Atlas

Christian Külker

2023-01-26

Contents

1 Introduction

Compiling HPC components on a Raspberry Pi does not make sense because the Raspberry Pi is not powerful. However, the process and considerations for building HPC software components such as Atlas are very similar to those on Intel or AMD based HPC systems. For educational purposes, this document describes how to build Atlas 3 on Raspberry Pi 4. Be patient, compiling Atlas takes time and depends on single core performance.

2 Raspberry Pi Atlas 3.10.3

This will build and install Atlas 3.10.3 for Raspberry Pi 4. Except for setting a non-throttling mode, this is similar to other architectures.

Atlas 3.10.3 from 2016 is still the latest (2022-06-18) release.

For this compilation, mpich was installed first. Dependencies on mpich are not listed here.

For Atlas to be useful in HPC, a homogeneous cluster should be considered. Atlas should be compiled from source for each hardware architecture as Atlas performs timing calculations during build time. The build time is highly dependent on the performance of the individual cores. On a Raspberry Pi 4 8GB it can take 15 hours and 44 minutes, while on a modern AMD it can take 6 hours and 30 minutes.

Preparations as root

```
mkdir -p /opt/hpc/src
chown -R $USER.$USER /opt/hpc
apitude install cpufrequtils
```

Make sure you set performance and disable CPU throttling. Assuming certain hardware, you can get the number of cores via a command (or you have to find out via `/proc/cpuinfo`)

1. Either try `cpufreq` to disable throttling

```
numactl --hardware|grep cpus|sed -e 's%node 0 cpus:%%'
 0 1 2 3 4 5 6 7 8 9 10 11
for c in `numactl --hardware|grep cpus|sed -e 's%node 0 cpus:%%'`;do\
/usr/bin/cpufreq-set -g performance $c;done
```

2. Or set `performance` manually:

```
numactl --hardware|grep cpus|sed -e 's%node 0 cpus:%%'
 0 1 2 3 4 5 6 7 8 9 10 11

for c in `numactl --hardware|grep cpus|sed -e 's%node 0 cpus:%%'`;do\
echo performance|sudo
 ↪  /sys/devices/system/cpu/cpu$c/cpufreq/scaling_governor;\
done

for c in `numactl --hardware|grep cpus|sed -e 's%node 0 cpus:%%'`;do \
echo -n "CPU $c ";cat
 ↪  /sys/devices/system/cpu/cpu$c/cpufreq/scaling_governor;\
done
```

3. Or set by kernel parameter as described in ATLAS/doc/atlas_install.pdf page 5 (not tested).
4. Or use `BLAS` (since performance cannot be guaranteed anyway, throttling cannot be disabled).

5. Or if you insist on ATLAS, disable timing with `--cripple-atlas-performance`

If throttling is **not** disabled and you are not using `--cripple-atlas-performance`, you may see this error (copied from a non-Raspberry Pi):

```
ERROR: enum fam=0, chip=32765, model=113, mach=-1785083552
make[3]: *** [Makefile:106: atlas_run] Error 100
make[2]: *** [Makefile:449: IRunArchInfo_x86] Error 2
CPU Throttling apparently enabled!
```

Either check the list above, the Atlas PDF `doc/atlas_install.pdf` included in the archive, the more recent online documentation, use `BLAS` or compile with `--cripple-atlas-performance`.

When building Atlas, do not use the `-j` option, as this will mess up Atlas timings. The make run will take some time. Make sure the system is up that long and is not being used by other processes. It might make sense to run it in `screen` or `tmux`.

As user

```
export VER=3.10.3
export PFX=/opt/hpc/rpi/la/atlas/$VER
mkdir -p $PFX/{bld,arc}
cd /opt/hpc/src
wget https://sourceforge.net/projects/math-
 ↪  atlas/files/Stable/$VER/atlas$VER.tar.bz2
cd $PFX/arc
tar xvjf /opt/hpc/src/atlas$VER.tar.bz2 --strip-components=1
cd $PFX/bld
../arc/configure --prefix=$PFX
time make
...
make[2]: Leaving directory '/opt/hpc/rpi/la/atlas/3.10.3/bld/bin'
   DONE  STAGE 5-1-0 at 05:57

ATLAS install complete.  Examine
ATLAS/bin/<arch>/INSTALL_LOG/SUMMARY.LOG for details.
make[1]: Leaving directory '/opt/hpc/rpi/la/atlas/3.10.3/bld'
make clean
make[1]: Entering directory '/opt/hpc/rpi/la/atlas/3.10.3/bld'
rm -f *.o x* config?.out *core*
make[1]: Leaving directory '/opt/hpc/rpi/la/atlas/3.10.3/bld'
make check # perform sanity tests (optional)
```

```
make ptcheck # checks of threaded code (optional)
make time # provide performance summary (optional)
make install
```

