HPCC Benchmarks in Chapel

Brad Chamberlain, Sung-Eun Choi, Steve Deitz, David Iten

Cray Inc.

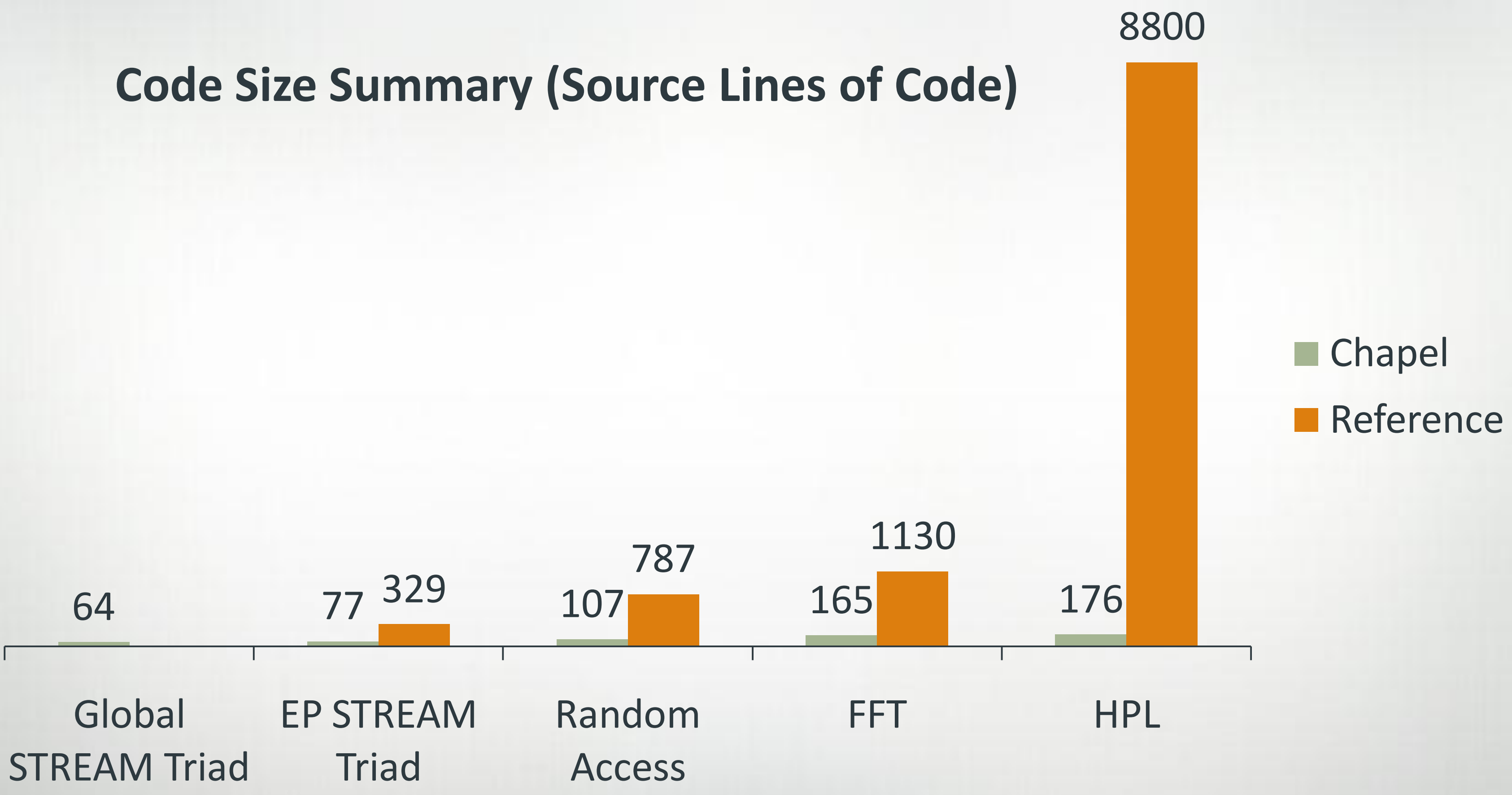
Chapel in a Nutshell

- What is Chapel?
  - A new parallel programming language from Cray Inc.
  - Developed as part of the DARPA HPCS program
- Past entries at HPCC BOF
  - HPCC 2006: Elegance-only entry
  - HPCC 2008:
    - First public performance numbers of STREAM and Random Access
    - First distribution ran only two months prior!

