AN AVX 512 EXTENSION TO OPENQCD

Jarno rantaharju, Mark Dawson, Ed Bennett and Michele Mesiti

Swansea Academy of Advanced Computing

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Introduction

- AVX 512
  - On new Intel chips
  - SIMD operations on 16 floats / 8 doubles
  - 32 floating point registers

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Introduction

- OpenQCD
  - luscher.web.cern.ch/luscher/openQCD/
  - Widely used, open source
- Fastsum Extension
  - Talk by Jonas Glesaaen (Friday at 17:10)
  - Finite temperature, Anisotropy
  - Modules as libraries

An AVX 512 Extention to OpenQCD

- Available at github.com/sa2c/OpenQCD-AVX512
- The Fastsum code: gitlab.com/fastsum/openqcd-fastsum
- Arxiv report: 1806.06043

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Implementation

- Targets: KNL and Skylake
- Memory bandwidth bound
  - Attention to register memory use
  - and cache use
  - at the expense of computation

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Implementation

- Intrinsics
  - Control of instructions used in C code

```
v3 = _mm512_mul_ps( v1, v2 );
```

Compiles into

```
vmulps zmm1, zmm2, zmm5
```

  - Flexible, can compile on different systems
  - Usually a one-to-one correspondence
  - Compiler manages registers

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Compared Versions

- Skylake:
  - The AVX 512 code with the Intel compiler (ICC)

    ```
    -O3 -xCORE-AVX512 -mtune=skylake -DAVX512
    ```

  - The original AVX2 code with GCC (AVX)

    ```
    -O3 -march=skylake-avx512 -DPM -DAVX -DFMA3
    ```

  - ICC assembly and AVX2 code (combined)

    ```
    -O3 -march=skylake-avx512 -DPM -DAVX -DAVX512
    -DAVX512_ASM
    ```

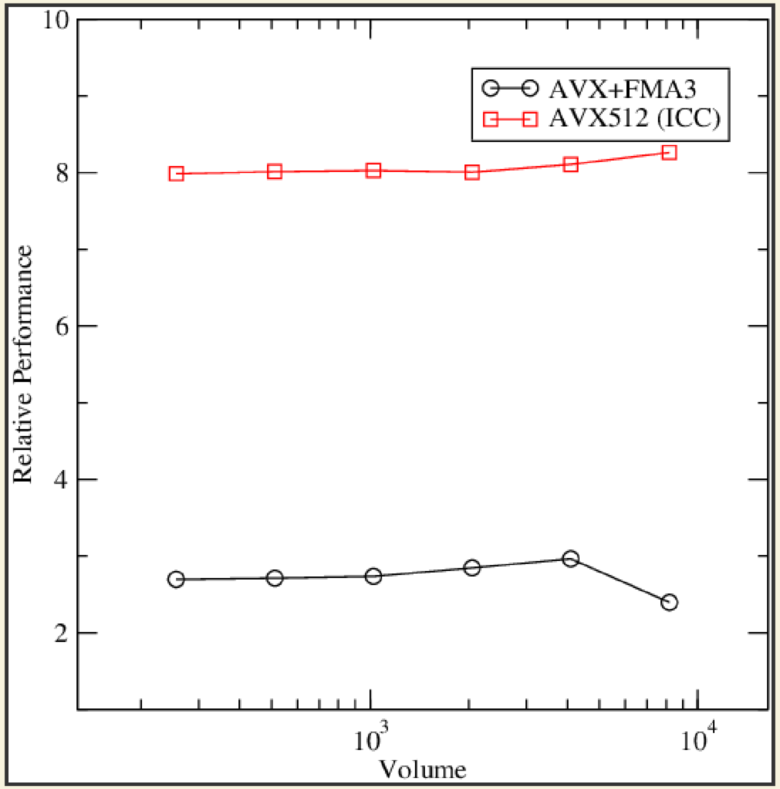
SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Note

