A high level SPMD programming model: HPspmd

and its Java language binding

Guansong Zhang, Bryan Carpenter, Geoffrey Fox
Xinying Li and Yuhong Wen

NPAC at Syracuse University

Observations

- Data parallel program and HP computing
- Distributed memory system: from multiprocessor system to NOW
- MPI: the address translation is error-prone.
- HPF attractive, but implementations problematic: especially programmer do not understand the reason for performance.

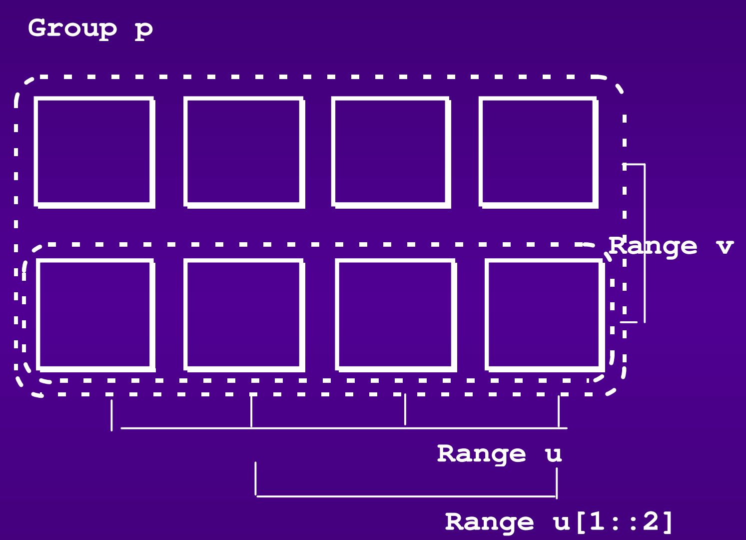
HPspmd programming model

- A new programming style:
  - Structured SPMD programming
  - Global name space
  - Hybrid of data and task parallel programming
  - Communication libraries
- Combined with Java

    -- simple, elegant, the language of the future ...

Structured SPMD programming

- **Group**: a grid structure and its dimensions
- **Range**: a grid dimension or integer interval mapped to a dimension with distribute patterns
- **Location**: an abstract element of Range, mapped by an integer.




```
Range u = p.dim(0);
Range v = p.dim(1);

Location j = v[1];
Group q = p / j;
```

Global name space

HPJava	HPF
Procs p = new Procs1(4); Range x = new BlockRange(100, p.dim(0));	!HPF$ PROCESSOR P(4) !HPF$ DISTRIBUTE T(BLOCK) ONTO P
float [[,*]] a = new float [[x, 100]] on p;	REAL A(100,100) !HPF$ ALIGN A(:,*) WITH T(:)
float [[]] b = new float [[x]] on p;	REAL B(100) !HPF$ ALIGN B(:) WITH T(:)

Hybrid of data and task parallel programming

- Explicitly coordinate groups of processors
- Groups cooperate in loosely synchronous manner
- Control constructs on, at and overall

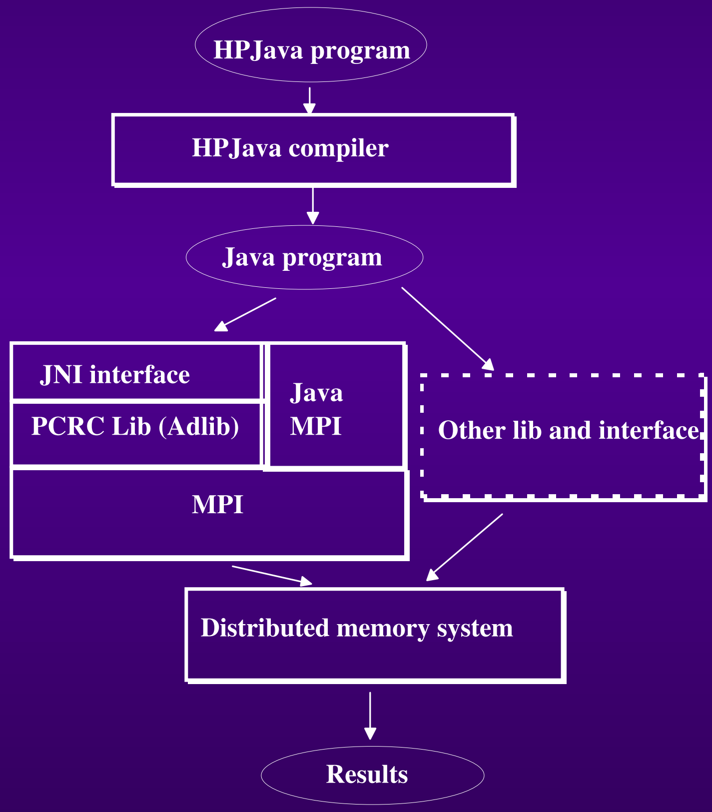
```
on(p) {
  overall(i=x[0:99])
    b[i]=0;

  at(j=x[3])
    b[j]=23;

  overall(i=x[2:99]) {
    for (k=0; k<50; k++)
      a[i, k]=b[i]+k;
    }
}
```

