LIMITLESS – Generic Tree-Based Overlay Network

User Manual

Alberto Cascajo García

Universidad Carlos III de Madrid

Grupo de Investigación ARCOS, Dep. de Informática

Contact: acascajo@inf.uc3m.es

LIMITLESS

LIMITLESS is an HPC framework that provides new strategies for resource management and job scheduling, based on executing different applications in shared compute nodes, maximizing platform utilization.

LIMITLESS is a scalable light-weight monitor that analyzes the compute nodes and detects interference-related hazards on the executing applications. This monitor is designed to provide scalability in two different ways. First, the monitor logic is distributed in a topological-aware configuration. The different monitor components are interconnected using a graph-based scheme that can be mapped to the cluster's network topology. The second scalability mechanism is aimed to reduce the memory and network monitoring overhead for large-scale platforms. The LDMs process the monitored data applying in-node filtering that reduces the network traffic based on the predefined tolerance. Additionally, the monitor includes a heartbeat protocol to detect faulting nodes.

The information retransmission between the components (from the nodes to the server) follows a Tree-Based Overlay Network (TBON) scheme. It allows the data packets to be aggregated in the intermediate nodes between the nodes and the server to reduce the number of connections to the server. Besides, in order to maintain the spatiality of the data, the intermediate nodes also retransmit the information in a concatenated packet that includes the IP address of the node that generates the data.

This manual will describe how to use the generic TBON to retransmit the data between the nodes to the servers.

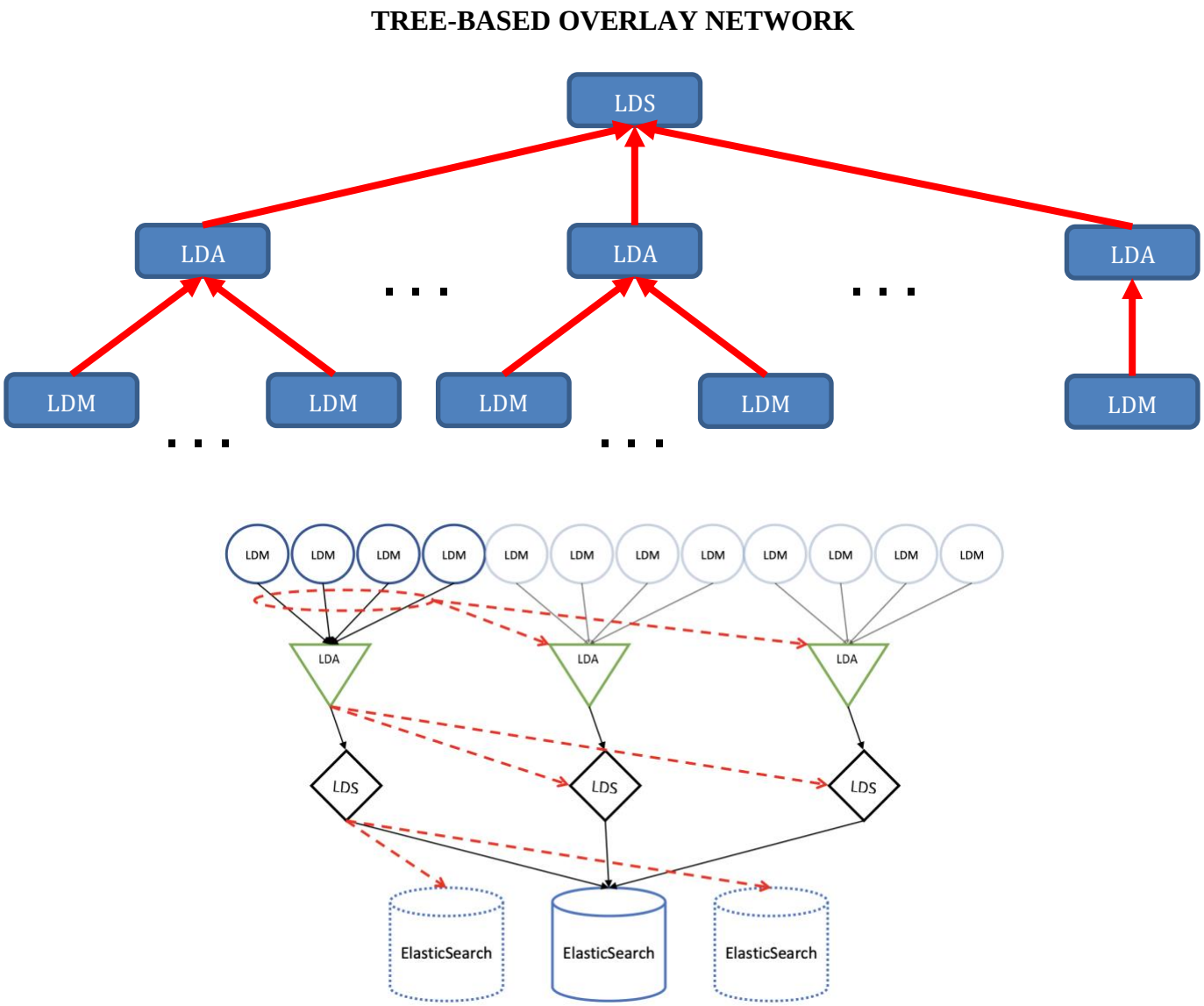

Figure 1 - Example of deployment with TMR and watchdog processes in the first monitoring branch