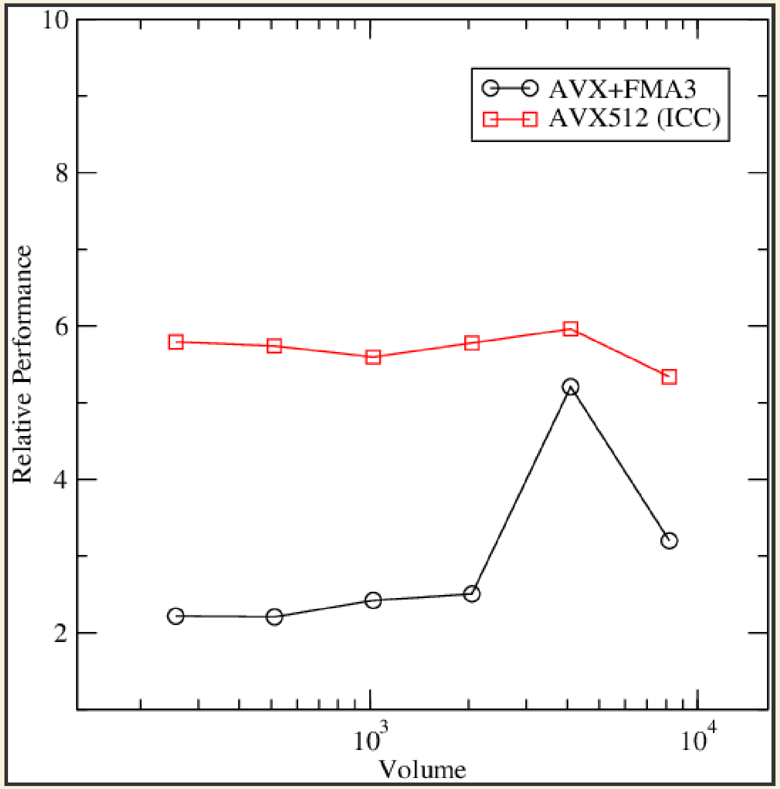
- Assembly files
  - enable including the AVX functions
  - included for convenience for Skylake cores
  - can be easily replaced for other systems