Communication libraries

- Language itself does NOT provide implicit data movement semantics.
- Different communication patterns are implemented as libraries.
- Benefits:
  - compiler writer: simple
  - programmer: exploiting locality
  - the language itself: extension (starting from MPI)

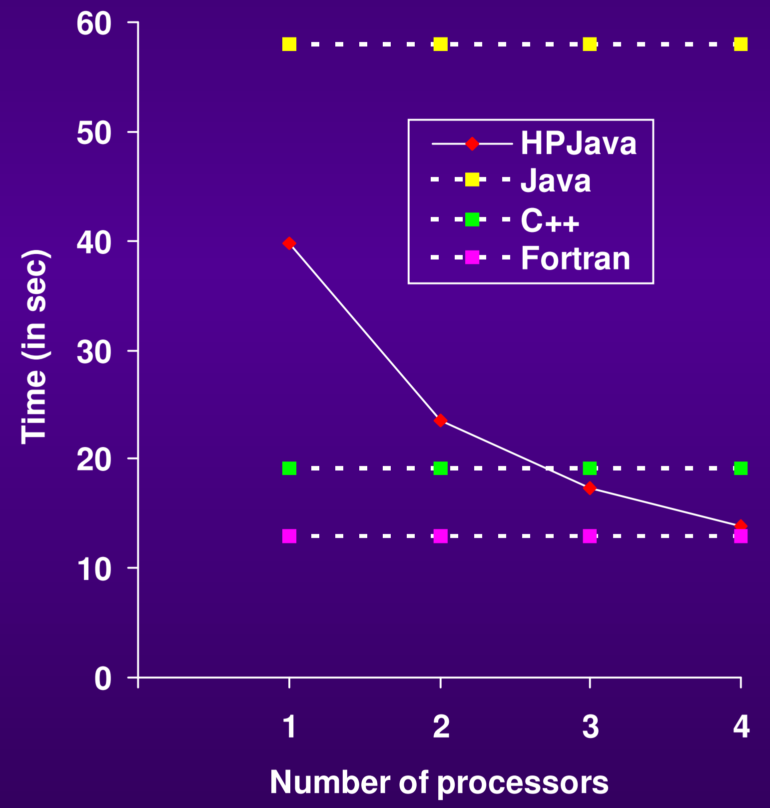
General approach

- HPJava compiler (source code translator)
- Java JNI interface
- PCRC run time library
- MPI library



Preliminary Performance

- Hand translated Jacobi iteration (HPJava to Java)
- Compared with sequential Java, C++ and Fortran (-O opt level)
- JDK 1.2Beta, JNI to Adlib, MPICH



Summary

- HPspmd style can be binded with other languages
  - Flexible as in SPMD programming
  - Convenient like HPF
  - Encourages expressing parallelism
  - Simplifies compiler
- More communication libraries can be integrated.

FAQ

- Why not extending Java by packages?
- Different from HPF?
- Distributed array plus Java array?
- ...

Extending Java

- An interface (Java package) is cumbersome to code with
  - a[i]=3
  - a.dat()[a.pos(i)]=3
- There is no performance for this kind of interface
  - a source code translator can implement address translation in a more efficient way

Different from HPF

- Explicit communication

```
A(10, 10) = B(30)
```

Adlib.remap(a[[9,9]], b[[29]]);

- Two kinds of array
  - new reference type implemented as Java class
  - array of distributed array

Data descriptor

- Distributed array
  - Data distribution group
  - Dimension information
  - Data reference
- Array section and type signature

  int #, int [[ ]], int [[,]], int[[*,]], int [[,*]] ...