Highlights

- Global STREAM Triad 10.8 TB/s (6.4x over 2008)
  - Executed on 2048 nodes (up from 512 nodes in 2008)
  - Better scaling by eliminating extra communication
- EP STREAM Triad 12.2 TB/s
  - More similar to EP STREAM reference version
- Random Access 0.122 GUP/s (111x over 2008)
  - Executed on 2048 nodes (up from 64 nodes in 2008)
  - Optimized remote forks + better scaling as with STREAM
- A distributed-memory implementation of FFT
- A demonstration of portability
  - Cray XT4, Cray CX1, IBM pSeries 575, SGI Altix

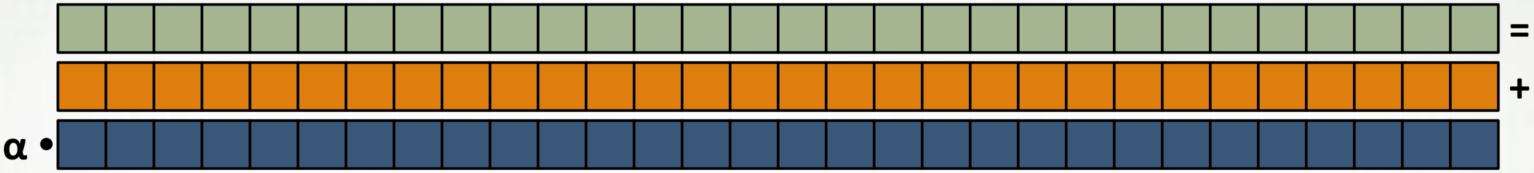
Chapel Implementation Characteristics



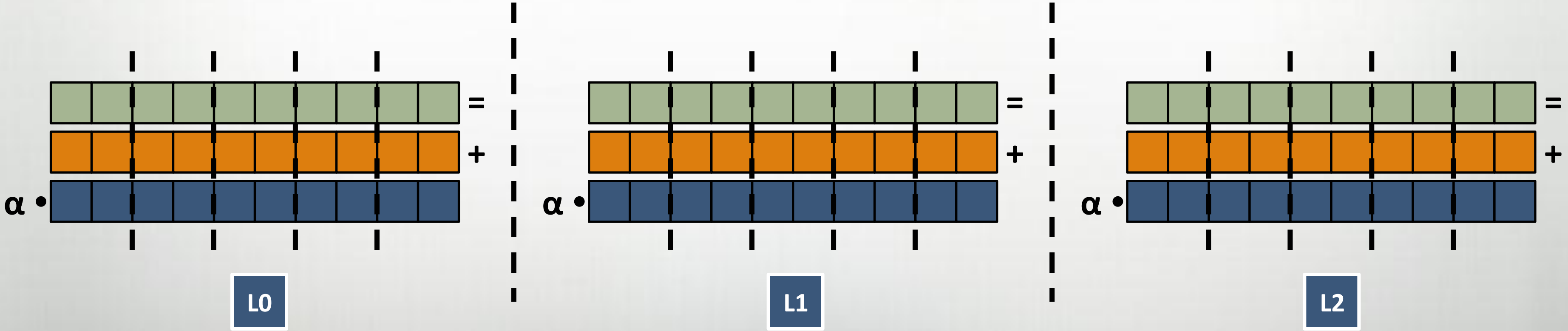
What is a Distribution?

A “recipe” for distributed arrays that...

Instructs the compiler how to map the global view...



...to a fragmented, per-processor implementation



Chapel Distributions

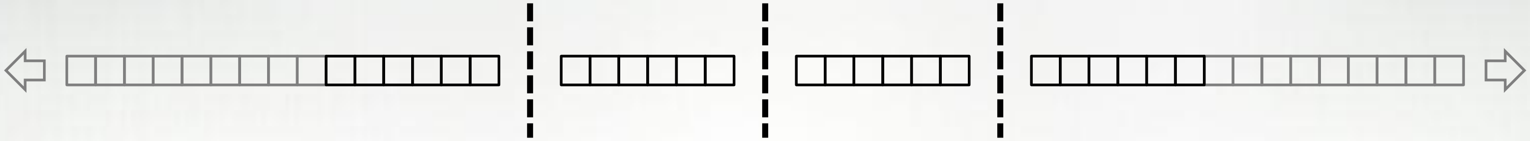
- Distributions are written entirely in Chapel
    - Advanced programmers can write their own
    - Classes define distributions, domains, and arrays
    - Compiler lowers code to a structural interface
    - Task-parallel constructs implement concurrency
- Standard distributions under development
    - Block
    - Cyclic
    - BlockCyclic
    - Associative
    - GPU

