

OpenMPIR

Implementing OpenMP tasks with Tapir

George Stelle¹ William S. Moses²
Stephen L. Olivier³ Patrick McCormick¹

¹Los Alamos National Laboratory

²MIT CSAIL

³Sandia National Laboratories

November 13, 2017

Outline

- OpenMP
- Motivation
- Tapir
- Implementation
- Results
- Discussion
- Questions

OpenMP

```
int fib(int n){
    if (n < 2)
        return n;
    else {
        int x, y;
        #pragma omp task
            x = fib(n-1);
        #pragma omp task
            y = fib(n-2);
        #pragma omp taskwait
        return x+y;
    }
}
```

OpenMP LLVM IR

```
...  
%19 = call i32 @__kmpc_omp_task(%ident_t* nonnull @0, i32 %  
%20 = call i8* @__kmpc_omp_task_alloc(%ident_t* nonnull @0,  
%21 = bitcast i8* %20 to i8**  
%22 = load i8*, i8** %21, align 8, !tbaa !8  
...  
%28 = load i32, i32* %2, align 4, !tbaa !4  
store i32 %28, i32* %27, align 4, !tbaa !13  
%29 = call i32 @__kmpc_omp_task(%ident_t* nonnull @0, i32 %  
%30 = call i32 @__kmpc_omp_taskwait(%ident_t* nonnull @0,  
...
```





Tapir

detach
reattach
sync

```

int fib(int n){
...
    #pragma omp task
        x = fib(n-1);
    #pragma omp task
        y = fib(n-2);
    #pragma omp taskwait
...
}

```

```

...
if.end:
    detach label %det.achd, label %det.cont

det.achd:
    %2 = load i32, i32* %n.addr, align 4
    %sub = sub nsw i32 %2, 1
    %call = call i32 @fib(i32 %sub)
    store i32 %call, i32* %x, align 4
    reattach label %det.cont

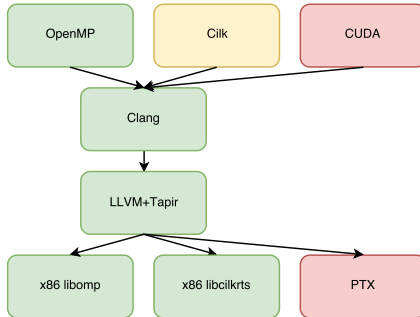
det.cont:
    detach label %det.achd1, label %det.cont4

det.achd1:
    %3 = load i32, i32* %n.addr, align 4
    %sub2 = sub nsw i32 %3, 2
    %call3 = call i32 @fib(i32 %sub2)
    store i32 %call3, i32* %y, align 4
    reattach label %det.cont4

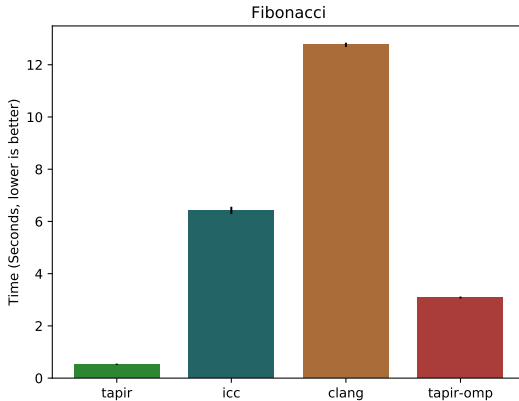
det.cont4:
    sync label %sync.continue
...

```

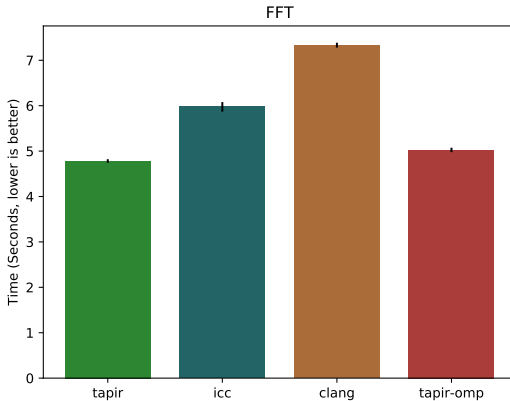
Overview



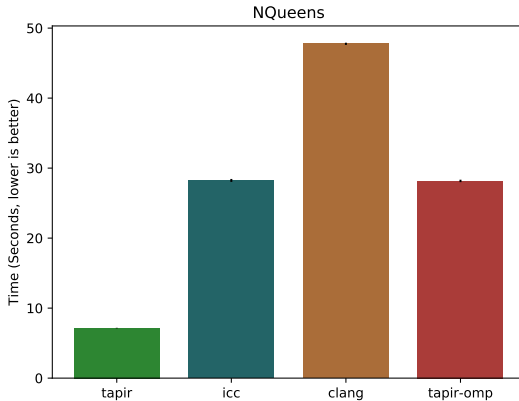
Results



Results



Results





The other 355 pages?

Questions?



Acknowledgements

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.