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 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;
```
All domain types can be dmapped. Semantics are independent of domain map. (Though performance and parallelism will vary...)

Dense

Strided

Sparse

Opaque

Associative

Chapel: Distributions and Layouts
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 provides a number of standard layouts and distributions

- All are written in Chapel

Examples

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

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

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

distributed over

<table>
<thead>
<tr>
<th>L0</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4</td>
<td>L5</td>
<td>L6</td>
<td>L7</td>
</tr>
</tbody>
</table>
The Block class constructor

```chapel
def Block(boundingBox: domain,
        targetLocales: [] locale = Locales,
        dataParTasksPerLocale = ...,
        dataParIgnoreRunningTasks = ...,
        dataParMinGranularity = ...,
        param rank = boundingBox.rank,
        type idxType = boundingBox.dim(1).eltType)
```
The Cyclic Distribution maps the indices of a domain in a round-robin fashion across the target Locales according to the `startIdx` argument.

```chapel
const Dist = new dmap(new Cyclic(startIdx=(1,1)));
var Dom: domain(2) dmapped Dist = [1..4, 1..8];
```

distributed over

L0 L1 L2 L3
L4 L5 L6 L7
def Cyclic(startIdx,
    targetLocales: [] locale = Locales,
dataParTasksPerLocale = ...,
dataParIgnoreRunningTasks = ...,
dataParMinGranularity = ...,
    param rank: int = inferred from startIdx,
    type idxType = inferred from startIdx)
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