FFT and HPL in a Nutshell

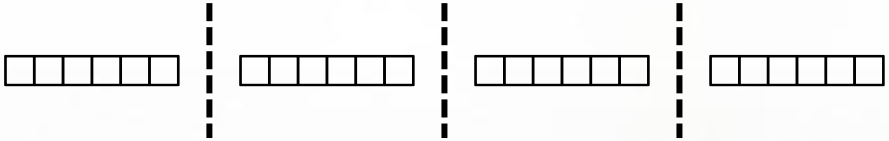
- FFT
  - Uses both Block and Cyclic distributions
  - Butterfly-patterned accesses are completely local
    - Communication with nearby neighbors is local with Block
    - Communication with far off neighbors is local with Cyclic
  - Executes on distributed memory, but is slow
- HPL
  - Implementation is ready for BlockCyclic distribution
  - Executes on single locale only, but is multi-threaded

Global STREAM Triad in Chapel (Excerpts)

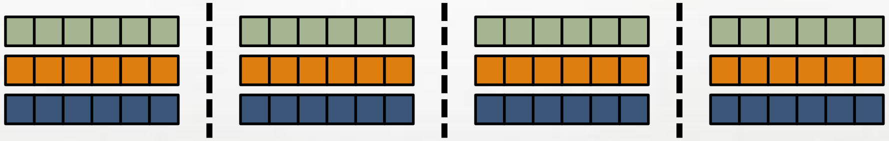
```
const BlockDist = distributionValue(new Block(…));
```



```
const ProblemSpace:
        domain(1,int(64)) distributed BlockDist = [1..m];
```



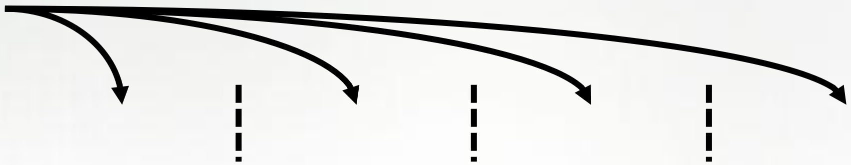
```
var A, B, C: [ProblemSpace] elemType;
```



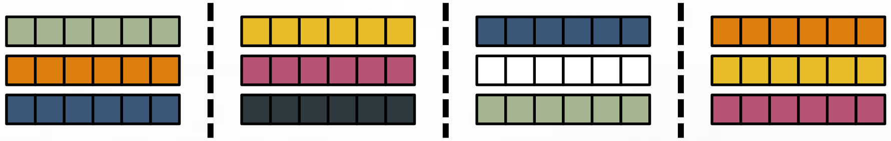
```
forall (a,b,c) in (A,B,C) do
  a = b + alpha * c;
```

EP STREAM Triad in Chapel (Excerpts)

```
coforall loc in Locales do on loc {
```



```
  local {
    var A, B, C: [1..m] elemType;
```



