

An Introduction to hpxMP – A Modern OpenMP Implementation Leveraging HPX, An Asynchronous Many-Task System

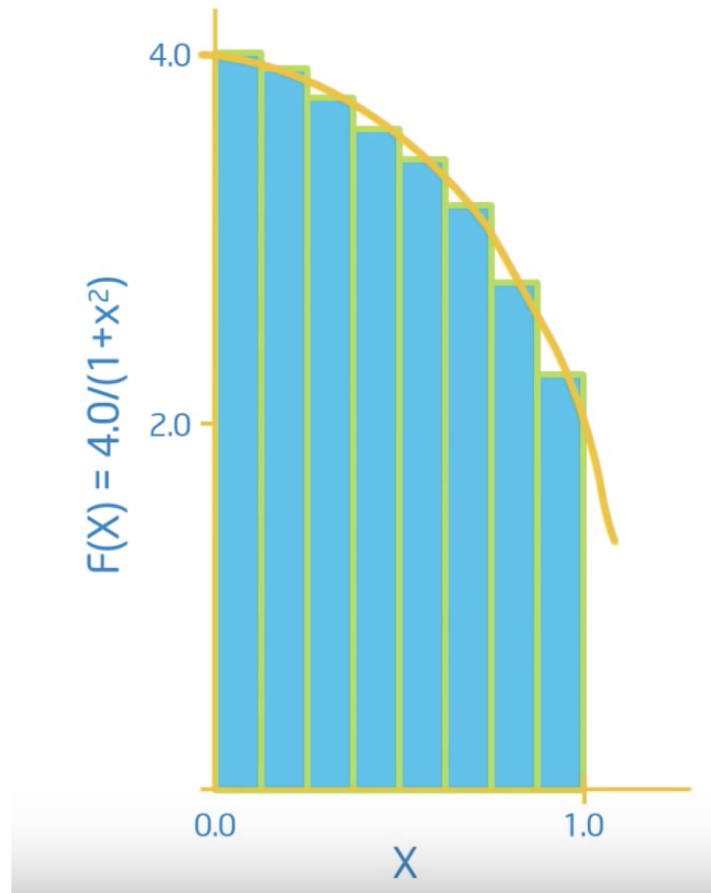
Zhang, T., Shirzad, S., Diehl, P., Tohid, R., Wei, W., Kaiser, H

Jeremy Kemp, Tianyi Zhang, Bryce Adelstein, Shahrzad Shirzad, Hartmut Kaiser, Parsa Amini, & Bibek Wagle. (2018, November 1). hpxMP (Version 0.1.0). Zenodo. <http://doi.org/10.5281/zenodo.2662481>

Presenter: Tianyi Zhang M.S. CS

A Simple OpenMP Program

$$\pi = 4 \int_0^1 \frac{1}{1+x^2} dx$$



```
step = 1.0/(double) steps;  
for (i=0; i < steps; i++) {  
    x = i*step + 0.5*step;  
    sum += 4.0 / (1.0+x*x);  
}  
pi = step * sum;
```

```
#pragma omp parallel for reduction(+:sum) private(x)  
for (i=0; i < steps; i++) {  
    x = i*step + 0.5*step;  
    sum += 4.0 / (1.0+x*x);  
}
```

[https://www.openmp.org/wp-content/uploads/Intro To OpenMP Mattson.pdf](https://www.openmp.org/wp-content/uploads/Intro_To_OpenMP_Mattson.pdf)

Why hpxMP ?

