Dynamic Scaling Algorithm

Threshold Monitor Daemon should start along with Nimbus and run every few seconds (depending on the number of topologies)

**Enum: LastAction**
Action – AddTask, AddThread, AddNode
CurrentBolt – string name

**Class : Threshold**
DesiredThreshold
CurrentThreshold
LastAction

/* Topologies that have auto scaling enabled
* Populated every time a topology is submitted with a auto-scale threshold value
* Removed if; topology is killed, topology is restarted, cannot scale further
*/

Map<Topology, Threshold>

// Pre-defined storage of bolts based on the processing order
List<Bolts>

// Stores bolt names sorted by max average increase in threshold
Sorted-DataStructure <Max-AvgIncrease-Bolt>

**Method: IterateMap**
For Each key,value in Map
   invoke AutoScale(key, value) as new thread

**Method: AutoScale (Topology, Threshold)**
Get DesiredThreshold value from Threshold
Get NewThreshold from Metrics

IF NewThreshold >= Desired Treshold
   Do nothing
   // We dont remove the topology from map because there could be an input burst which can bring
   // down the threshold value at any point. Even after meeting the threshold condition, the daemon
   // should still check the topology in every cycle
ELSE
   Get last action from threshold-object
   Get current bolt from Threshold-Object
   Get current threshold from Threshold-object
Default Case: values are not available // First call for the topology
next-bolt = Topology->getNextBolt()
Invoke CLI script (increase num tasks by one for current bolt)
LastAction.value = AddTask
LastAction.currentbolt = next-bolt
LastAction.currentthreshold = new threshold

Case: value is AddTask
IF new threshold <= current threshold
  // Try to invoke both at the same time
  Invoke CLI script (decrease num tasks by one for current bolt)
  Invoke CLI script (increase num threads by one for current bolt with same number of tasks)
  LastAction.value = AddThread
ELSE
  Invoke CLI script (increase num tasks by one for current bolt)
  Calculate percentage increase
  Store current bolt in a sorted data structure by avg percentage increase
  LastAction.currentthreshold = new threshold

Case: value is AddThread

IF new threshold > current threshold
  Invoke CLI script (increase num threads by one for current bolt)
  Calculate percentage increase
  Store current bolt in a sorted data structure by percentage increase
  LastAction.currentthreshold = new threshold
ELSE
  next-bolt = Topology->nextBolt()
  IF next-bolt IS NOT NULL
    Invoke CLI script (increase num tasks by one for next bolt)
    Invoke CLI script (decrease num threads by one for current bolt)
    LastAction.value = AddTask
    LastAction.currentbolt = next-bolt
  ELSE
    // This case might be out of scope for the project. We are not sure if
    // Nimbus can be prompted to dynamically add a bolt to the topology.
    MaxAvgIncrease-DS.nextBolt()
    Prompt Nimbus to allocate a new parallel node for that bolt with the same
degree of parallelism/inputs and outputs
    LastAction.value = AddNode
    LastAction.currentbolt = maxavgincrease-bolt
    rebalance topology if required (field groupings)

Case: value is AddNode

IF new threshold > current threshold
  Get Last Action.current bolt
  Prompt Nimbus to allocate a new parallel node for that bolt with the same
degree of parallelism/inputs and outputs
Last action.currentthreshold = new threshold
rebalance topology if required (field groupings)
calculate percentage increase in threshold
Add average for all cloned nodes and original
ELSE
  maxavgincrease-bolt = MaxAvgIncrease-DS.nextBolt()
  IF maxavgincrease-bolt NOT NULL
    Prompt Nimbus to allocate a new parallel node for that bolt with the same
degree of parallelism/inputs and outputs
    LastAction.currentbolt = maxavgincrease-bolt
    rebalance topology if required (field groupings)
  ELSE
    We cant auto-scale anymore
    // This might be a good time to remove the topology from the map
ELSE
  We cant auto-scale anymore