

Web Server Sizing

Precise sizing of the web server is something that is rarely done in projects. Since the web server is using few resources compared to the application server, web servers are often oversized in projects and hardly use a high percentage of available resources in production. This document aims to provide detailed sizing considerations as well as helping to better understand the behavior of the web server and additional modules that can be used to improve the performance.

Reverse proxy of incoming requests, compression of static and dynamic content and serving static resources are typical non-functional requirements delegated to a web server such as Apache Web server. Compression and SSL offloading are CPU resource intensive, while concurrency has a correlation with memory. When using Apache to cache static content, depending on your choice of disk or memory caching, this can be either IO or memory bound. More detailed considerations can be found at [Go-live - Web Server](#) and [Web Server Tuning](#).

The considerations and recommendations in this article are driven by experience with Apache web servers. They cannot be applied on other web servers although the main principals are similar.

Sizing Cores

This section provides some formulas to calculate the approximate memory and number of cores required for a web server. As a guide, web servers tend to be sized between 4GB - 16GB RAM with 4 - 8 cores depending on the type of load and non-functional requirements. Once an estimate of the number of cores and RAM have been calculated consideration must be given to how they will be distributed over server instances. At a minimum, two web servers will be required for high availability configuration. To determine the optimal configuration of web server instances it is recommended to performance test various combinations to come to the optimal configuration for your project.

The key contributing factors of Apache CPU utilization is content compression and SSL termination. Having many worker processes that require CPU time means that the number of cores limits the amount of work that can be done. Having more cores means that more worker processes can execute concurrently.

The following formula can be used to estimate the number of cores required:

$$\text{Cores} = \text{Concurrent requests/second} * \text{Task Time}$$

The following assumptions can be used and adjusted according to your project:

- The number of concurrent requests should be adjusted down if using a CDN to serve static resources for product and content related data.
- The number of concurrent requests should be adjusted down to accommodate for resources requests that should not be compressed.
- An average task time of 10ms to compress a resource request.
- The amount of CPU time taken for a simple reverse proxy is minimal compared to the processing time required to compress a resource.

The table below shows an example of the number of cores based on the concurrent request per second.

Cores	Task Time	Concurrent requests/second
8	10 MS	800
16	10 MS	1600

Sizing RAM

Apache web server memory usage can roughly be broken down to:

- Multi-process module process size.
- Use of static content caching.

- Operating System.
- Other running processes.

To estimate the total amount of memory required for Apache web server use the following formula:

```
RAM = Concurrent requests/second / (1/ Task time) * Process size + RAM allocated to OS+ RAM allocated other processes
```

The following assumptions can be used and adjusted according to your project:

- The size of the apache MPM process is typically between 10MB and 30MB.
- The number of concurrent requests should be adjusted down if using a CDN to serve static resources for product and content related data.
- An average task time of 50ms to serve 1 request.

The table below shows an example of the number of cores based on the concurrent request per second.

Total RAM	OS RAM	Task time	Process size	Concurrent requests/second
4024 MB	1024 MB	50 ms	30 MB	1000
7024 MB	1024 MB	75 ms	30 MB	2000
8524 MB	1024 MB	75 ms	20 MB	1500

Sizing Disk Space

Apache web server disk space usage can be broken down into the following categories:

- Storage of static resources
- Storage of server logs

When employing a strategy of serving static resources from the Apache web server to reduce the load on the SAP Hybris servers these static resources will need to be stored somewhere. Generally, a shared file system is used or the files are stored locally on each Apache web server. Where a CDN is used to deliver product and content related images there is no need to store the files locally on the Apache web servers. Technical static resources such as Javascript, CSS and graphical files should also be taken into account.

To estimate the amount of disk space required to store product static media use the following formula:

Calculate Static Media size

```
number of products * number of media items * average size of medias
```

Access log files on Apache web server can grow significantly depending on your traffic and the configuration of your log file format. As a starting point, an assumption of 1MB per 10k requests can be used to estimate your storage requirements. Using an example site that has 25 Page impression/second of traffic with an average of 49 static resource requests per page we can make the following calculation:

Calculate Log files

```
(Page Request + Static Resource Requests) * Page views = requests per second  
(1 + 49) * 25 = 1250 requests per second
```

Using the assumption that the access.log files grow by 1MB for every 10K requests we can calculate that the access file will grow by 1MB every 8 seconds. Using this figure we can estimate about 10GB growth per day. Log files should be rotated and compressed to reduce disk space usage. Depending on your archiving and [data maintenance](#) strategy, keeping log files for 30 days is quite common and can estimate approximately 60GB disk space to allow for some contingency.

For transient files created such as access and error log files, ensure an appropriate [data maintenance](#) strategy is planned.