

RDMA for Apache Kafka 0.9.1 User Guide

HIGH-PERFORMANCE BIG DATA TEAM

<http://hibd.cse.ohio-state.edu>

NETWORK-BASED COMPUTING LABORATORY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THE OHIO STATE UNIVERSITY

Copyright (c) 2011-2018
Network-Based Computing Laboratory,
headed by Dr. D. K. Panda.
All rights reserved.

Last revised: 10/25/2018

Contents

- 1 Overview of the RDMA for Apache Kafka Project** **1**
- 2 Features** **1**
- 3 Installation Instructions** **1**
 - 3.1 Prerequisites 1
 - 3.2 Download 1
 - 3.3 Installation 2
- 4 Configuration** **2**
 - 4.1 Basic Configuration 2
 - 4.2 Advanced Configuration 3
- 5 Basic Usage Instructions** **4**
 - 5.1 Zookeeper 4
 - 5.2 Kafka Broker 4
 - 5.3 Shutdown 4
- 6 RDMA-Kafka Benchmarks** **4**
 - 6.1 Microbenchmarks 4
- 7 Troubleshooting with RDMA-Kafka** **5**

1 Overview of the RDMA for Apache Kafka Project

RDMA for Apache Kafka is a high-performance design of Apache Kafka over RDMA-enabled Interconnects. This version of RDMA for Apache Kafka is based on Apache Kafka 1.0.0. This file is intended to guide users through the various steps involved in installing, configuring, and running RDMA for Apache Kafka over InfiniBand. If there are any questions, comments or feedbacks regarding this software package, please post them to rdma-hadoop-discuss mailing list (rdma-hadoop-discuss@cse.ohio-state.edu).

2 Features

High-level features of RDMA for Apache Kafka 0.9.1 are listed below.

- Based on Apache Kafka 1.0.0
- High performance design with native InfiniBand support at the verbs level for Apache Kafka
- Compliant with Apache Kafka 1.0.0 APIs and applications
- Easily configurable for native InfiniBand, RoCE, and the traditional sockets based support (Ethernet and InfiniBand with IPoIB)
- Support for RDMA Device Selection
- On-demand connection setup
- Tested with
 - Native Verbs-level support with Mellanox InfiniBand adapters (DDR, QDR, FDR, and EDR)
 - Various multi-core platforms

3 Installation Instructions

3.1 Prerequisites

In order to use the RDMA-based features provided with RDMA for Apache Kafka, install the latest version of the OFED distribution that can be obtained from <http://www.openfabrics.org>.

3.2 Download

The latest version of RDMA for Apache Kafka package can be downloaded from <http://hibd.cse.ohio-state.edu/download/hibd/rdma-kafka-2.11-0.9.1.tgz>.

3.3 Installation

The following steps can be used to install the RDMA for Apache Kafka package.

1. Extract the RDMA for Apache Kafka distribution tarball using the following command:

```
tar -xzf rmda-kafka_2.11-0.9.1.tgz
```

2. Change directory to rdma-kafka-0.9.1

```
cd rdma-kafka-0.9.1
```

Please email us at rdma-hadoop-discuss@cse.ohio-state.edu if you experience any trouble installing the package on your system.

4 Configuration

4.1 Basic Configuration

Steps to configure RDMA for Apache Kafka include:

1. Add the path to JNI layer shared libraries to the `LD_LIBRARY_PATH` environment variable. For e.g. if the `rdma-kafka` folder is contained in the home directory of the user then the following command could be used to set the environment variable.

```
# Assuming the rdma-kafka-0.9.1 is located in the home directory
export LD_LIBRARY_PATH="$~/rdma-kafka-0.9.1/native:$LD_LIBRARY_PATH"
```

The `LD_LIBRARY_PATH` should be correctly configured on all machines where Kafka brokers or any of the Kafka clients i.e. producers or consumers are launched.

2. Configure `server.properties`, `producer.proprties` and `consumer.properties` to select the network protocol. Configuration of IB mode:

```
# Used to enable/disable RDMA over InfiniBand.
# Default value is true.
ib.enabled=true

# Used to enable/disable ROCE over InfiniBand.
# Default value is false.
roce.enabled=false
```

Configuration of the RoCE mode:

```
# Used to enable/disable RDMA over InfiniBand.
# Default value is true.
ib.enabled=false

# Used to enable/disable ROCE over InfiniBand.
# Default value is false.
roce.enabled=true
```

Configuration of the TCP/IP Mode:

```
# Used to enable/disable RDMA over InfiniBand.
# Default value is true.
ib.enabled=false

# Used to enable/disable ROCE over InfiniBand.
# Default value is false.
roce.enabled=false

# In server.properties
security.inter.broker.protocol=PLAINTEXT

# In producer.properties/consumer.properties
security.protocol=PLAINTEXT
```

We can also configure more specific items according to actual needs. For example, we can configure the item `zookeeper.connect` in `server.properties` to specify the hostname and port number of the servers where zookeeper instances are located. To get more information on this parameters, please visit <https://kafka.apache.org/10/documentation.html>.

4.2 Advanced Configuration

Users can specify which RDMA device Kafka should utilize in the case where more than one of such devices is installed. Users can select the RDMA device to be used either by ID or by name in the `server.properties`, `producer.properties` and `consumer.properties` files, depending on which module is being configured. By default, RDMA for Apache Kafka will use the device with ID 0 if none of the following parameters are defined. These parameters are applicable for all modes.

```
#Specify the name of the RDMA device to be used.
#This takes precedence over rdma.dev.num if both are set.
rdma.dev.name=mlx5_0

#Specify the ID of the RDMA device to be used.
rdma.dev.num=0
```

5 Basic Usage Instructions

RDMA for Apache Kafka 0.9.1 has management options similar to default Apache Kafka 1.0.0. This sections lists the basic usage instructions.

5.1 Zookeeper

Before starting the Kafka broker, make sure to start the Zookeeper instance. Use the following command to start the Zookeeper instance:

```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

5.2 Kafka Broker

The Kafka Broker can be started using the following command:

```
bin/kafka-server-start.sh config/server.properties
```

At this point, your Kafka cluster should be up and running.

5.3 Shutdown

Stop the cluster by first shutting down the Kafka broker and then the Zookeeper instance:

```
bin/kafka-server-stop.sh  
bin/zookeeper-server-stop.sh
```

6 RDMA-Kafka Benchmarks

6.1 Microbenchmarks

Kafka provides microbenchmarks for its Producer and Consumer API included with other packages which can be used to benchmark RDMA-Kafka package as well.

A brief description of these benchmarks is provided below:

Producer Benchmark:

This micro-benchmark measures latency and throughput of messages sent to Kafka broker(s) by one or more producers. The following command can be used to run the benchmark to produce a total of 500000 messages for topic “test” with record size of 100B and throughput throttling disabled.

```
bin/kafka-producer-perf-test.sh --topic test --num-records 500000  
--throughput -1 --record-size 100
```

Consumer Benchmark:

This micro-benchmark measures throughput of a Consumer client for the messages that it reads from Kafka broker(s). The following command can be used to run the benchmark to consume a total of 500000 messages for topic “test” from a Kafka broker instance running on localhost port 9092.

```
bin/kafka-consumer-perf-test.sh --topic test --messages 500000  
--broker-list localhost:9092
```

For more information on these benchmarks execute them with `—help` flag to get a list of all configurable parameters.

7 Troubleshooting with RDMA-Kafka

If you are experiencing any problems with the RDMA-Kafka package, please feel free to contact us by sending an email to rdma-hadoop-discuss@cse.ohio-state.edu.