The System Monitor has been designed following the Tree-Based Overlays Network (TBON) architecture. It offers simplicity and scalability and allows the implementation of fault tolerance mechanisms. The monitor consists of one LIMITLESS Daemon Monitor (LDM) per node, which periodically collects the performance metrics; a set of LIMITLESS DaeMon Aggregators (LDAs), that forwards the information from the LDMs to other aggregators or servers; and the LIMITLESS DaeMon Server (LDS) that gathers and stores the monitoring information in ElasticSearch. Figure 1 shows a typical deployment of LIMITLESS with replication of the LDAs and LDSs. The purpose of replicating these components is to enhance the monitor scalability and resilience.

LIMITLESS components:

- **LIMITLESS Daemon Monitor** (LDM): Collects and sends the performance counters in the nodes.
- **LIMITLESS Daemon Aggregator** (LDA): Aggregates and retransmits the received performance counters from the LDMs.
- **LIMITLESS Daemon Server** (LDS): Unpacks the monitoring packets and processes them.

Due to this deployment structure, the monitor can scale and process the data as the user needs. There are two ways to deploy the monitoring tool:

- Small/Medium cluster (up to 200 nodes): One LDS can manage 200 LDMs.
- Large-scale clusters: In this case, each LDM should send the collected metrics to a certain LDA (the topology is designed by the user). The LDA aggregates the received metrics from the LDMs and retransmit the result to the LDS or another LDA (if more than one aggregation layer is required).

Generic TBON description

If developers want to leverage the TBON features (which do not include the monitoring daemons), they can do it including a piece of code in their own processes. The target information should be sent following the next requirements:

- Applications should send the data using sockets to LDAs with the following format:
    - Key;Value;Key;Value…
- *Key* represents the name of the field that will store the value. It is a string with a maximum size of 64 bytes.
- *Value* represents the value of the counter. It is an unsigned long value (8 bytes).

COMPILATION

Known dependencies and requirements:

- Need of a Linux based distro, 64 bits.
- Prometheus database (data exporter and exposer). It will be optional in future versions.
- For compiling: cmake 2.8 or later

Compilation

- LDM and example (in *src*): *root_directory/*

```
mkdir build
cd build
cmake ..
make
```

- LDA and LDS (same binary): *root_directory/test_server/*

```
cd test_server
mkdir build && cd build
cmake ..
make
```

LIMITLESS arguments

The monitoring tool uses configuration files to pass the arguments to the different components. They include tags to identify the arguments and make them understandable. There are two configuration files: init.dae and init.serv. The first one should be in the same directory as the LDM binary and the second one in the same directory as the LDA/LDS binary. They are automatically processed and re-processed every sampling interval (it allows reconfigurations in run time).

Generic TBON arguments

This manual assumes that the application sends the information to the LDAs based on the specified requirements. In order to run the LDA and LDS components with the correct arguments to build the TBON, it is important to know their arguments:

- LDA:
    - -p : socket listener port.
    - -r : socket dest. port.

  - -s : socket dest. IP address.
- LDS:
  - -p : socket listener port.

In order to build and send the packets, the applications must call two main functions provided by the LIMITLESS code:

```
Packed_sample *ps = new Packed_sample()
Ps->Pack_sample_generic ( string key_val_string )
Send_monitor_generic( ps  )
```

Complete deployment in a cluster

- A startup and stopping script are provided in folder *conf*:

```
cd conf
./start.sh -m <server addr> -p <server port> -f <hostfile for daemons> -t <sampling interval> -q <mem threshold> -e <network threshold>

./stop.sh -m <address of master node> -f <text file with client addresses>
```

- This deployment is made through ssh connections. For its correct working it is necessary that nodes accept input connections and that keys are deployed properly to avoid the password requirements.

The start script executes the LIMITLESS daemon (LDM) on each node identified in the hostfile. It also executes the LIMITLESS server (LDS) in the node identified by the "m" flag argument. In order to stop the monitoring components, the stop script needs the server address and the hostfile with the nodes that execute the LDM processes. Note that the script arguments are the same for each LDM.

If the users want a topological deployment using LDMs, LDAs and LDSs, please, contact us.

Generic TBON deployment

- A startup and stopping script are provided in folder *conf*:

```
cd conf
./start_tbon.sh -m <lds_addr> -r <lds_port> -p <port_listener> -f <lda_hostfile>

./stop.sh -m <lds_address> -f <lda_hostfile>
```

- This deployment is made through ssh connections. For its correct working it is necessary that nodes accept input connections and that keys are deployed properly to avoid the password requirements.
- Users applications must include the specific function calls to send the information to the LDAs.

Note that this deployment executes LDAs → LDS communication pattern. If a different topological deployment is needed, please contact us. However, it could be done by executing this pattern with different IP addresses.