

Using AVX Vector Extensions on Discovery

Introduction:

The X86 instruction set provide the Advanced Vector Extensions (AVX) on microprocessors from both Intel and AMD. The first Intel CPU to support AVX was Sandy Bridge in 2011 and the first AMD CPU to support AVX was Bulldozer in 2011.

AVX2 was first supported on the Intel Haswell processor, expanding most integer operations to 256 bits. AVX-512 was introduced on Intel's SkyLake CPUs and expands AVX to 512-bit support in 2017. Wikipedia provides a detailed list of Intel and AMD X86 CPUs that support AVX, AVX2 and AVX-512 [1].

To run AVX on Discovery:

The Northeastern Discovery system provides a variety of AVX options for our class. The systems in our class reservation are Haswell CPUs, so have support for AVX2. AMD zen2 nodes only have AVX2 as well. The best way to get AVX-512 support is to include:

```
#SBATCH --constraint=cascadelake
```

To get access to an interactive node:

```
srun --partition=short --constraint=cascadelake --nodes=1 --pty /bin/bash
```

In terms of C compiler support, gcc >= 4.9 should support AVX-512, but C++ support should use gcc 10.1 or later. So make sure you are using version 10.1.0 by issuing:

```
module add gcc/10.1.0.
```

While compiling programs, add the compiler flag `-march=native` to enable all native extensions provided by the local CPU including AVX.

WARNING: This might make your program not compatible with other platforms and might require recompilation.

[1] Advanced Vector Extensions – Wikipedia, URL:

https://en.wikipedia.org/wiki/Advanced_Vector_Extensions