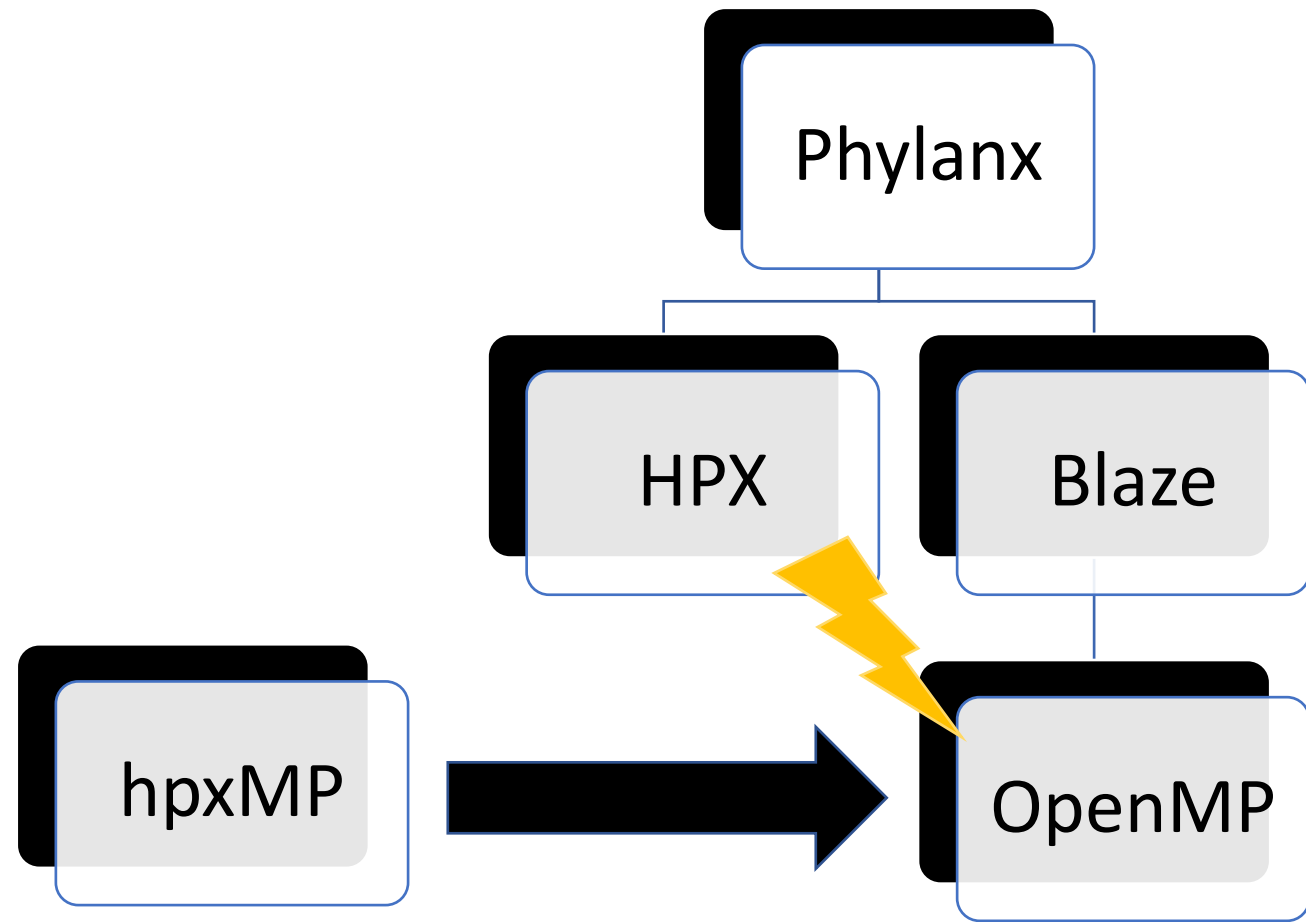
AMT

OpenMP

