads or multiple workloads.

The configuration of the systems under test are listed here:

Hub server systems under test (SUT):

- Supermicro IoT SuperServer SYS-110D-20C-FRAN8TP (zdh01) features an Intel® Xeon® D-2796NT CPU @ 2.00GHz, 128 GB of RAM, single socket, 20 cores and Intel® Ethernet Server Adapter I350.
- Supermicro IoT SuperServer SYS-110D-20C-FRAN8TP (zdh02) features an Intel® Xeon® D-2796NT CPU @2.00GHz, 128 GB of RAM, single socket, 20 cores and Intel® Ethernet Server Adapter I350.
- Supermicro Ultra SuperServer SYS-620U-TNR (zi02) features dual Intel® Xeon® Gold 6330N CPUs @ 2.20 GHz, 384 GB of RAM, dual socket, 28 cores and Intel® Ethernet Converged Network Adapter X540-T2 network card.

Figure 5. Supermicro IoT SuperServer SYS-110D-20C-FRAN8TP used as a hub server in this testing.

Edge servers:

- Supermicro IoT SuperServer SYS-510D-10C-FN6P (zdl01) features an Intel® Xeon® D-1747NTE CPU @ 2.50GHz, 128 GB of RAM, single socket, 10 cores and Intel® Ethernet Controller I210.
- Supermicro IoT SuperServer SYS-510D-10C-FN6P (zdl02) features an Intel® Xeon® D-1747NTE CPU @ 2.50GHz, 128 GB of RAM, single socket, 10 cores and Intel® Ethernet Controller I210.

Test Result:

The tests were performed to monitor CPU and memory usage of the edge servers after the installation of Zeblok Ai-MicroCloud®. These tests demonstrated very low CPU utilization, on the order of 0.5%. Memory utilization was also low at around 700MB on the edge server, leaving a lot of compute and memory head room for the AI workloads.

An additional test was performed to showcase multi-tenancy using Ai-MicroCloud®. Two AI-API pods running Intel® distribution of OpenVINO™ pre-trained Resnet50v2 model were deployed by two different ML ops users in two different namespaces on a single edge server(zdl01). In this case, 0.65% CPU utilization was observed and 1.93 GB of memory utilization was observed.

Test III - Application Scaling

These tests were designed to show scalable deployment of AI at the edge with Ai-MicroCloud®

Test Methodology

To simulate a real-world application, the test team then ran the 'Person Vehicle Detection' AI model from the Intel® Distribution of OpenVINO™ Model Zoo on one of the edge servers describe in Test II with the following setup to monitor its resources. Workflows in Ai-MicroCloud® package AI inferences and applications as API and microservices. We used a combination of APIs microservices. Here are the details:

- The AI-API was the OpenVINO model called PersonVehicle Detection trained model.
- Real Time Streaming Protocol (RTSP server is where the FFmpeg video platform publishes its RTSP stream continuously which is used for the simulation of a real time camera stream.
- FFmpeg is used for pulling off an image from the video, through S3, and playing it in a loop so that a never-ending video stream is sent to RTSP.
- Python-stream-client is used as an aggregator that takes RTSP server port for the stream and forwards it as camera stream to AI-API for inference and displays the response inference frames.

Ai-MicroCloud® workflows allocated resources to microservices. Multiple python-clients were spawned which emulated as streams coming from CCTV cameras. Table 2 shows the resources dedicated to each software component.

In a real-world application, ISVs would be able to containerize their applications within the Ai-MicroCloud®.

Test Results

Some details on the software platforms:

1. Model Zoo model inference was not modified or optimized.
2. SUT utilizes older Python versions.
3. Python-client generates 12 frames per second (FPS) stream and is not optimized

After running the above microservices, the below resource utilization reading was observed Inference as API services all requests coming from 14 different camera feeds. Each camera feed is simulated by python-client.

The test showed that the time to deploy a microservice from the AI AppStore to the edge servers is less than 1 minute. Additionally, the tests showed that the time needed to deploy an open source docker image on the edge was less than 10 minutes.

The interconnected set of microservices clearly decouple workloads on the edge server providing a scalable, infrastructure-as-a-code based architecture with public cloud like experience for the edge servers. It also provides a scalable software architecture for ISVs to deliver components that can be independently optimized, packaged, scheduled, and scaled. Carefully selected components, from open-source environments, can be easily brought in and delivered to the edge servers significantly boosting the productivity of modelers and ml ops personnel and improving performance.

Multi-tenancy is simulated in this test through a variety of workloads delivered to the edge servers (AI-API, AI-Microservice, Non-AI-Microservice) as well as through namespaces. These workloads could potentially come from multiple ISVs or departments within an enterprise. The tests show that the overhead on each edge server is minimal leaving enough capacity for new workloads to be introduced into the edge server. Continuous monitoring and alerting further adds to full lifecycle management of workloads delivered within the Ai-MicroCloud®.

Some of the other ISV workloads could be timeseries analytics, deep neural network-based inference from long short-term memory (LSTM) for timeseries or for industrial IoT applications. Another possibility is a lightly trained conformer model that serves as inference for voice processing on edge server, perhaps running on kiosk at a quick service restaurant chain. Ai-MicroCloud® provides for full lifecycle management of such solutions including packaging and delivery of software while bringing multi-tenancy to the edge server and establishing future proofed rails for deployment.

COMPONENT	CPU	RAM	STORAGE	EDGE SERVER
AI-API	4 vCPU	4 GB	30 GB	zdl01
Rtsp	4 vCPU	4 GB	30 GB	zdl02
Ffmpeg	1 vCPU	1 GB	5 GB	zdl01
Python-client	1 vCPU	1 GB	5 GB	zdl01

Table 2. Resources dedicated to each software component.

Conclusion

Large-scale deployment of AI-enabled devices is a challenge. With infrastructure-as-code, developers don't need to manually provision the server each time they stand-up a new edge location. Zeblok's Ai-MicroCloud® provides an end-to-end solution to scale edge servers in large numbers and deploy AI microservices on every edge server, all while using very little memory and overhead. Zeblok tested resource utilization of Intel® architecture-based Supermicro edge servers after deploying Ai-MicroCloud® using a test set-up of 1,000 virtual machine edge servers connected to three physical hub servers. Testing demonstrated the low overhead that Ai-MicroCloud® generates on Supermicro configurations, leaving room to handle additional workloads.

Learn More

Zeblok Ai-MicroCloud®

Supermicro IoT SuperServer SYS-510D-10C-FN6P

Intel® Xeon® Platinum 8260 Processor

Intel® Xeon® Gold 6338N Processor

Intel® Ethernet Network Adapter X722

Intel® Ethernet Network Adapter XXV710

Intel® Network Builders

OpenVINO™ Person Vehicle Bike Detection

Python Stream Client

Supermicro Ultra SuperServer SYS-620U-TNR

[1] https://www.idc.com/getdoc.jsp?containerId=prUS50386323
[2] https://www.grandviewresearch.com/industry-analysis/edge-ai-market-report

0623/DC/H09/PDF ♻Please Recycle 355187-001US