

# Titanium

*Titanium* is a research language from Berkeley

- Based on Java
  - ... but not JVM
  - Back-ends for SMP, MPI, and more
- SPMD (single program multiple data)
  - Like MPI, starts `main` in P processes
  - Like OpenMP, mutable values can be shared

<http://titanium.cs.berkeley.edu/>

# Hello World in Titanium

```
class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println("Process " + Ti.thisProc()  
                            + " of " + Ti.numProcs());  
    }  
}
```

# Barriers

```
class Barrier {  
    public static void main (String[] args) {  
        System.out.println("Start " + Ti.thisProc());  
        Ti.barrier();  
        System.out.println("Finish " + Ti.thisProc());  
    }  
}
```

# Broadcasting Values

```
class Bcast {  
    public static void main (String[] args) {  
        int v;  
        v = broadcast Ti.thisProc() from 0;  
        System.out.println("At " + Ti.thisProc() + ": " + v);  
    }  
}
```

# Broadcasting Mutable Values

```
class BcastMut {
  public static void main (String[] args) {
    int[] va;
    va = broadcast new int[1] from 0;
    System.out.println("Pre " + Ti.thisProc() + ": " + va[0]);
    if (Ti.thisProc() == 0)
      va[0] = 17;
    Ti.barrier();
    System.out.println("Post " + Ti.thisProc() + ": " + va[0]);
  }
}
```

# Cost of Non-Local Access

See `GlobArray`

# Multi-Dimensional Arrays

Type for an  $N$ -dimensional array of  $T$ :

```
T [Nd]
```

Example, a 2-D array of integers:

```
int [2d] m;
```

Use ranges to create an instance:

```
new T [lo1:hi1, lo2:hi2, ... loN:hiN]
```

Examples:

```
m = new int [0:9,0:19]; // a 10x20 array
```

```
int [1d] a = new int [50:99];
```

# Points

A *point* accesses a place in an array

- `Point<N>` is the type of a point in  $N$  dimensions
- `[x1, ... xN]` produces a `Point<N>`

Examples:

```
Point<2> a2 = [1,2];
```

```
Point<2> b2 = [8,1];
```

```
Point<5> p5 = [8,1,0,9,12];
```

Points of the same dimension can be added, etc.



# Domains

A **domain** is a region of an array

- `Domain<N>` is the type of a domain in  $N$  dimensions
- If  $p_1$  and  $p_2$  are `Point<N>`s, then `[p1 : p2]` produces a rectangular `Domain<N>`

Example:

```
Point<2> a2 = [1,2];  
Point<2> b2 = [8,10];  
Domain<2> d = [a2 : b2];
```

`foreach` iterates over points within a domain

# Copying Non-Local Arrays to Local

See `GlobArray2`