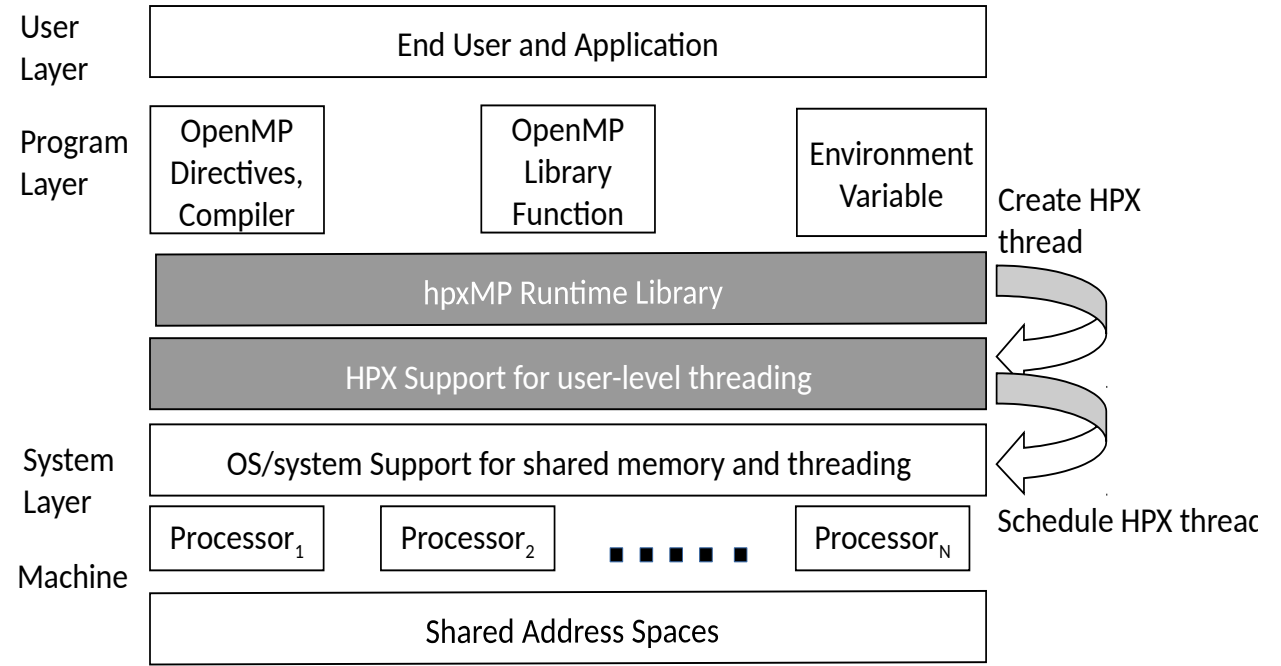
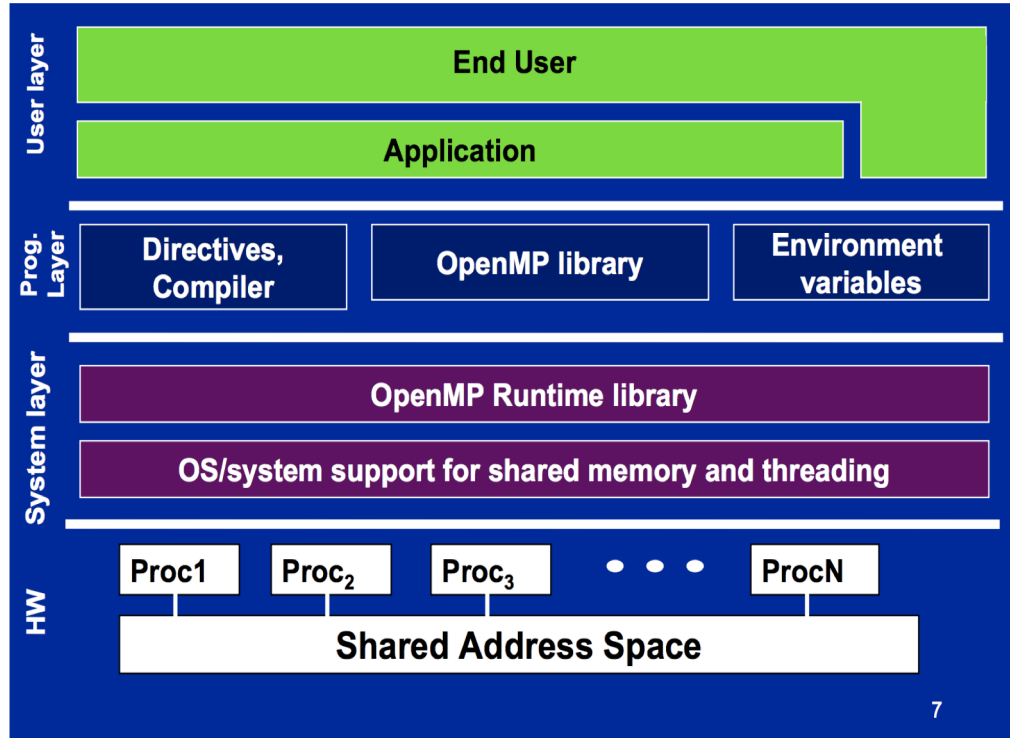