Single Core Performance

Against vanilla baseline



single Precision

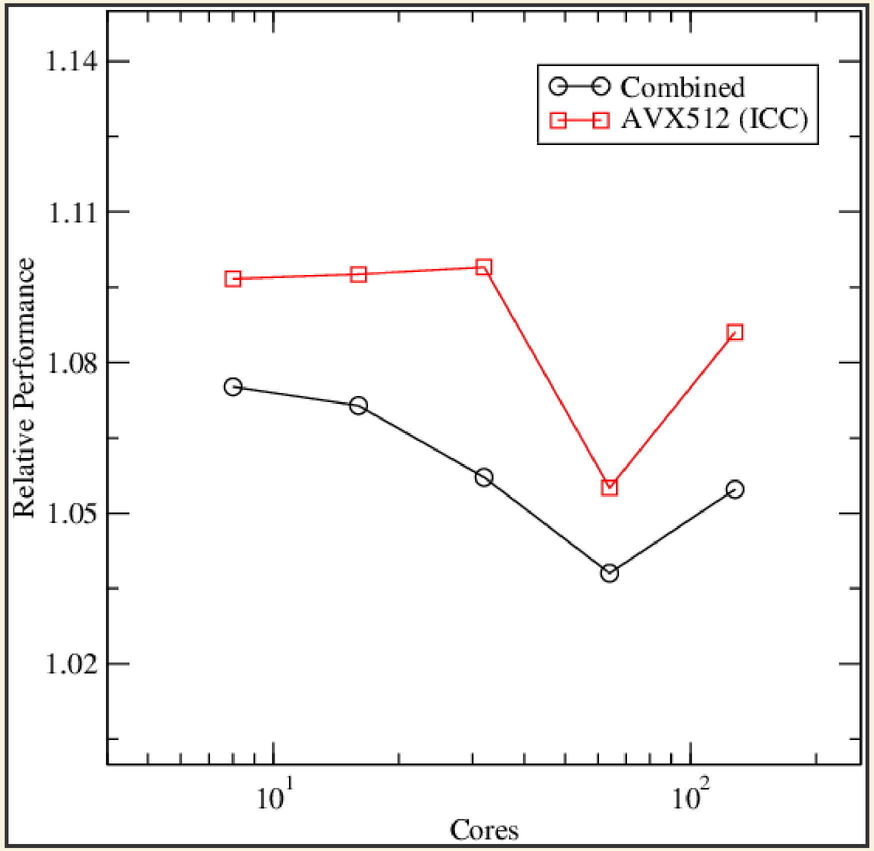


Double Precision

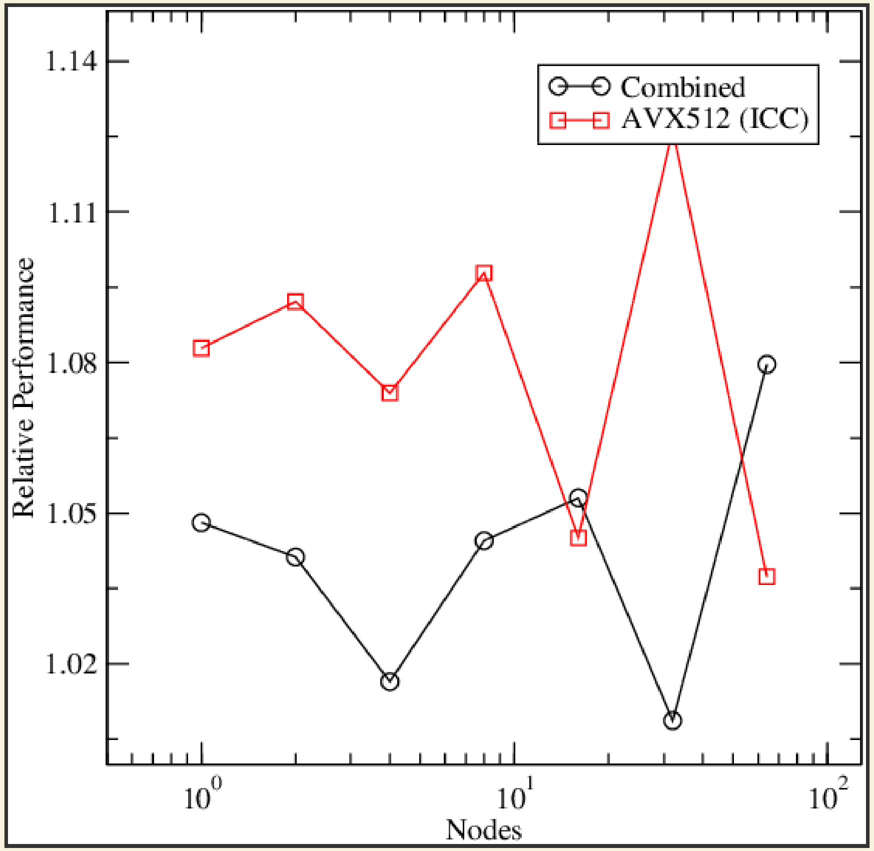
SAC[2]

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Strong Scaling on a Skylake Cluster



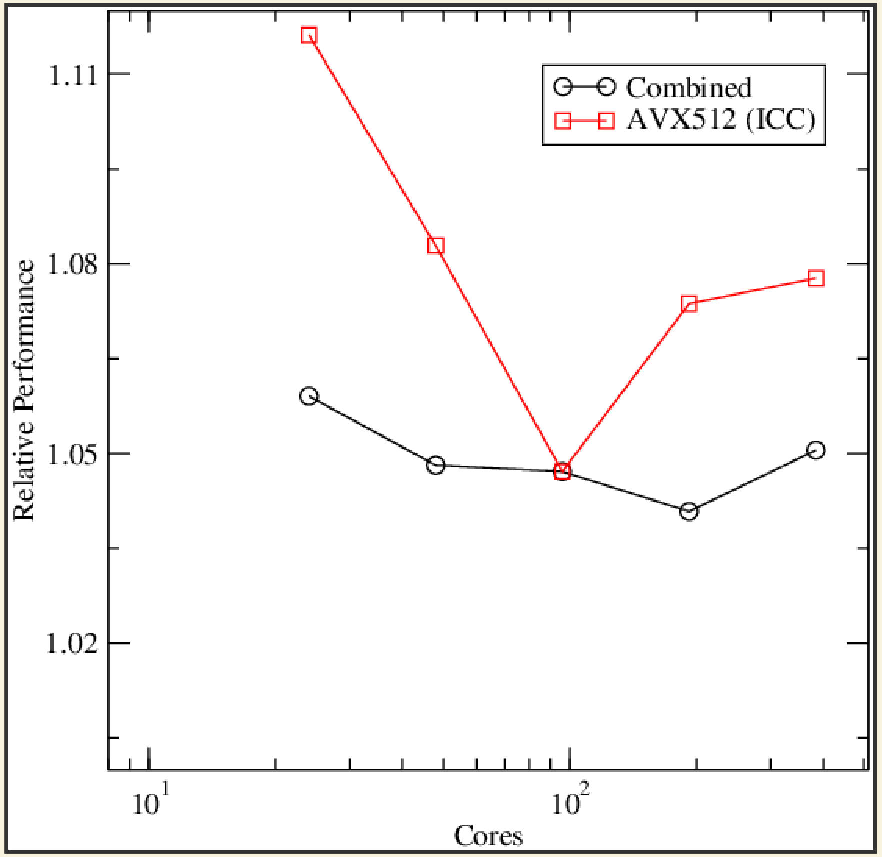
$V=24^3x48$



$V=24^3x48$

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Weak Scaling on a Skylake Cluster



