# **Yarn Experiments for Rack-Awareness**

We run the experiments with a 7 node cluster with 1x replication(1 master, 6 data nodes/node managers) and 1 block file.

**Hardware configuration**

Master (hostname: simple37)：

CPU: 2 x Intel(R) Xeon(R) E5-2620 v2 @ 2.10GHz /15M Cache 6-Core 12-Thread

Memory: 32GB (4x8GB) 1600MHz

Disk: 2TBx2 3.5-inch with RAID-1

Network bandwidth: 968Mb/s

Slaves（hostname: simple27~simple33）：

CPU: 2 x Intel(R) Xeon(R) E5-2620 v2 @ 2.10GHz /15M Cache 6-Core 12-Thread

Memory: 32GB (4x8GB) 1600MHz

Disk: 2TBx2 3.5-inch with RAID-1

Network bandwidth: 968Mb/s

**Software configuration**

Spark-1.6.2 Hadoop-2.7.1

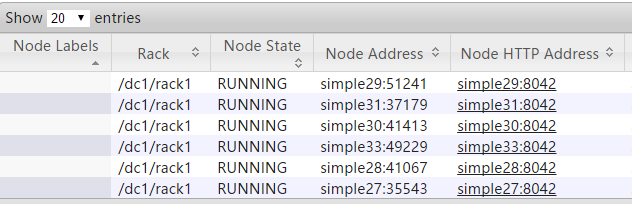
**Experiment 1**

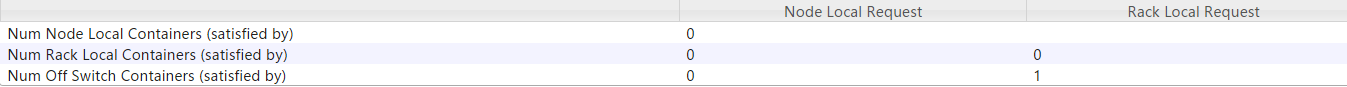
* 7node Hadoop cluster (1 master, 6 data nodes/node managers)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HostName** | Simple37 | Simple27 | Simple28 | Simple30 | Simple31 | Simple32 | Simple33 |
| **Role** | Master | Node1 | Node2 | Node3 | Node4 | Node5 | node6 |

* Configure HDFS with replication factor 1
* File has a single block in HDFS
* Configure the rack awareness for all the nodes to be in the same rack
* Configure Spark to use dynamic allocation
* Configure Yarn for both mapreduce shuffle service and Spark shuffle service
* Add a single small file (few bytes) to HDFS
* Run wordcount on the file (using Spark/MapReduce)

The rack awareness configuration and the results are shown in the WebUI in the following way:

****

****

**Results of experiment 1 (run 10 times):**

7 node cluster(1 master, 6 data nodes/node managers), 1x replication, 1 block file

All the nodes are in the same rack

MapReduce wordcount

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Round NO. | Data location | Scheduled node | Locality Level | Time Cost |
| 1 | Node2 | Node2 | NodeLocal | 17s |
| 2 | Node1 | Node2 | RackLocal | 16s |
| 3 | Node4 | Node5 | RackLocal | 18s |
| 4 | Node5 | Node3 | RackLocal | 14s |
| 5 | Node4 | Node3 | RackLocal | 17s |
| 6 | Node4 | Node2 | RackLocal | 14s |
| 7 | Node4 | Node1 | RackLocal | 15s |
| 8 | Node2 | Node3 | RackLocal | 16s |
| 9 | Node1 | Node4 | RackLocal | 19s |
| 10 | Node4 | Node3 | RackLocal | 17s |

**Data Locality Hit Rate : 10%**

Spark wordcount

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Round NO. | Data location | Scheduled node | Locality Level | Time Cost |
| 1 | Node2 | Node6 | RackLocal | 23s |
| 2 | Node1 | Node2 | RackLocal | 24s |
| 3 | Node4 | Node4 | NodeLocal | 24s |
| 4 | Node5 | Node3 | RackLocal | 24s |
| 5 | Node4 | Node6 | RackLocal | 23s |
| 6 | Node4 | Node6 | RackLocal | 23s |
| 7 | Node4 | Node4 | NodeLocal | 24s |
| 8 | Node2 | Node6 | RackLocal | 24s |
| 9 | Node1 | Node | RackLocal | 24s |
| 10 | Node4 | Node3 | RackLocal | 23s |