HPX is a C++ Standard Library for Concurrency and Parallelism. It implements all of the corresponding facilities as defined by the C++ Standard. Additionally, in HPX we implement functionalities proposed as part of the ongoing C++ standardization process. We also extend the C++ Standard APIs to the distributed case.

<https://github.com/STELLAR-GROUP/hpx>



Some Idea ...



[https://www.openmp.org/wp-content/uploads/Intro To OpenMP Mattson.pdf](https://www.openmp.org/wp-content/uploads/Intro_To_OpenMP_Mattson.pdf)

OpenMP 5.0

```
#pragma omp taskgroup task_reduction ( operator : list )
```

```
omp_set_num_threads(int);          #pragma omp barrier
```

[https://github.com/STELLAR-](https://github.com/STELLAR-GROUP/hpxMP)
[GROUP/hpxMP](https://github.com/STELLAR-GROUP/hpxMP)

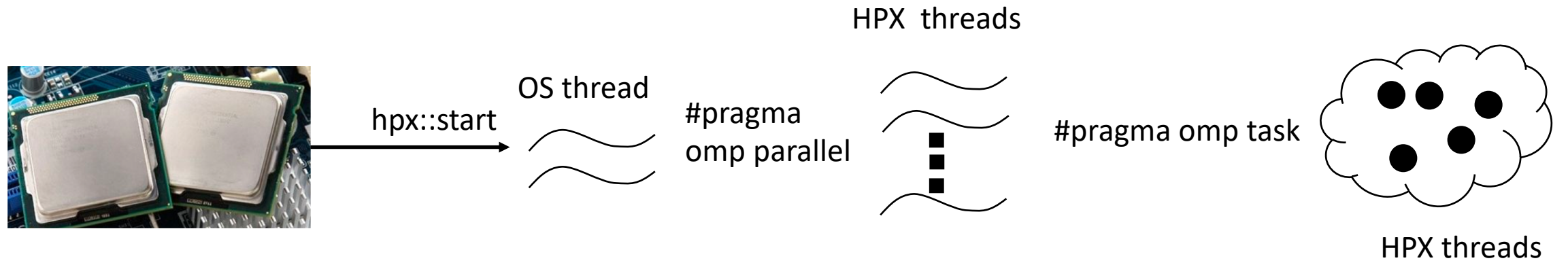
```
omp_get_num_threads()
```

```
#pragma omp task
```

```
#pragma omp parallel for
```

```
#pragma omp parallel
```

- Compile your program as a openMP program `clang || gcc`
- `LD_PRELOAD = .../libhpxmp.so`
- `OMP_NUM_THREADS = 2`



Examples ...


```
int main() {  
    int i = 0;  
    #pragma omp parallel for schedule(static)  
    for(i = 0; i < 11; i++)  
    {  
        printf("Hello World #%d, from thread: %d\n", i, omp_get_thread_num());  
    }  
    return 0;  
}
```

```
HPX OpenMP runtime has started  
Hello World #6, from thread: 1  
Hello World #7, from thread: 1  
Hello World #8, from thread: 1  
Hello World #9, from thread: 1  
Hello World #0, from thread: 0  
Hello World #1, from thread: 0  
Hello World #2, from thread: 0  
Hello World #3, from thread: 0  
Hello World #4, from thread: 0  
Hello World #5, from thread: 0  
Hello World #10, from thread: 1  
Stopping HPX OpenMP runtime  
Stopped
```