$V = 24^3 \mathrm{x}\, N_{cores}$

Compared Versions

- Xeon Phi:
  - The AVX 512 code with the Intel compiler (ICC)

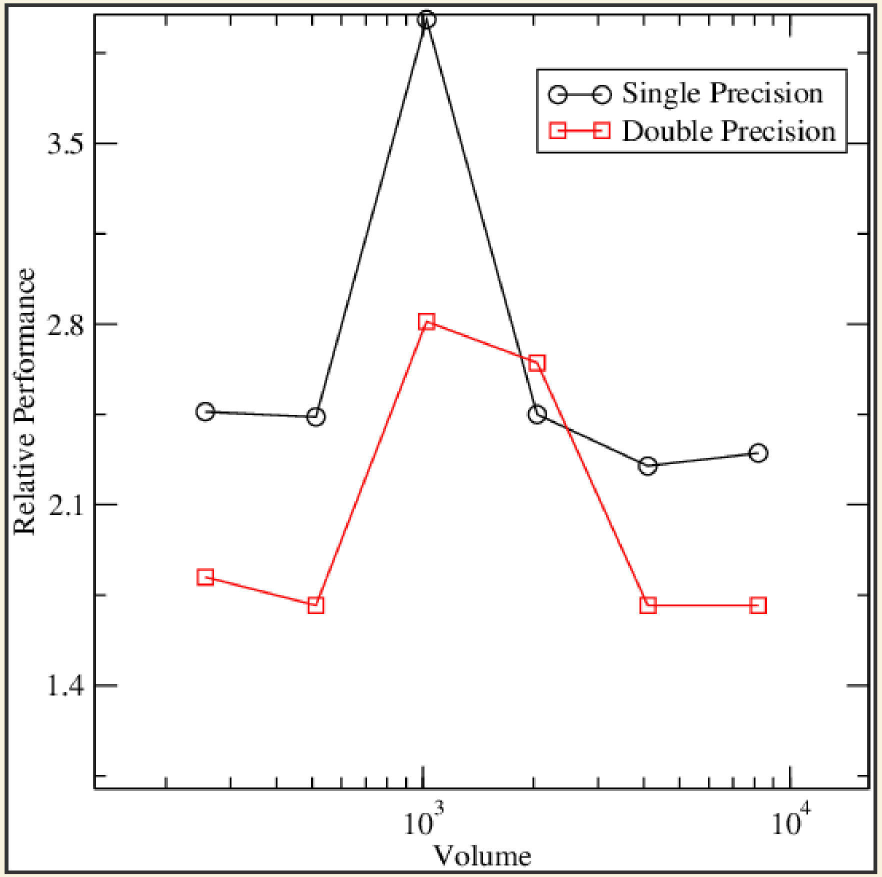
```
-xCORE-AVX512 -mtune=skylake -O3 -DAVX512
```

  - Standard C code with the Intel compiler (Vanilla)

```
-xCORE-AVX512 -mtune=skylake -O3
```