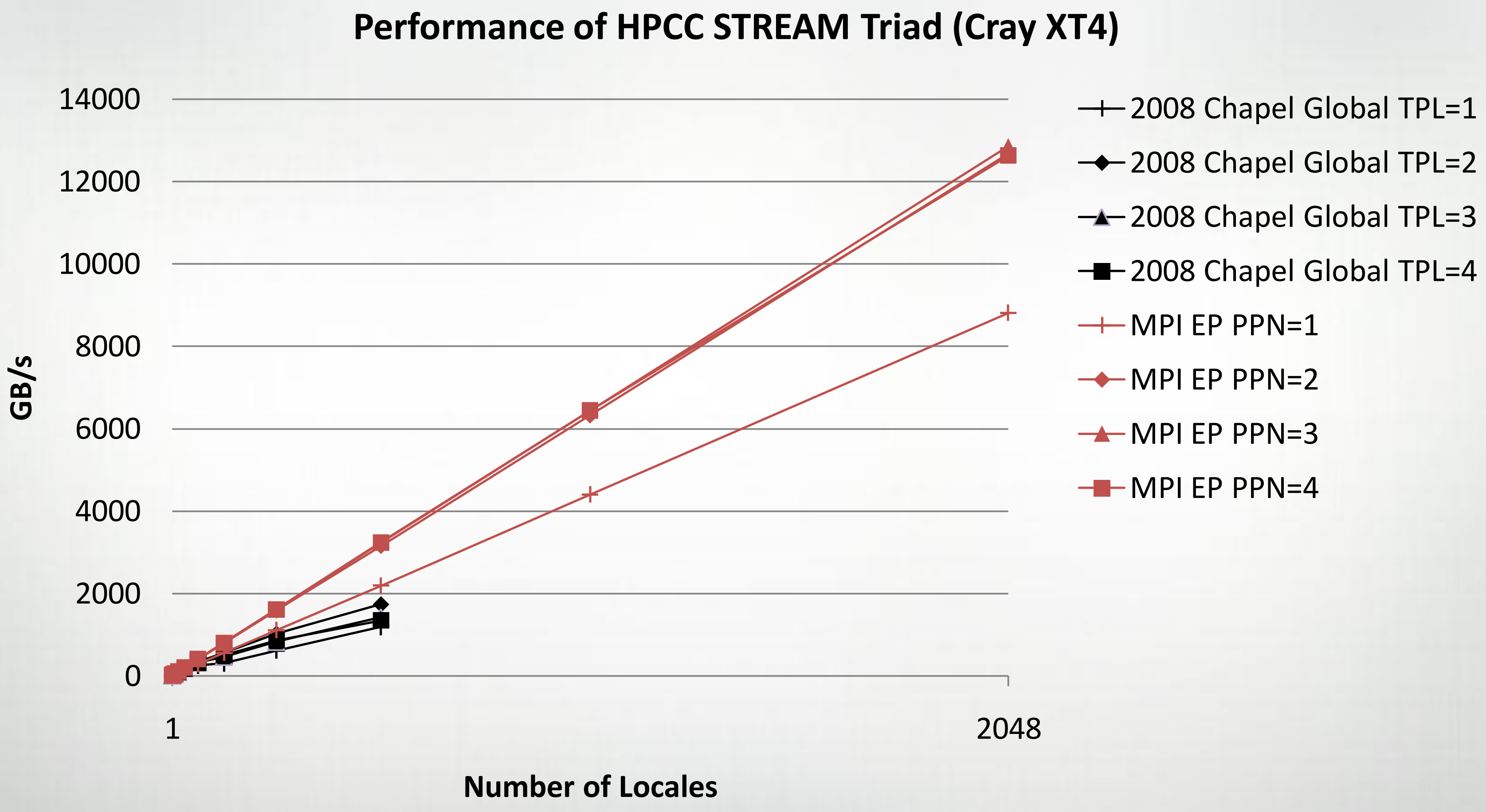
```
    forall (a,b,c) in (A,B,C) do
      a = b + alpha * c;
  }
}
```