Deeper into...
`#pragma omp parallel`

```
void  
__kmpc_fork_call(ident_t *loc, kmp_int32 argc, kmpc_micro microtask, ...)
```

```
void  
xexpand(KMP_API_NAME_GOMP_PARALLEL)(void (*task)(void *), void *data, unsigned num_threads, unsigned int flags)
```

```
for( int i = 0; i < parent->threads_requested; i++ ) {  
    hpx::applier::register_thread_nullary(  
        std::bind( &thread_setup, kmp_invoke, thread_func, argc, argv, i, &team, parent,  
                  boost::ref(barrier_mtx), boost::ref(cond), boost::ref(running_threads) ),  
        "omp_implicit_task", hpx::threads::pending,  
        true, hpx::threads::thread_priority_low, i );  
        //true, hpx::threads::thread_priority_normal, i );  
}
```

Time check Point
Questions?

Performance ?

hpxMP vs llvm-OpenMP

Table 4: System configuration of the marvin node. All benchmarks are run on this node.

Category	Property
Server Name	Rostam
CPU	2 x Intel(R) Xeon(R) CPU E5-2450 0 @ 2.10GHz
RAM	48 GB
Number of Cores	16

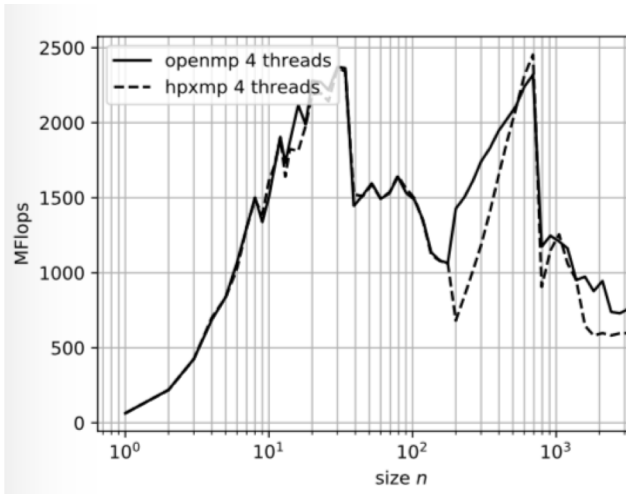
Table 5: Overview of the compilers, software, and operating system used to build hpxMP, HPX, Blaze and its dependencies.

Category	Property
OS	CentOS Linux release 7.6.1810 (Core)
Kernel	3.10.0-957.1.3.el7.x86_64
Compiler	clang 6.0.1
gperftools	2.7
boost	1.68.0
OpenMP	3.1
HPX ¹¹	140b878
Blaze ¹²	3.4

