Chapel: Distributions and Layouts

Sung-Eun Choi and Steve Deitz

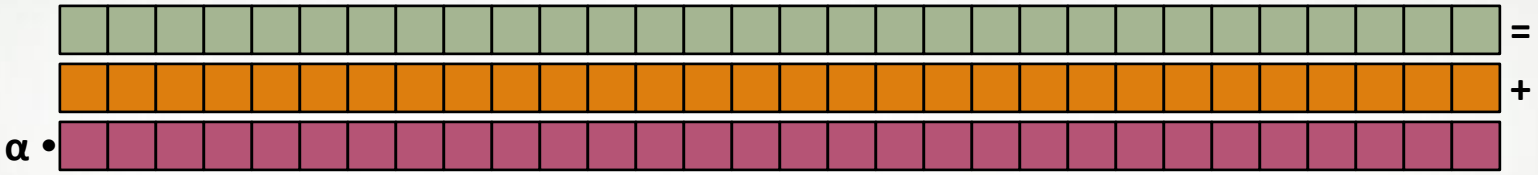
Cray Inc.

Outline

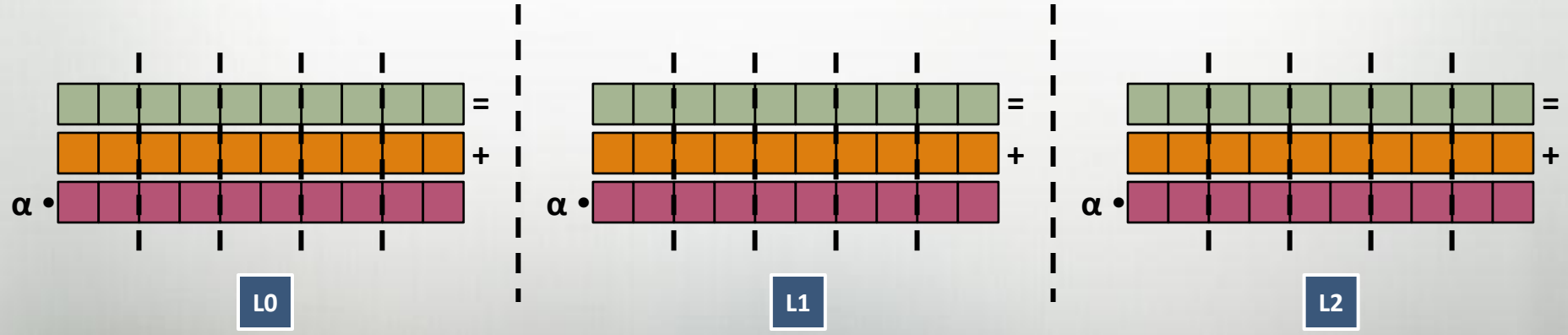
- Domain Maps
  - Layouts
  - Distributions
- Chapel Standard Layouts and Distributions
- User-defined Domain Maps

Domain Maps

Domain maps are a "recipe" that instructs the compiler how to map the global view…



…to memory and/or locales



More on Domain Maps

A domain map defines:

- Ownership of domain indices and array elements
- Underlying representation
- Standard set of operations on domains and arrays
  - E.g, slicing, reindexing, rank change
- How to farm out work
  - E.g., forall loops over distributed domains/arrays

Domain maps are built using language-level constructs

- No special compiler support

Using Domain Maps

- Syntax

```
dmap-type:
  dmap(dmap-class(...))
dmap-value:
  new dmap(new dmap-class(...))
```

- Semantics
  - Domain map classes are defined in Chapel
- Examples

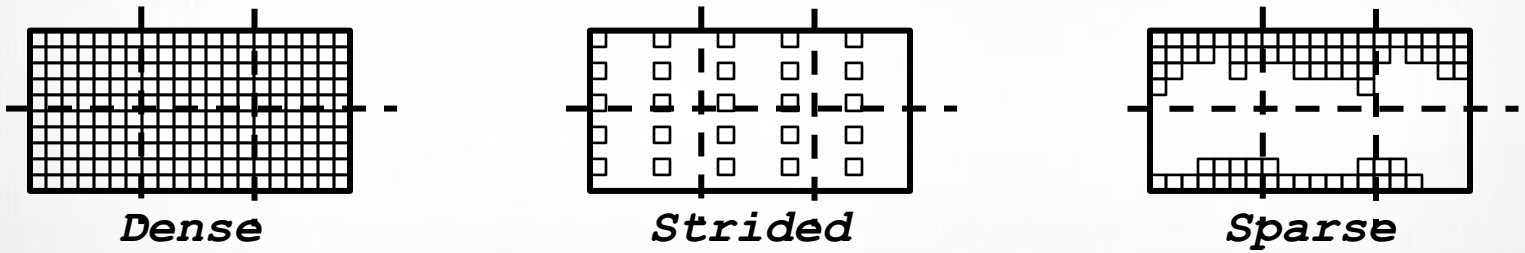
```
use myDMapMod;
var DMap: dmap(myDMap(...)) = new dmap(new myDMap(...));

var Dom: domain(...) dmapped DMap;
var A: [Dom] real;
```

