**Table of Contents**

[**Project: OpenSource Ecosystem for POWER** 2](#_Toc438469542)

[**Task1: Porting Mongo-tools on RHEL ppc64le** 2](#_Toc438469543)

[**A.** **Building Go and setting up environment for bulding mongodb-tools on RHEL PPC** 2](#_Toc438469544)

[**B.** **Building mongoimport on RHEL PPC** 4](#_Toc438469545)

[**C.** **Building all mongodb tools on RHEL PPC** 4](#_Toc438469546)

[**D.** **Setting up Environment for Smoke Tests on RHEL PPC** 5](#_Toc438469547)

[**E.** **Common Test Run on RHEL PPC** 5](#_Toc438469548)

[**F.** **Mongoimport Test Run on RHEL PPC** 5](#_Toc438469549)

[**G.** **Other tools Test Run on RHEL PPC** 6](#_Toc438469550)

[**H.** **RESULT on RHEL PPC** 7](#_Toc438469551)

[**Task2: Porting Mongo-tools on Ubuntu ppc64le** 7](#_Toc438469552)

[**A.** **Building Go and setting up environment for building mongodb-tools on Ubuntu PPC** 7](#_Toc438469553)

[**B.** **Building mongoimport Ubuntu PPC (Same as Task1)** 8](#_Toc438469554)

[**C.** **Building all mongodb tools on Ubuntu PPC (Same as Task1)** 8](#_Toc438469555)

[**D.** **Setting up Environment for Smoke Tests** 8](#_Toc438469556)

[**E.** **Common Test Run on Ubuntu PPC (Same as Task1)** 8](#_Toc438469557)

[**F.** **Mongoimport Test Run Ubuntu PPC (Same as Task1)** 8](#_Toc438469558)

[**G.** **Other tools Test Run Ubuntu PPC (Same as Task1)** 8](#_Toc438469559)

[**H.** **RESULT on Ubuntu PPC** 8](#_Toc438469560)

# **Project: OpenSource Ecosystem for POWER**

# **Task1: Porting Mongo-tools on RHEL ppc64le**

## **Building Go and setting up environment for building mongodb-tools on RHEL PPC**

1. Prepare linux-ppc64le bootstrap **on Ubuntu x86 machine** and copy to the **RHEL** **Power PC.**
   1. Downloaded go 1.4 binaries using

wget <https://storage.googleapis.com/golang/go1.4.linux-amd64.tar.gz>

mkdir go1.4

tar –xvzf go1.4.linux-amd64.tar.gz –C go1.4

Set GOROOT\_BOOTSTRAP to the path of the go binaries.

export GOROOT\_BOOTSTRAP=/root/deepali/go1.4/go/

* 1. Download the Go v1.5.1 source code on the **Ubuntu** **x86** machine and extract it  
     wget <https://storage.googleapis.com/golang/go1.5.1.src.tar.gz>

tar -xvzf go1.5.1.src.tar.gz

Note: I have extracted the source code to the folder /root/deepali/prebuiltpackages/go1.5.1.

* 1. Prepare bootstrap with arch as ppc64le on **the Ubuntu x86 machine**.

cd go/src

GOOS=linux GOARCH=ppc64le ./bootstrap.bash

This cross-compiles a toolchain for that GOOS/GOARCH combination, leaving the resulting tree in ../../go-${GOOS}-${GOARCH}-bootstrap.

* 1. Copy the so generated bootstrap to the target **RHEL PPC** machine.

scp go-linux-ppc64le-bootstrap.tbz root@10.77.67.126:/root/deepali/gobootstrap\_using1.5

Note: The directory /root/deepali/gobootstrap\_using1.5 is present on power pc.

1. Extract the copied bootstrap file **on RHEL PPC**.

cd /root/deepali/gobootstrap\_using1.5

tar -xvjf go-linux-ppc64le-bootstrap.tbz

1. Download and extract Go v1.5.1 source code on **RHEL** PPC for locally building Go v1.5.1 using the bootstrap.

cd /root/deepali/gobootstrap\_using1.5

wget <https://storage.googleapis.com/golang/go1.5.1.src.tar.gz>

tar -xvzf go1.5.1.src.tar.gz

1. Setup GOROOT\_BOOTSTRAP as below on power pc.

export GOROOT\_BOOTSTRAP=/root/deepali/gobootstrap\_using1.5/go-linux-ppc64le-bootstrap

1. Build Go locally on **RHEL PPC**.

cd go/src

./make.bash

This generates go binaries. It displays the following at the end.