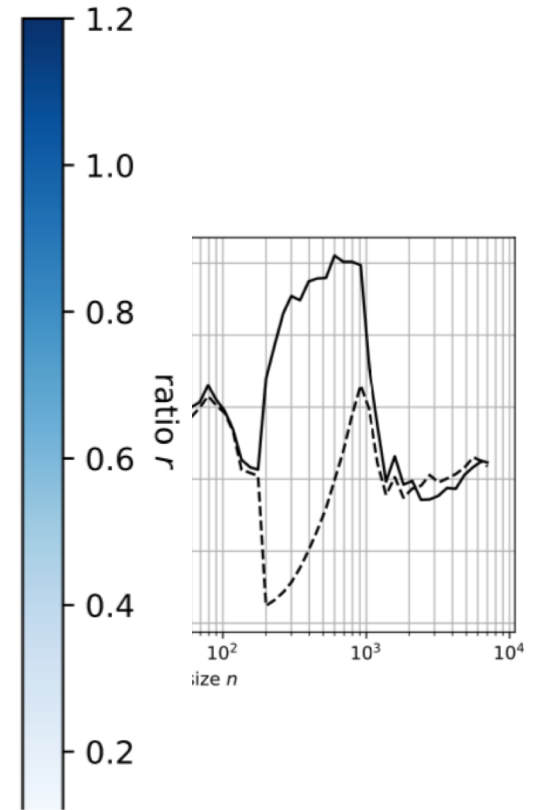
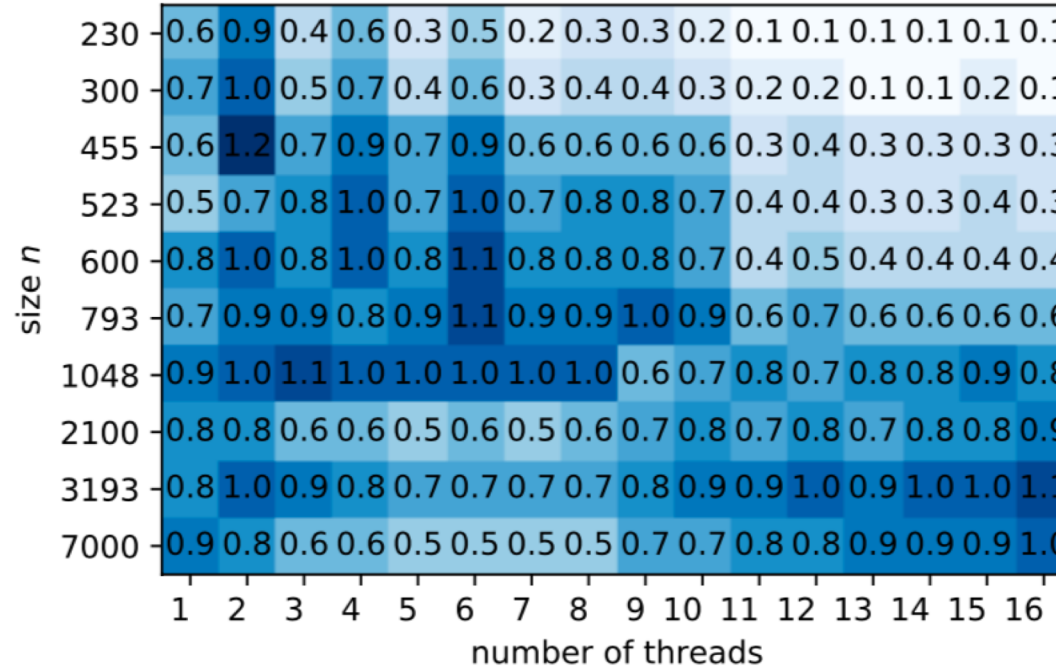


<http://www.hpc.lsu.edu/resources/hpc/system.php?system=QB2>

BLAZEMARK DMATDMATADD



Performance Ratio (higher number means better performance of hpxMP)



Why hpxMP is fast ?

Join Phase in user level

Why hpxMP is slow ?

