

Virtualization Hardware Provisioning

This topic will address how to size the cluster when using the AtScale Virtualization layer. AtScale makes the following recommendations when provisioning your hardware for a Virtualized cluster.

Storage Systems

All Virtualization jobs will read input data from an external storage system, such as Hadoop, Snowflake, or BigQuery. It is important to place your storage system as close to the external storage system as possible, but in the case of cloud-based databases this is not possible.

For on-premise data warehouses and for Hadoop based systems we recommend the following:

- ▲ For Hadoop based systems, run on different nodes in the same local-area network as HDFS.
- ▲ For low-latency data stores, it may be preferable to run on different machines than the storage system to avoid interference.

Local Disks

While the AtScale Virtualization cluster will perform a lot of its computation in memory, it uses local disks to store data that does not fit in RAM. The local disks also preserve output between stages. As a result, AtScale recommends provisioning four to eight disks per node (configured without RAID, as separate mount points). In Linux, mount the disks with the `noatime` option to reduce unnecessary writes.

Memory

In general, The AtScale Virtualization Cluster can run well with anywhere from 8 to hundreds of gigabytes of memory per machine.

In all cases, AtScale recommends allocating (at most) 75% of the memory for Virtualization; leaving the rest for the operating system and buffer cache.

- ▲ How much memory you will need will depend on your application. To determine how much your application uses for a certain dataset size, use the following calculation:

$$\text{Concurrent Users} \times \text{Activity Rate} \times (2 \times \text{Average Row Size}) \times \text{Average Rows Per Query Returned} \times 200\% \text{ Join Factor} \times (200\% \text{ of Average Query Complexity [1-2]})$$

Where average query complexity is relative to the use of features such as Time Relative calculations, or calculations that require MultiPass.

- ▲ For example:

$$1000 \text{ concurrent users} \times 15\% \text{ activity rate} \times (2 \times 2 \text{ kb/row}) \times 50,000 \text{ rows} \times 200\% \times 200\% \times 1 = 120\text{GB RAM}$$

- ▲ Finally, note that the Java VM does not always behave well with more than 200 GB of RAM.



Note: Activity rate is the percentage of concurrent users that would execute a query at roughly the same time.

Network

When the data is in memory, a lot of Virtualization performance is network-bound. Using a 10GB or higher network is the best way to make these applications faster. This is especially true for "distributed reduce" applications such as group-bys, reduce-bys, and SQL joins. In any given application, you can see how much data Virtualization shuffles across the network from the application's monitoring UI at (<http://<VirtualizationNode>:4040>).

CPU Cores

Virtualization scales well to tens of CPU cores per machine because it performs minimal sharing between threads. AtScale recommends provisioning at least 8-16 cores per machine. Depending on the CPU cost of your workload, you may also need more: once data is in memory, most applications are either CPU- or network-bound.

- ▲ How much CPU you will need will depend on your application. To determine how much your application uses for a certain dataset size, use the following Calculation (Where average query complexity is relative to the use of features such as Time Relative calculations, or calculations that require MultiPass):

$$\text{Concurrent users} \times \text{activity rate} \times 1 \text{ cores per query} \times \text{average query complexity [1-2]}$$

- ▲ For example:

$$1000 \text{ concurrent users} \times 15\% \text{ activity rate} \times 1 \text{ cores per query} \times 1 = 150 \text{ cores}$$

- ▲ A ~5 machine cluster each machine with 32 cores and 24-32GB RAM would suffice for the above calculation.