Installed Go for linux/ppc64le in /root/deepali/gobootstrap\_using1.5/go

Installed commands in /root/deepali/gobootstrap\_using1.5/go/bin

1. Install the dependencies with the following commands on **RHEL** PPC
   1. Install sasl libs  
              sudo yum install cyrus-sasl  
              sudo yum install cyrus-sasl-devel

   b. Install ssl libs  
         sudo yum intall openssl  
        sudo yum install -y openssl-devel

1. Download mongodb-tools from git on **RHEL PPC. (**This example uses code from branch r3.2.0.)

cd /root/deepali/packages/

git clone https://github.com/mongodb/mongo-tools.git

cd mongo-tools

git checkout r3.2.0

1. Modify /root/deepali/packages/mongo-tools/set\_gopath.sh to file and execute it (Add –x to the first line so that it looks like **#!/bin/bash -x**.

This will show the value that should be set for GOPATH.

./set\_gopath.sh

+ TOOLS\_PKG=github.com/mongodb/mongo-tools

+ setgopath

+ '[' Windows\_NT '!=' '' ']'

++ pwd

+ SOURCE\_GOPATH=/root/deepali/packages/mongo-tools.gopath

++ pwd

+ VENDOR\_GOPATH=/root/deepali/packages/mongo-tools/vendor

+ rm -rf .gopath/

++ dirname github.com/mongodb/mongo-tools

+ mkdir -p .gopath/src/github.com/mongodb

++ pwd

+ ln -sf /root/deepali/packages/mongo-tools .gopath/src/github.com/mongodb/mongo-tools

++ pwd

++ pwd

+ export GOPATH=/root/deepali/packages/mongo-tools/.gopath:/root/deepali/packages/mongo-tools/vendor

+ GOPATH=/root/deepali/packages/mongo-tools/.gopath:/root/deepali/packages/mongo-tools/vendor

1. Setup GOPATH using the output from 8 above.

export GOPATH=/root/deepali/packages/mongo-tools/.gopath:/root/deepali/packages/mongo-tools/vendor

1. Setup path to Go binaries as:

export PATH=/root/deepali/gobootstrap\_using1.5/go/bin:$PATH

## **Building mongoimport on RHEL PPC**

1. Build mongoimport

mkdir bin

go build -o bin/mongoimport -tags "ssl sasl" mongoimport/main/mongoimport.go

This generates the mongoimport binary under bin folder.

## **Building all mongodb tools on RHEL PPC**

go build -o bin/mongoimport -tags "ssl sasl" mongoimport/main/mongoimport.go; go build -o bin/mongoexport -tags "ssl sasl" mongoexport/main/mongoexport.go; go build -o bin/mongostat -tags "ssl sasl" mongostat/main/mongostat.go; go build -o bin/mongodump -tags "ssl sasl" mongodump/main/mongodump.go; go build -o bin/mongorestore -tags "ssl sasl" mongorestore/main/mongorestore.go; go build -o bin/mongofiles -tags "ssl sasl" mongofiles/main/mongofiles.go; go build -o bin/mongotop -tags "ssl sasl" mongotop/main/mongotop.go; go build -o bin/mongooplog -tags "ssl sasl" mongooplog/main/mongooplog.go; go build -o bin/bsondump -tags "ssl sasl" bsondump/main/bsondump.go

## **Setting up Environment for Smoke Tests on RHEL PPC**

* 1. Install the dependencies

sudo yum install python-devel

  pip install pymongo

* 1. Copy mongo server files. (You need to have the latest stable version of the rebuilt tools, mongod, mongos, and mongo in your current working directory. <https://github.com/mongodb/mongo-tools/blob/master/CONTRIBUTING.md> )

cd /root/deepali/packages/mongo-tools/test/qa-tests

Copy the mongd, mongos and mongo to the current directory.

cp /usr/local/mongo/build/install/bin/mongod .

cp /usr/local/mongo/build/install/bin/mongos .

cp /usr/local/mongo/build/install/bin/mongo .