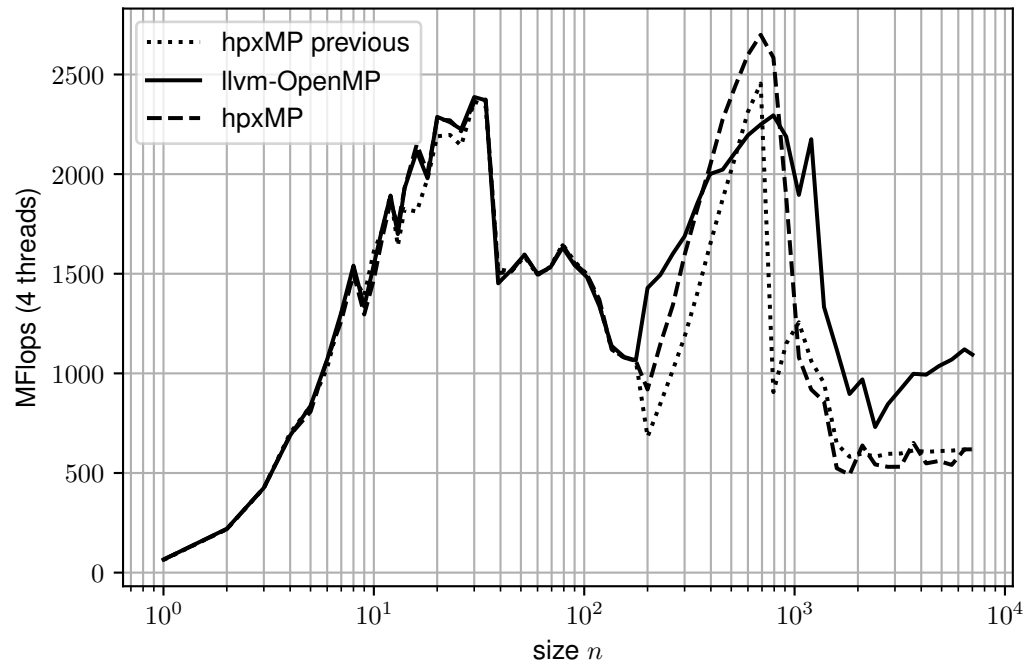
HPX Scheduler

OMPT (OpenMP Performance Toolkit)

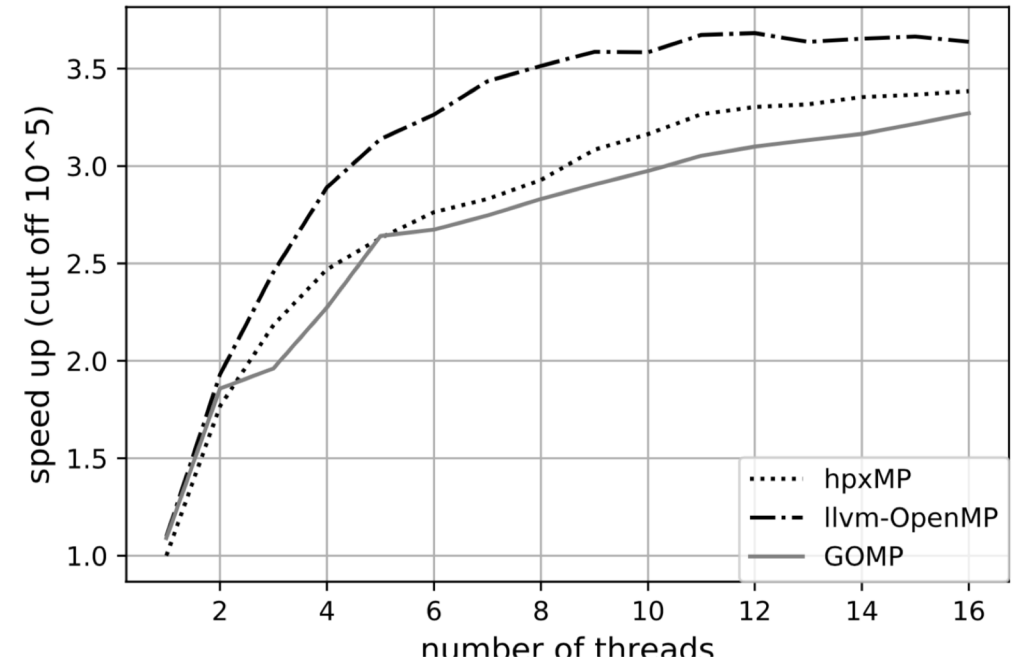
Conclusions and Future Work

- Implement OpenMP using AMTs
- compatibility gap between OpenMP and AMT systems

- non-suspending threads to HPX
- OpenMP task-based benchmark



BLAZEMARK DMATDMATADD



Barcelona OpenMP Task Suit
Sort of 10^7 elements

- Acknowledgment

We would like to thank Jeremy Kemp for providing the initial implementation of hpxMP which was extended by the authors. The work on hpxMP is funded by the National Science Foundation (award 1737785) and the Department of Defense (DTIC Contract FA8075-14-D-0002/0007). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Department of Defense.

- Tianyi Zhang
- tzhan18@lsu.edu