NUMA Aware YARN Containers

1. Purpose

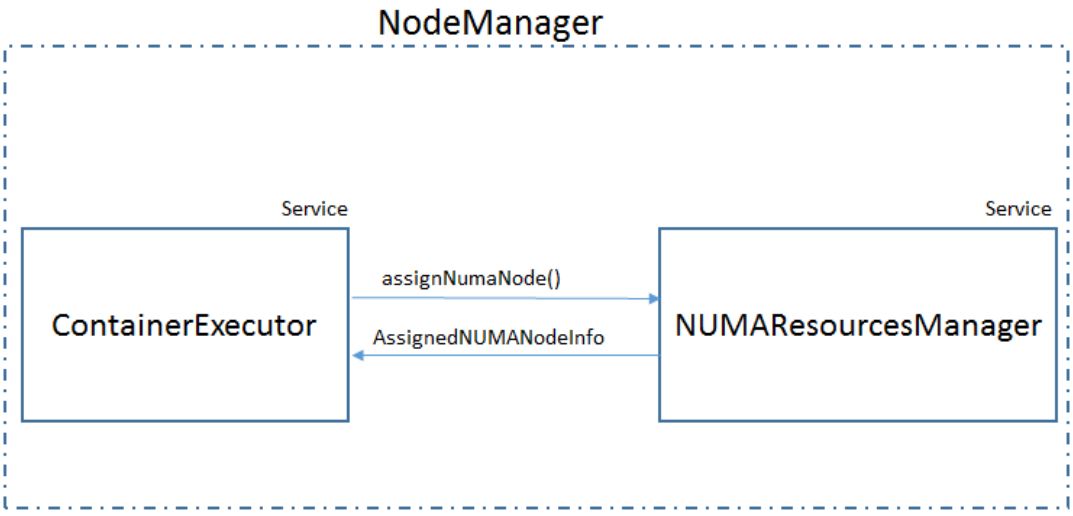
The purpose of this document is to provide design and implementation details for the Yarn containers to be aware of the Non-uniform memory access (NUMA) and bound to the allocated node/s.

2. Introduction

Non-uniform memory access (NUMA) is a computer memory design used in multiprocessing, where the memory access time depends on the memory location relative to the processor. Under NUMA, a processor can access its own local memory faster than non-local memory (memory local to another processor or memory shared between processors).

Yarn Containers can make benefit of this NUMA design to get better performance by binding to a specific NUMA node and all subsequent memory allocations will be served by the same node, reducing remote memory accesses.

3. Design Proposal



a. NUMAResourcesManager service class

NUMAResourcesManager is a service class added in Node Manager to handle the NUMA nodes assigning/scheduling. It will be started when the NUMA awareness is enabled in Node Manager. As part of its initialization, it reads the NUMA topology (nodes and their capabilities about memory and cpus) from system or from the configurations based on the property. It maintains a data structure about the numa nodes capability and current usage of those nodes resources. Whenever container executor requests numa node info using the API NUMAResourcesManager. assignNumaNode(), NUMAResourcesManager assigns the nodes for memory and CPUS from available resources in a Round Robin fashion. Some cases if there are no requested resources available in a single node then NUMAResourcesManager may assign multiple nodes. Once the containers complete the execution, their assigned resources will be released using the NUMAResourcesManager. releaseNumaResources() API.

b. Container Executor getting the numa node info

ContainerExecutor gets the NUMA node info (for memory and cpu's) from NUMAResourcesManager using the assignNumaNode () API. And then it binds the container to those node/s for memory and cpu's while launching using the NUMA API (Ex: numactl --membind=<node> --cpunodebind=<node> in Unix).

4. Configurations

a. Enable/Disable the NUMA awareness (yarn.nodemanager.numa-awareness.enabled)

This property enables the NUMA awareness feature in the Node Manager for the containers. By default the value of this property is false which means it is disabled. If this property is true then only the below configurations will be applicable otherwise they will be ignored.

Ex:

```
<property>
     <name>yarn.nodemanager.numa-awareness.enabled</name>
     <value>true</value>
</property>
```

b. NUMA topology reading (yarn.nodemanager.numa-awareness.read-topology)

This property decides whether to read the NUMA topology from the system or from the configurations. If this property value is true then the topology will be read from the system using 'numactl --hardware' command in UNIX systems and similar way in windows. If this property is false then the topology will be read using the below configurations. Default value of this configuration is false which means NodeManager will read the NUMA topology from the below configurations.

Ex:

```
<property>
     <name>yarn.nodemanager.numa-awareness.read-topology</name>
     <value>false</value>
</property>
```

c. NUMA nodes id's (yarn.nodemanager.numa-awareness.node-ids)

This property is used to provide the NUMA node ids as comma separated values. It will be read only when the 'yarn.nodemanager.numa-awareness.read-topology' is false.

Ex:

```
<property>
     <name>yarn.nodemanager.numa-awareness.node-ids</name>
     <value>0,1,2</value>
</property>
```

a. NUMA Node memory (yarn.nodemanager.numa-awareness.<NODE_ID>.memory)

This property will be used to read the memory(in MB) configured for each NUMA node specified in 'yarn.nodemanager.numa-awareness.node-ids' by substituting the node id in the place of <NODE_ID>. It will be read only when the 'yarn.nodemanager.numa-awareness.read-topology' is false.

Ex:

```
<property>
     <name>yarn.nodemanager.numa-awareness.<NODE_ID>.memory</name>
     <value>8192</value>
</property>
```

b. NUMA Node CPUs (yarn.nodemanager.numa-awareness.<NODE_ID>.cpus)

This property will be used to read the number of CPUs configured for each node specified in 'yarn.nodemanager.numa-awareness.node-ids' by substituting the node id in the place of <NODE_ID>. It will be read only when the 'yarn.nodemanager.numa-awareness.read-topology' is false.

Ex:

```
<property>
     <name> yarn.nodemanager.numa-awareness.0.cpus </name>
     <value>8</value>
</property>
```

5. Prerequisites

In addition to the NUMA support availability in each node, these tools/API's need to be available.

a. 'numactl' in non-windows systems
b. NUMA API's for reading and binding the numa node to a process

6. References

https://en.wikipedia.org/wiki/Non-uniform_memory_access