**Data Locality Hit Rate : 20%**

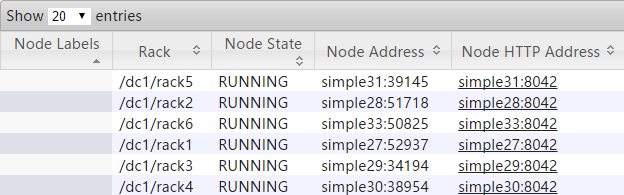
**Experiment 2:**

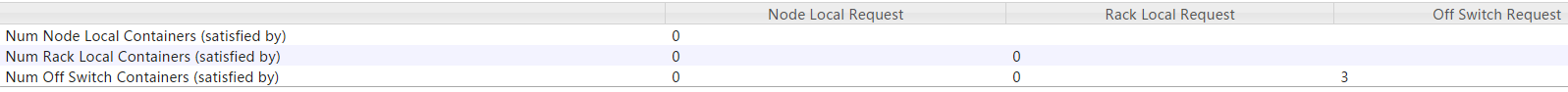
* 7node Hadoop cluster (1 master, 6 data nodes/node managers)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| HostName | Simple37 | Simple27 | Simple28 | Simple30 | Simple31 | Simple32 | Simple33 |
| Role | Master | Node1 | Node2 | Node3 | Node4 | Node5 | node6 |

* Configure HDFS with replication factor 1
* File has a single block in HDFS
* Configure the rack awareness for all the nodes to be in separate racks
* Configure Spark to use dynamic allocation
* Configure Yarn for both mapreduce shuffle service and Spark shuffle service
* Add a single small file (few bytes) to HDFS
* Run wordcount on the file (using Spark/MapReduce)

The rack awareness configuration and the results are shown in the WebUI in the following way:





**Results of experiment2 (run 10 times):**

7 node cluster(1 master, 6 data nodes/node managers), 1x replication, 1 block file

All the nodes are in the separate racks

MapReduce wordcount

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Round NO. | Data location | Scheduled node | Locality Level | Time Cost |
| 1 | Node2 | Node3 | OffSwitch | 17s |
| 2 | Node1 | Node2 | OffSwitch | 15s |
| 3 | Node4 | Node2 | OffSwitch | 15s |
| 4 | Node5 | Node4 | OffSwitch | 14s |
| 5 | Node4 | Node6 | OffSwitch | 14s |
| 6 | Node4 | Node3 | OffSwitch | 14s |
| 7 | Node4 | Node4 | NodeLocal | 13s |
| 8 | Node2 | Node2 | NodeLocal | 15s |
| 9 | Node1 | Node3 | OffSwitch | 15s |
| 10 | Node4 | Node3 | OffSwitch | 17s |

**Data Locality Hit Rate : 20%**

Spark wordcount

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Round NO. | Data location | Scheduled node | Locality Level | Time Cost |
| 1 | Node2 | Node2 | NodeLocal | 17s |
| 2 | Node1 | Node2 | OffSwitch | 20s |
| 3 | Node4 | Node5 | OffSwitch | 21s |
| 4 | Node5 | Node3 | OffSwitch | 18s |
| 5 | Node4 | Node3 | OffSwitch | 24s |
| 6 | Node4 | Node3 | OffSwitch | 19s |
| 7 | Node4 | Node4 | NodeLocal | 18s |
| 8 | Node2 | Node3 | OffSwitch | 16s |
| 9 | Node1 | Node2 | OffSwitch | 17s |
| 10 | Node4 | Node3 | OffSwitch | 17s |

**Data Locality Hit Rate : 20%**

**Experimental Results in Summary.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Experiment NO. | Number of Datanodes | Number of Replications | Number of Blocks | Locality Level | MapReduce  Data Locality Hit Rate | Spark  Data Locality Hit Rate |
| 1 | 6 | 1 | 1 | NodeLocality  RackLocality | 10% | 20% |
| 2 | 6 | 1 | 1 | NodeLocality  OffSwitch | 20% | 20% |

**Conclusion:**

From the results, we can find that the tasks get NodeLocality and RackLocality in the experiment 1 in which we configure the rack awareness for all the nodes to be in the same rack. The tasks get NodeLocality and OffSwitch in the experiment 2 in which we configure the rack awareness for all the nodes to be in separate racks. Thus, the rack awareness will not have an influence on DataLocality in former experiments.