After a full build, the following should be installed:

```
/opt/hpc/rpi/la/atlas/3.10.3/include/cblas.h
/opt/hpc/rpi/la/atlas/3.10.3/include/clapack.h
/opt/hpc/rpi/la/atlas/3.10.3/include/atlas/* # 161 files.
/opt/hpc/rpi/la/atlas/3.10.3/lib/libatlas.a
/opt/hpc/rpi/la/atlas/3.10.3/lib/libcblas.a
/opt/hpc/rpi/la/atlas/3.10.3/lib/liblapack.a
/opt/hpc/rpi/la/atlas/3.10.3/lib/libf77blas.a
/opt/hpc/rpi/la/atlas/3.10.3/lib/libptcblas.a
/opt/hpc/rpi/la/atlas/3.10.3/lib/libptf77blas.a
/opt/hpc/rpi/la/atlas/3.10.3/lib/libsatlas.dylib # sometimes not build
/opt/hpc/rpi/la/atlas/3.10.3/lib/libtatlas.dylib # sometimes not build
/opt/hpc/rpi/la/atlas/3.10.3/lib/libsatlas.dll # sometimes not build
/opt/hpc/rpi/la/atlas/3.10.3/lib/libtatlas.dll # sometimes not build
/opt/hpc/rpi/la/atlas/3.10.3/lib/libsatlas.so # sometimes not build
/opt/hpc/rpi/la/atlas/3.10.3/lib/libtatlas.so # sometimes not build
```

2.1 Make time

As root:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq
1500000
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
1500000
```

As user

```
make time
make -f Make.top time
make[1]: Entering directory '/opt/hpc/rpi/la/atlas/3.10.3/bld'
./xatlbench -dc /opt/hpc/rpi/la/atlas/3.10.3/bld/bin/INSTALL_LOG \
-dp /opt/hpc/rpi/la/atlas/3.10.3/bld/ARCHS/UNKNOWN64
Enter Clock rate in Mhz [0]: 1500

The times labeled Reference are for ATLAS as installed by the authors.
NAMING ABBREVIATIONS:
   kSelMM : selected matmul kernel (may be hand-tuned)
```

```
   kGenMM : generated matmul kernel
   kMM_NT : worst no-copy kernel
   kMM_TN : best no-copy kernel
   BIG_MM : large GEMM timing (usually N=1600); estimate of asymptotic peak
   kMV_N  : NoTranspose matvec kernel
   kMV_T  : Transpose matvec kernel
   kGER   : GER (rank-1 update) kernel
Kernel routines are not called by the user directly, and their
performance is often somewhat different than the total
algorithm (eg, dGER perf may differ from dkGER)


Clock rate=1500Mhz
              single precision        double precision
            ********************    ********************
               real      complex       real      complex
Benchmark   %   Clock   %   Clock   %   Clock   %   Clock
=========   =========   =========   =========   =========
  kSelMM       460.6       405.2       291.5       276.6
  kGenMM       154.6       152.4       147.4       135.9
  kMM_NT       142.4       136.8       126.0       121.8
  kMM_TN       150.4       145.2       133.8       133.1
  BIG_MM       430.2       425.7       282.5       286.9
   kMV_N        84.6       126.6        66.2        92.9
   kMV_T        99.3       126.5        61.3       109.6
    kGER        44.9        89.9        22.0        48.6
make[1]: Leaving directory '/opt/hpc/rpi/la/atlas/3.10.3/bld'
```

3 Installation from Package Dependencies

Installing `atlas` on Debian 11 (Bullseye) will also pull in `mpich`.

```
aptitude install libatlas-base-dev libmpich-dev gfortran
```

This will install:

```
1  gfortran gfortran-10{a} hwloc-nox{a} libatlas-base-dev libatlas3-base{a}
2  libgfortran-10-dev{a} libhwloc-plugins{a} libhwloc15{a} libmpich-dev
3  libmpich12{a} libslurm36{a} libxnvctrl0{a} mpich{a}
```

4 History

Version	Date	Notes
0.1.2	2023-01-26	Improve writing
0.1.1	2023-01-25	Note for package installation of Atlas
0.1.0	2022-06-19	Initial release

5 Disclaimer of Warranty

6 Limitation of Liability