Experimental Setup

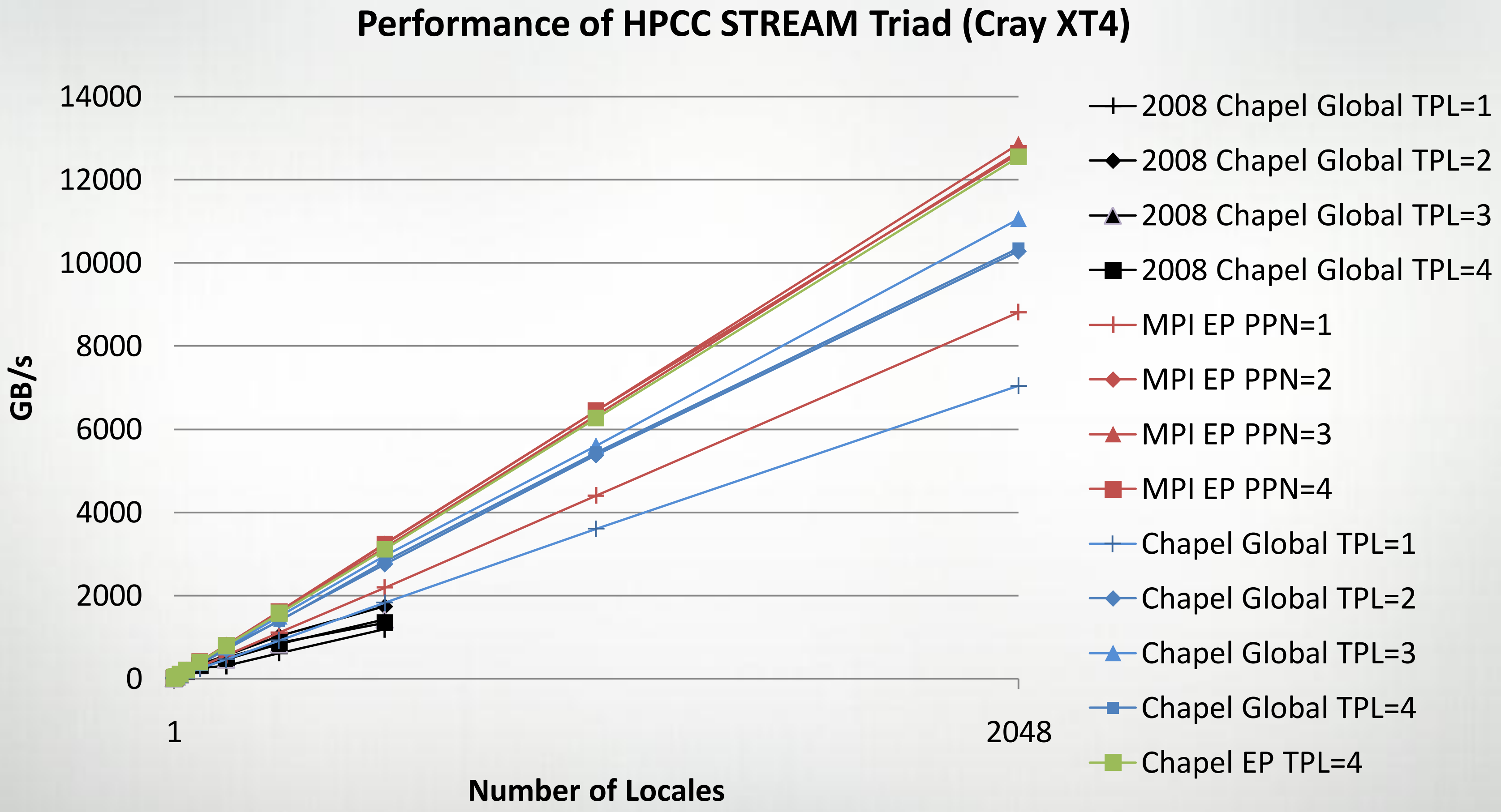
Machine Characteristics	
Model	Cray XT4
Location	ORNL
Nodes	7832
Processor	2.1 GHz Quadcore AMD Opteron
Memory	8 GB per node

Benchmark Parameters	
STREAM Triad Memory	Least value greater than 25% of memory
Random Access Memory	Least power of two greater than 25% of memory
Random Access Updates	2^{n-10} for memory equal to 2^n

