[Hadoop](https://acadgild.com/big-data/big-data-development) – the solution for deciphering the avalanche of [Big Data](https://acadgild.com/big-data/big-data-development) – has come a long way from the time Google published its paper on Google File System in 2003 and MapReduce in 2004. It created waves with its scale-out and not a scale-up strategy. Inroads from Doug Cutting and the team at Yahoo and Apache Hadoop project resulted in popularizing MapReduce programming – which is intensive in I/O and is constrained in interactive analysis and graphics support. This paved the way for further evolving of Hadoop1 to Hadoop2. The following table describes the major differences between them:

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Hadoop1** | **Hadoop2** |
| **1** | Supports MapReduce (MR) processing model only. Does not support non-MR tools | Allows to work in MR as well as other distributed computing models like Spark, Hama, Giraph, Message Passing Interface) MPI & HBase coprocessors. |
| **2** | MR does both processing and cluster-resource management. | YARN (Yet Another Resource Negotiator) does cluster resource management and processing is done using different processing models. |
| **3** | Has limited scaling of nodes. Limited to 4000 nodes per cluster | Has better scalability. Scalable up to 10000 nodes per cluster |
| **4** | Works on concepts of slots – slots can run either a Map task or a Reduce task only. | Works on concepts of containers. Using containers can run generic tasks. |
| **5** | A single Namenode to manage the entire namespace. | Multiple Namenode servers manage multiple namespaces. |
| **6** | Has Single-Point-of-Failure (SPOF) – because of single Namenode- and in the case of Namenode failure, needs manual intervention to overcome. | Has to feature to overcome SPOF with a standby Namenode and in the case of Namenode failure, it is configured for automatic recovery. |
| **7** | MR API is compatible with Hadoop1x. A program written in Hadoop1 executes in Hadoop1x without any additional files. | MR API requires additional files for a program written in Hadoop1x to execute in Hadoop2x. |
| **8** | Has a limitation to serve as a platform for event processing, streaming and real-time operations. | Can serve as a platform for a wide variety of data analytics-possible to run event processing, streaming and real-time operations. |
| **9** | A Namenode failure affects the stack. | The Hadoop stack – Hive, Pig, HBase etc. are all equipped to handle Namenode failure. |
| **10** | Does not support Microsoft Windows | Added support for Microsoft windows |

**Detail Description:**

Now, let us see the above details on how Hadoop1 and Hadoop2 are different in brief.

**Scalability**

In Hadoop2.x with the help of YARN  architecture, we can run larger clusters than Hadoop v1. Hadoop v1 hits scalability bottlenecks in the region of 4,000 nodes and 40,000 tasks, deriving from the fact that the job tracker has to manage both jobs and tasks. YARN overcomes these limitations by virtue of its split resource manager/application master architecture: It is designed to scale up to 10,000 nodes and 100,000 tasks.

In contrast to the jobtracker, each instance of an application  – here, a MapReduce job – has a dedicated application master, which runs for the duration of the application. This model is actually closer to the original GFS paper, which describes how a master process is started to coordinate map and reduce tasks running on a set of workers.

**Ability to run non-MapReduce – jobs**

In Hadoop1.x, we can only run MapReduce framework jobs to process the data which is stored in HDFS. We couldn’t had the opportunity to run other applications than MapReduce in the HDFS cluster. Thus, Hadoop2.x came up with new framework YARN which provides the ability to run non-MapReduce jobs like Spark, Hama, Giraph, Message Passing Interface) MPI & HBase coprocessors.

**Namenode High Availability**

Previously, in Hadoop1.x we had single namenode which maintained a directory tree of HDFS files and tracked where data was stored in the cluster.  If the Namenode is down due to some unplanned event such as a machine crash, the whole Hadoop cluster will be down as well.

Hadoop2.x comes with the solution for this problem, which allows users to configure clusters with redundant namenodes, removing the chance that a lone namenode will become a single point of failure within a cluster.

**Native Windows Support**

Hadoop was originally developed to support the UNIX family of operating systems. With Hadoop2, the Windows operating system is natively supported. This extends the reach of Hadoop significantly to a sizable Windows Server market.

**Beyond Batch Oriented application**

Hadoop goes beyond Batch oriented nature in its version 2.0 and now can run interactive, streaming application also.

**Utilization**

In MapReduce v1, each tasktracker is configured with a static allocation of fixed-size “slots”, which are divided into map slots and reduce slots at configuration time. A map slot can only be used to run a map task, and a reduce slot can only be used for a reduce task. In YARN, a nod manager manages a pool of resources, rather than a fixed number of designated slots.

MapReduce running on YARN will not hit the situation where a reduce task has to wait because only map slots are available in the cluster, which can happen in MapReduce v1. If the resources to run the task are available, then the application will be eligible for them. Furthermore, resources in YARN are fine grained, so an application can make a request for what it needs, rather than for an indivisible slot, which may be too big (which is wasteful of resources) or too small (which may cause a failure) for the particular task. Multitenancy in some ways, the biggest benefit of YARN is that it opens up Hadoop to other types of distributed application beyond MapReduce

MapReduce is just one YARN application among many. It is even possible for users to run different versions of MapReduce on the same YARN cluster, which makes the process of upgrading MapReduce more manageable.

So this is the main differences between Hadoop1 and Hadoop architecture.