* 1. Set path to the mongodb-tools binaries as:

export PATH=$PATH:/root/deepali/packages/mongo-tools/bin

## **Common Test Run on RHEL PPC**

1. Execute unit and integration tests for common folder.
   1. For each folder under common dir execute i.e for bsonutil, db,intents, run the following command:

go test -v -test.types=unit,integration

* 1. For other folders –options, json, log, progress, util, archive and text, run the following command

go test -v

## **Mongoimport Test Run on RHEL PPC**

1. Execute unit tests for mongoimport

cd /root/deepali/packages/mongo-tools/mongoimport

go test -v -test.types=unit

**The Summary is:**

298 assertions thus far

--- PASS: TestTSVConvert (0.00s)

**PASS**

ok \_/root/deepali/packages/mongo-tools/mongoimport 0.207s

1. Execute integration tests for mongoimport

Integration tests require a mongod (running on port 33333). So mongod server should be installed and running on port 33333. (Install mongod if it is not already installed on the machine.) In my machine mongod is installed at /usr/local/mongo/build/install/bin/mongod.

* 1. Started mongod on another terminal

mongod --port 33333

* 1. Executed integration tests.

go test -v -test.types=integration

* 1. **The summary is:**

57 assertions thus far

--- PASS: TestImportDocuments (0.76s)

=== RUN TestTSVStreamDocument

--- SKIP: TestTSVStreamDocument (0.00s)

=== RUN TestTSVReadAndValidateHeader

--- SKIP: TestTSVReadAndValidateHeader (0.00s)

=== RUN TestTSVConvert

--- SKIP: TestTSVConvert (0.00s)

**PASS**

ok \_/root/deepali/packages/mongo-tools/mongoimport 0.765s

**Note**: The unit and integration tests can be executed in one go with the command  
 go test -v -test.types=unit,integration

1. Execute smoke tests for **mongoimport**
   1. Execute smoke tests

cd /root/deepali/packages/mongo-tools/test/qa-tests

python buildscripts/smoke.py import

## **Other tools Test Run on RHEL PPC**

1. Execute unit and integration tests for mongoexport
   1. Start mongod server on port 33333 on another terminal.

mongod --port 33333

* 1. Execute tests

cd /root/deepali/packages/mongo-tools/mongoexport

go test -v -test.types=unit,integration

1. Execute smoke tests for mongoexport

cd /root/deepali/packages/mongo-tools/test/qa-tests

python buildscripts/smoke.py export

1. Steps similar to G1 can be used to execute unit and integration tests for other tools namely mongodump, mongorestore, mongostat, mongofiles,mongooplog.
2. Steps similar to G2 can be used to execute smoke tests for all other tools.
3. To execute smoke tests for all the tools the following can be used:

cd /root/deepali/packages/mongo-tools/test/qa-tests

python buildscripts/smoke.py bson export files import oplog restore stat top

## **RESULT on RHEL PPC**

The integration tests are not passing for mongodump.

The smoke tests are not passing for mongorestore, mongostat, mongofiles and mongotop.

# **Task2: Porting Mongo-tools on Ubuntu ppc64le**

## **Building Go and setting up environment for building mongodb-tools on Ubuntu PPC**

All the steps are same as that of Task1, except the step at which dependencies are installed. So install the dependencies using the following commands on Ubuntu PPC and refer other steps from Task1.

* 1. Install pkg-config

sudo apt-get install pkg-config

* 1. Install sasl libs

sudo apt-get install libsasl2-dev

* 1. Install ssl libs

sudo apt-get install openssl libssl-dev

## **Building mongoimport Ubuntu PPC (Same as Task1)**

## **Building all mongodb tools on Ubuntu PPC (Same as Task1)**

## **Setting up Environment for Smoke Tests**

1. Install the dependencies

sudo apt-get install build-essential python-dev

sudo apt-get install python3-pip

sudo pip3 install pymongo

1. Copy mongo server files. (You need to have the latest stable version of the rebuilt tools, mongod, mongos, and mongo in your current working directory. <https://github.com/mongodb/mongo-tools/blob/master/CONTRIBUTING.md> )

cd /root/deepali/packages/mongo-tools/test/qa-tests

Copy the mongd, mongos and mongo to the current directory.

cp /usr/local/mongo/build/install/bin/mongod .

cp /usr/local/mongo/build/install/bin/mongos .

cp /usr/local/mongo/build/install/bin/mongo .

1. Set path to the mongodb-tools binaries as:

export PATH=$PATH:/root/deepali/packages/mongo-tools/bin

## **Common Test Run on Ubuntu PPC (Same as Task1)**

## **Mongoimport Test Run Ubuntu PPC (Same as Task1)**

## **Other tools Test Run Ubuntu PPC (Same as Task1)**

## **RESULT on Ubuntu PPC**

The integration tests are not passing for mongodump.

The smoke tests are not passing for mongorestore, mongostat, mongofiles and mongotop.