STREAM Triad Performance



STREAM Triad Performance



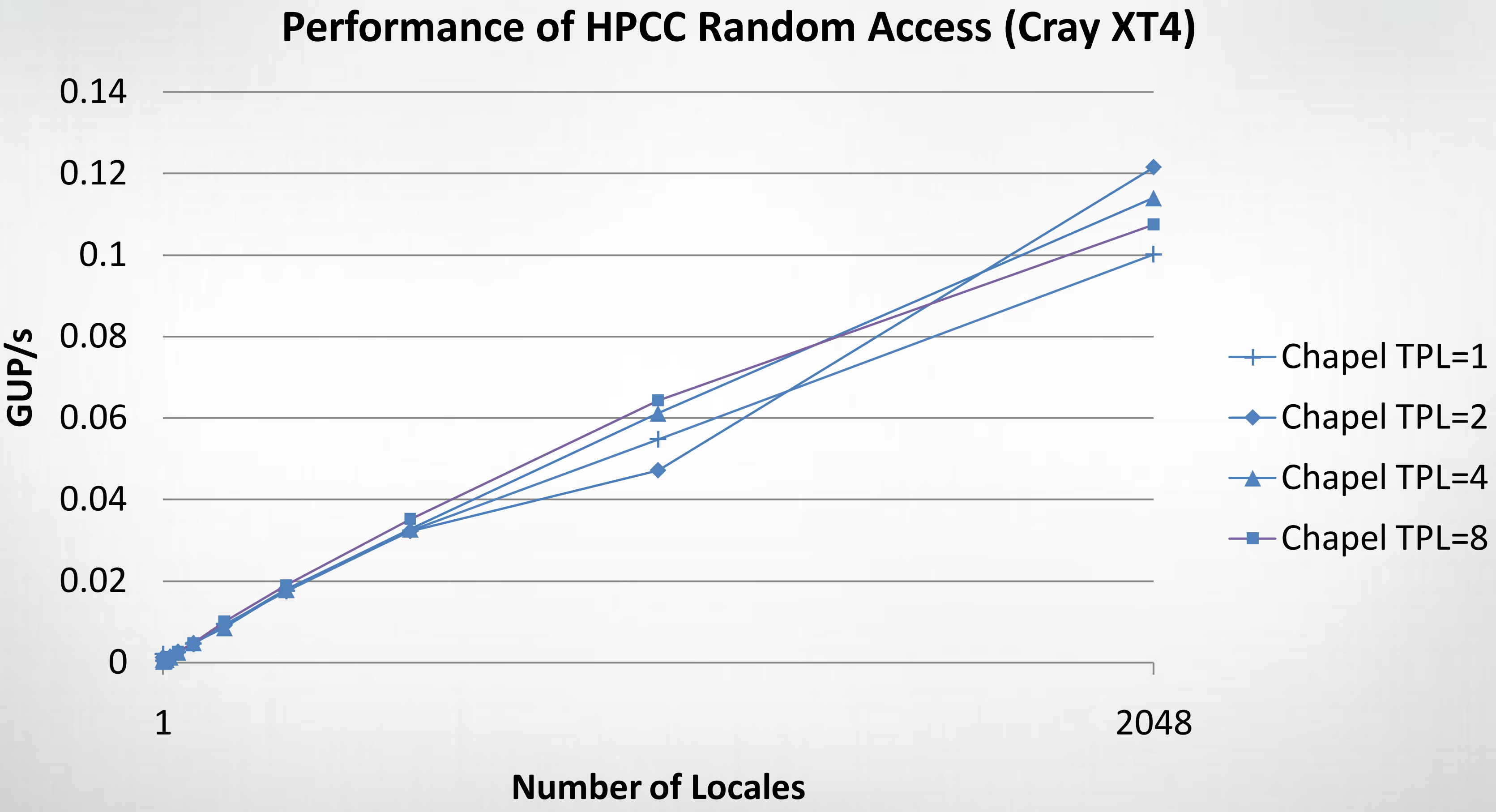
Global Random Access in Chapel (Excerpts)

```
const TableDist = distributionValue(new Block(…0..m…)),
      UpdateDist = distributionValue(new Block(…0..N_U…));

const TableSpace: domain … distributed TableDist = …,
      Updates: domain … distributed UpdateDist = …;

var T: [TableSpace] elemType;

forall (_,r) in (Updates,RAStream()) do
  on TableDist.ind2loc(r & indexMask) {
    const myR = r;
    local T(myR & indexMask) ^= myR;
  }
```