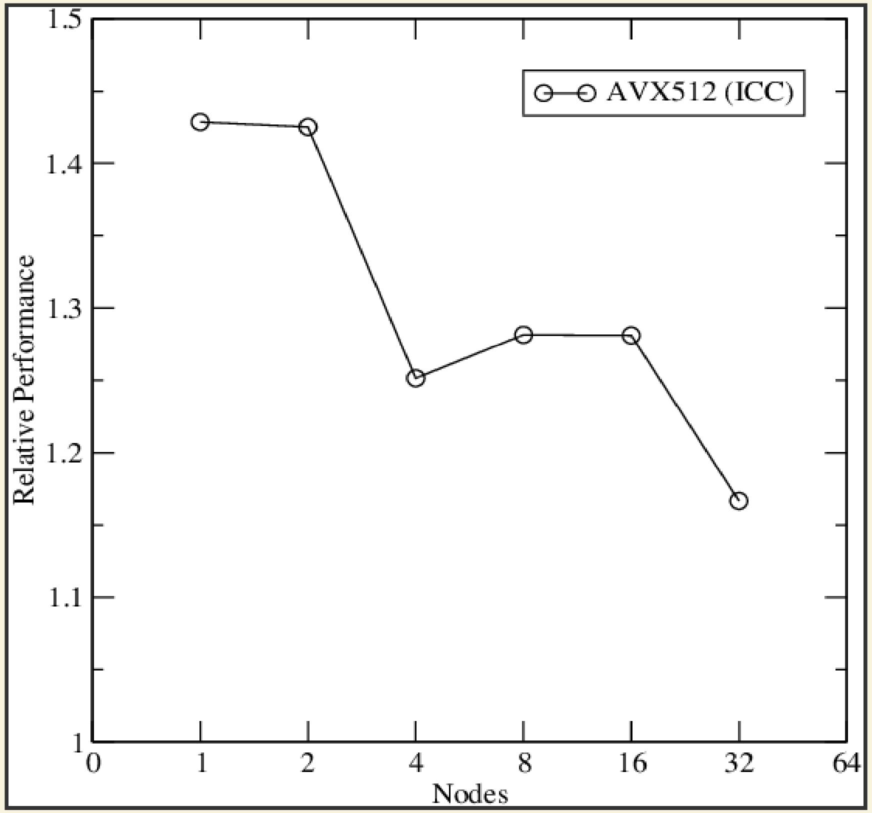
SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Single Core Performance on a Xeon phi

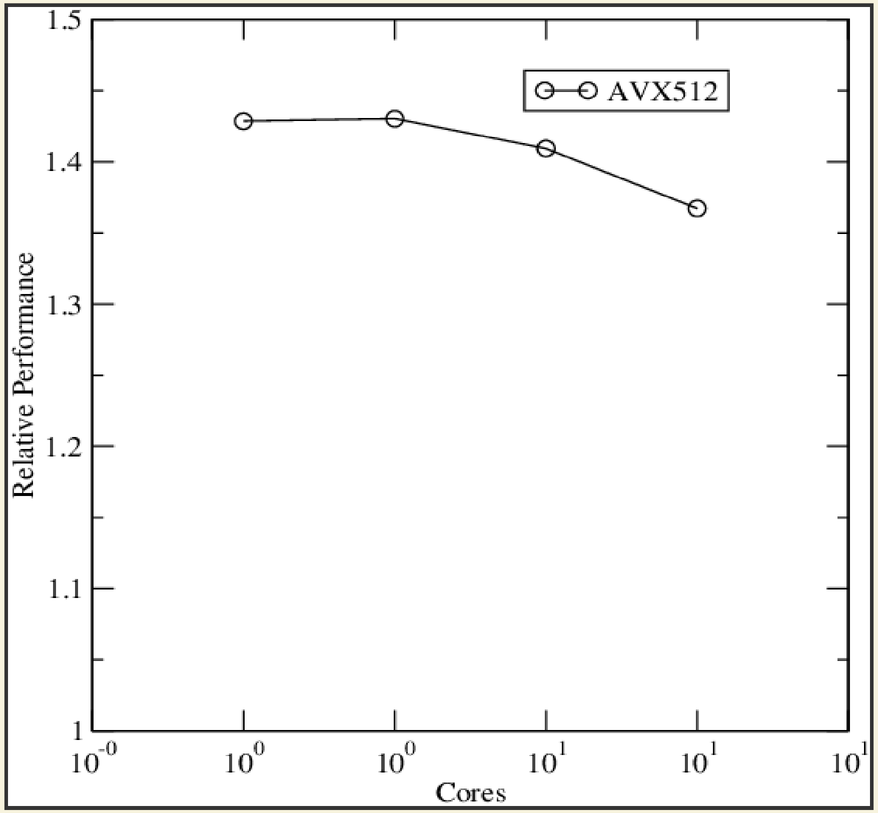


SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Scaling on a Xeon Phi Cluster



Strong Scaling, $V=32^4$



Weak Scaling, $V=32^3 x 32N$

SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Wrap-Up

- AVX 512 implementation to openQCD
  - github.com/sa2c/OpenQCD-AVX512
- Targets: KNL and Skylake
  - 5% to 10% improvement in Skylake
  - 20% to 40% against vanilla on Xeon Phi
- Memory bandwidth bottleneck
  - Independent operations might perform better