



Sizing Service Engines

Avi Technical Reference (v18.2)

Copyright © 2020

Sizing Service Engines

[view online](#)

Overview

Avi Networks publishes minimum and recommended resource requirements for new Avi Service Engines. However, network and application traffic may vary. This guide provides some guidance on sizing. You can consult with your local Avi sales engineer for more recommendation that is tailored to the exact requirements.

Note for deploying into AWS: The [Installing Avi Vantage in Amazon Web Services](#) article provides additional sizing guidance unique to the AWS environment, including [this link](#) to Amazon's own guidance on various maxima.

Avi Service Engines can be configured with as little as 1 vCPU core and 1 GB RAM, or up to 64 vCPU cores and 256 GB RAM. In write access mode, Service Engine resources for newly created SEs can be configured within the SE Group properties. When creating an Avi SE in read or no access modes, Avi SE resources are allocated manually by an administrator via the hypervisor or by the size hardware used for bare metal servers.

CPU

CPU scales very linearly as more cores are added. Note that Avi Vantage cannot differentiate between a physical core and a hyperthreaded core, which generally provides reduced performance compared to a physical core. CPU is a primary factor in SSL handshakes (TPS), throughput, compression, and WAF inspection. For vCenter clouds, the default is 1 CPU core, not reserved. However, CPU reservation is encouraged.

Memory

Memory scales near linearly. It is used for concurrent connections and HTTP caching. Doubling the memory will double the ability of the Avi Service Engine to perform these tasks. The default is 2 GB memory, reserved within the hypervisor for VMware clouds. Refer to [SE Memory Consumption](#) for more details.

PPS

For throughput-related metrics, the hypervisor is likely going to be the bottleneck. Limitations are different for each hypervisor and version. For instance, VMware's vSphere/ESX 5.x supports about 550k packets per second to a virtual machine such as Avi's Service Engine. ESX version 6 doubles this number to around 1.2M PPS. Bare metal (no hypervisor) numbers depend on the type of NIC used. For DPDK mode, 82599, X520, X540, X550, X552, X710, and XL710 are the supported Intel NICs. DPDK can be used in some environments to bypass PPS limitations of the hypervisor and provide line rate speeds.

RPS

RPS is dependent on the CPU or the PPS limits. It indicates the performance of the CPU and the limit of PPS that the SE can push. The following table provides the value of RPS per core for different ecosystems:

Ecosystem	RPS per core
GCP	~30k
Gen 1 VM	
VMware	~40k
v3 intel servers	
Baremetal	~40k
v3 intel servers	
AWS	~40k
c5 instance family	
Azure	
FS_v2 instance family	~30k
DS_v2 instance family	~20k
CSP	~60k
CSP5K hardware	

On an average, the RPS per core would be around 35 to 40k.

Disk

Avi Service Engines may store logs locally before they are sent to the Avi Controllers for indexing. Increasing the disk will increase the log retention on the SE. SSD is preferred over hard drives, as they can write the log data faster. This is not a requirement. The recommended minimum size for storage is 10 GB, ((2 * RAM) + 5 GB) or 10 GB, whichever is greater. 10 GB is the default for SEs deployed in VMware clouds.

Disk Capacity for Logs

Avi Vantage computes the disk capacity it can use for logs based on equations incorporating the following parameters: * T, the SE's total disk capacity * N, the number of CPU cores * M, the main memory (RAM) of the SE * p, `log_agent_max_storage_ignore_percent`, a parameter in SE runtime properties * D, the capacity reserved for debug logs * C (reserved for cores), $2 * M * \text{MIN}$, the minimum size allocated for logs * L, the capacity reserved for client logs

Avi Vantage calculates D as follows:

$$D = p * T / 100$$

Initially, Avi Vantage sets L, the client log size, as follows:

$$L = T - D$$

Finally, Avi Vantage adjusts L and C based on the following conditions:

```
if L > C + MIN, then set L to (L - C)
  else if L > MIN, then set C to L - MIN and then set L to MIN
    else set C to 0
```