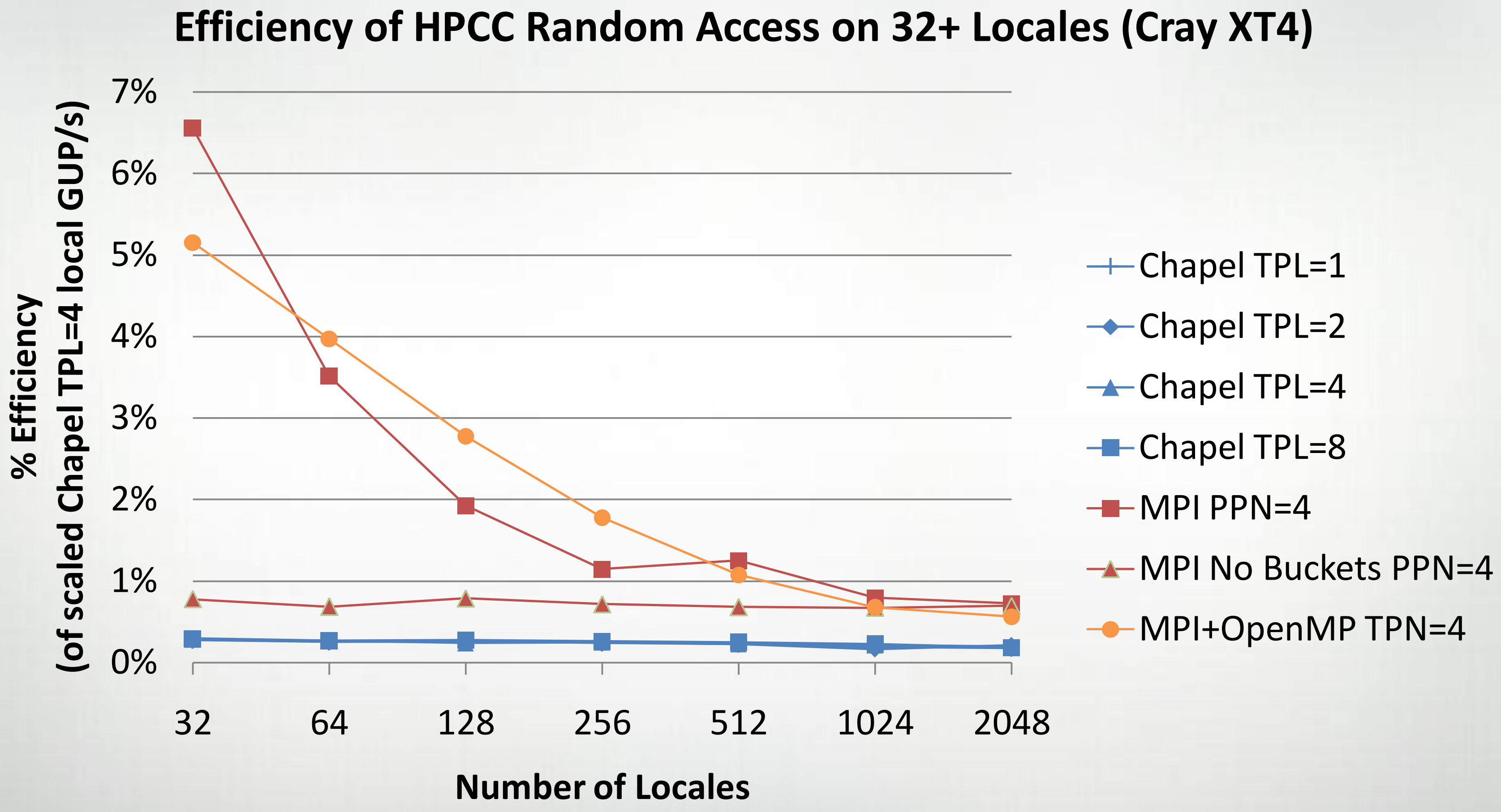
More elegant on-block

```
on T(r&indexMask) do
  T(r&indexMask) ^= r;
```