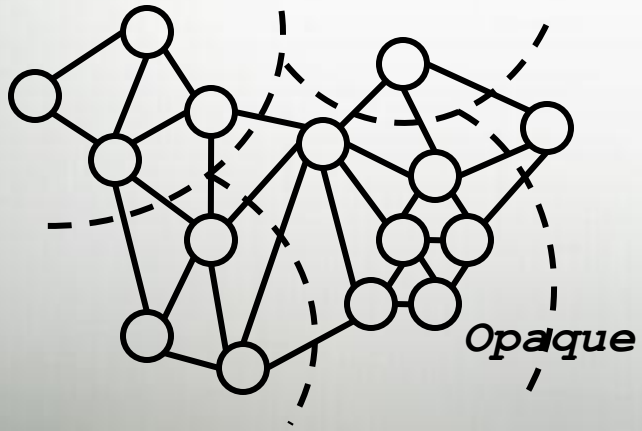
Dmapping Domains

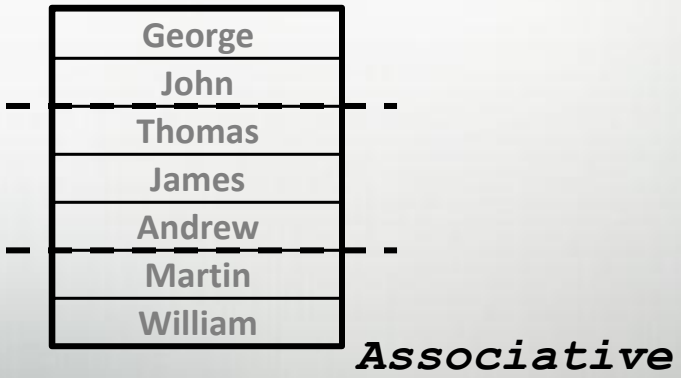
All domain types can be dmapped.

Semantics are independent of domain map.

(Though performance and parallelism will vary...)







Layouts

Layouts are single-locale domain maps

- Uses begin, cobegin, coforall to implement data parallelism
- May take advantage of locale resources, *e.g.*, multiple cores

Examples

- Sparse CSR
- GPU

Distributions

Distributions are multi-locale domain maps

- Uses begin, cobegin, coforall to implement data parallelism
- Uses on to control data and task locality
- May use layouts for per-locale implementation

Examples

- Block
- Cyclic
- Block-Cyclic
- Block CSR
- Recursive bisection

Outline

- Domain Maps
- Chapel Standard Layouts and Distributions
  - Block
  - Cyclic
- User-defined Domain Maps

Chapel Standard Layouts and Distributions

Chapel provides a number of standard layouts and distributions

- All are written in Chapel

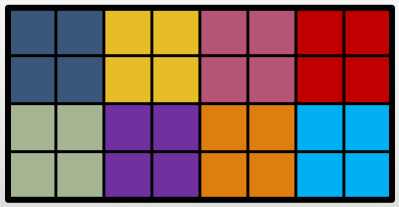
Examples

- Block distribution
- Cyclic distribution

The Block Distribution

The Block Distribution maps the indices of a domain in a dense fashion across the target Locales according to the `boundingBox` argument

```
const Dist = new dmap(new Block(boundingBox=[1..4, 1..8]));

var Dom: domain(2) dmapped Dist = [1..4, 1..8];
```



distributed over

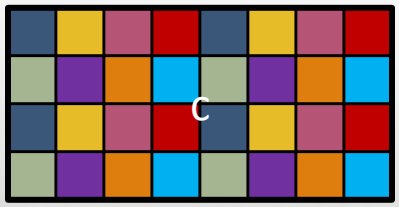
The Block class constructor

```
def Block(boundingBox: domain,
          targetLocales: [] locale = Locales,
          dataParTasksPerLocale = ...,
          dataParIgnoreRunningTasks = ...,
          dataParMinGranularity = ...,
          param rank = boundingBox.rank,
          type idxType = boundingBox.dim(1).eltType)
```

The Cyclic Distribution

The Cyclic Distribution maps the indices of a domain in a round-robin fashion across the target Locales according to the `startIdx` argument

```
const Dist = new dmap(new Cyclic(startIdx=(1,1)));

var Dom: domain(2) dmapped Dist = [1..4, 1..8];
```



distributed over

The Cyclic class constructor

```
def Cyclic(startIdx,
           targetLocales: [] locale = Locales,
           dataParTasksPerLocale = ...,
           dataParIgnoreRunningTasks = ...,
           dataParMinGranularity = ...,
           param rank: int = infered from startIdx,
           type idxType = infered from startIdx)
```

A little under the covers..

- Both the Block and Cyclic distributions use coforall and on to implement forall loops

```
coforall locDom in locDoms do on locDom {
  ... local portion ...
}
```

- Each locale's local portion uses the same knobs for intra-locale parallelism as default arrays and domains

Outline

- Domain Maps
- Chapel Standard Layouts and Distributions
- User-defined Domain Maps

User-defined Domain Maps

(Advanced) programmers can write domain maps

- The compiler uses a structural interface to build domain maps:
  - Create domains and arrays
  - Map indices to locales
  - Access array elements
  - Iterate over indices/elements sequentially, in parallel, zippered
  - …

Standard Domain Maps ***are*** user-defined domain maps

Design goal: User-defined domain maps should perform as well as the Chapel Standard Domain Maps

Future Directions

- More standard distributions and layouts
- Specify interface for user-defined domain maps

Questions?

- Domain maps
  - Layouts
  - Distributions
- The Chapel Standard Distributions
  - Block Distribution
  - Cyclic Distribution
- User-defined Domain Maps