

Basics - Distributed System

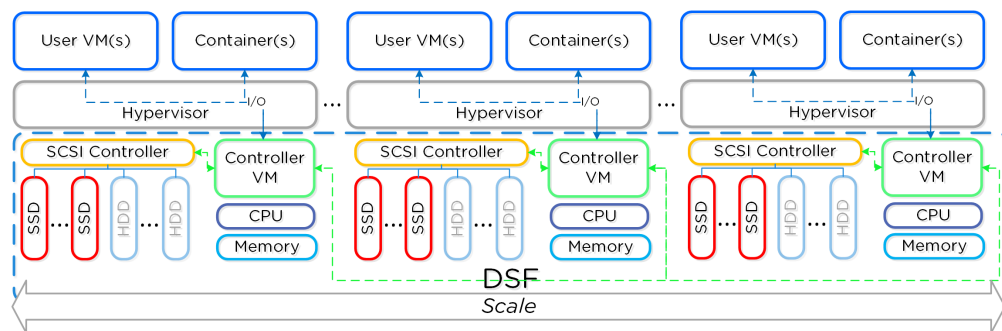
[PDF generated June 03 2025. For all recent updates please see the Nutanix Bible releases notes located at https://nutanixbible.com/release_notes.html. Disclaimer: Downloaded PDFs may not always contain the latest information.]

There are three core principles for distributed systems:

1. Must have no single points of failure (SPOF)
2. Must not have any bottlenecks at any scale (must be linearly scalable)
3. Must leverage concurrency (MapReduce)

Together, a group of Nutanix nodes forms a distributed system (Nutanix cluster) responsible for providing the Prism and AOS capabilities. All services and components are distributed across all CVMs in a cluster to provide for high-availability and linear performance at scale.

The following figure shows an example of how these Nutanix nodes form a Nutanix cluster:



Nutanix Cluster - Distributed System

These techniques are also applied to metadata and data alike. By ensuring metadata and data is distributed across all nodes and all disk devices we can ensure the highest possible performance during normal data ingest and re-protection.

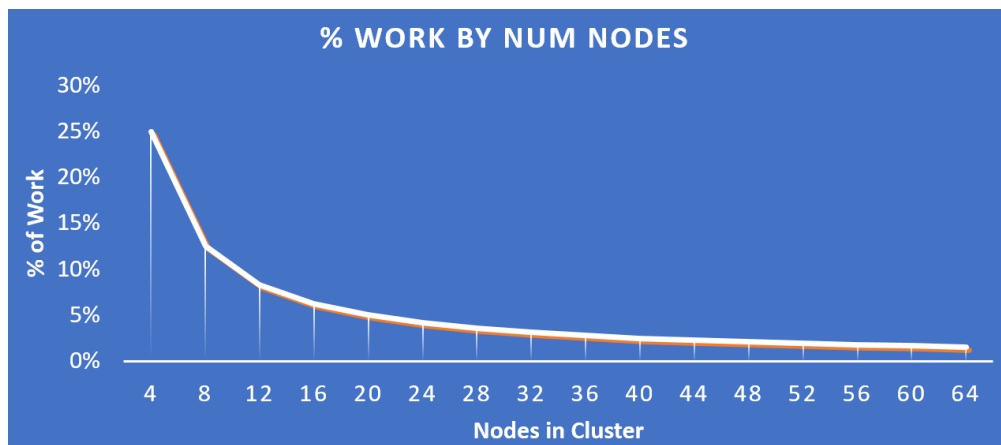
This enables our MapReduce Framework (Curator) to leverage the full power of the cluster to perform activities concurrently. Sample activities include that of data re-protection, compression, erasure coding, deduplication, etc.

The Nutanix cluster is designed to accommodate and remediate failure. The system will transparently handle and remediate the failure, continuing to operate as expected. The user will be alerted, but rather than being a critical time-sensitive item, any remediation (e.g. replace a failed node) can be done on the admin's schedule.

If you need to add additional resources to your Nutanix cluster, you can scale out linearly simply by adding new nodes. With traditional 3-tier architecture, simply adding additional servers will not scale out your storage performance. However, with a hyperconverged platform like Nutanix, when you scale out with new node(s) you're scaling out:

- The number of hypervisor / compute nodes
- The number of storage controllers
- The compute and storage performance / capacity
- The number of nodes participating in cluster wide operations

The following figure shows how the % of work handled by each node drastically decreases as the cluster scales:



Work Distribution - Cluster Scale

Key point: As the number of nodes in a cluster increases (cluster scaling), certain activities actually become more efficient as each node is handling only a fraction of the work.