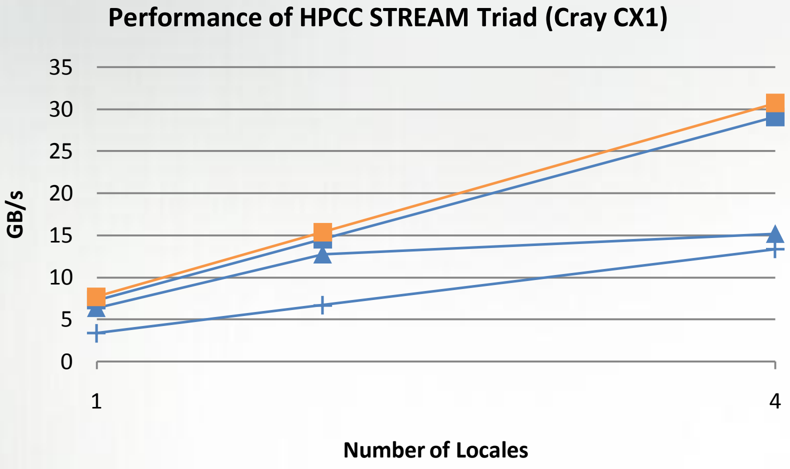
Random Access Performance

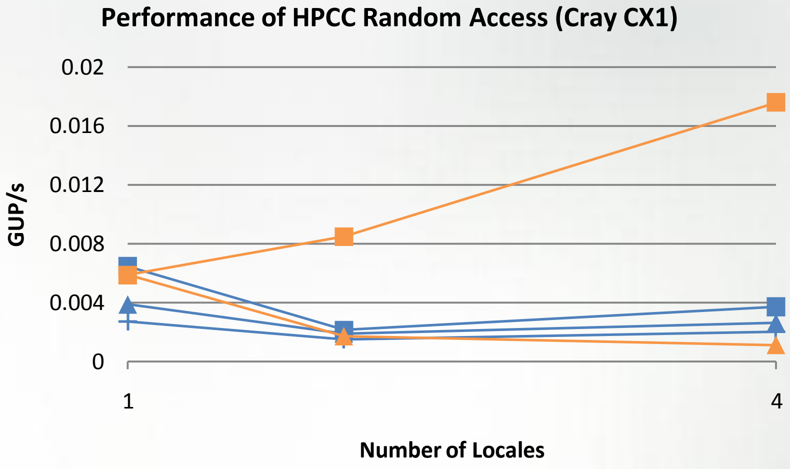


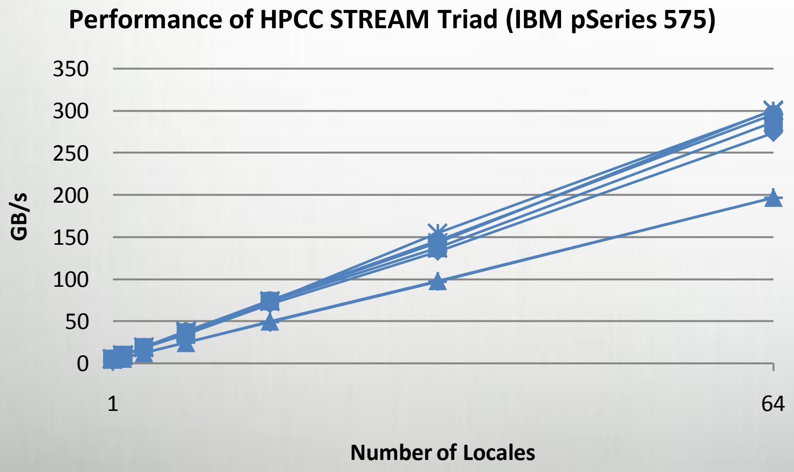
Random Access Efficiency on 32+ Nodes

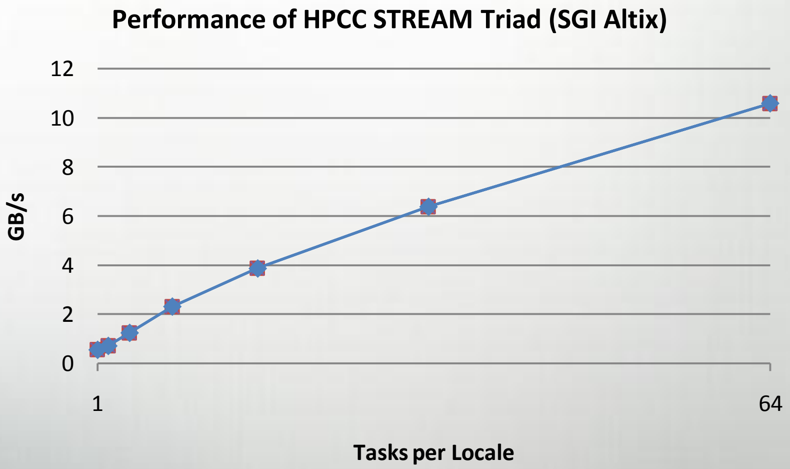


Portability Results


Performance of HPCC STREAM Triad (Cray CX1)


Performance of HPCC Random Access (Cray CX1)


Performance of HPCC STREAM Triad (IBM pSeries 575)


Performance of HPCC STREAM Triad (SGI Altix)

Summary

- Chapel is a work-in-progress
  - Performance is improving
  - Development of distributions is progressing

Score Card	Elegance	Performance
Global STREAM Triad	64 SLOC	10.8 TB/s
EP STREAM Triad	77 SLOC	12.2 TB/s
Random Access	107 SLOC	0.11 GUP/s
FFT	165 SLOC	0.00015 Gflop/s
HPL	176 SLOC	Multi-threaded, single-locale

No library routines were used in this entry.

Thanks

http://chapel.cray.com/

http://sourceforge.net/projects/chapel/

chapel_info@cray.com

Chapel is at Supercomputing ’09:

- All-Day PGAS Tutorial Monday
- HPCC BOF (Now)
- PGAS Booth (Tuesday, 2-4 PM)
- PGAS BOF (Tuesday, 5:30 PM)