

# 计算科学与工程中的 并行编程技术

## **Parallel Programming Technology in Computational Science and Engineering**

都志辉

清华大学计算机系

Email : `duzh@tsinghua.edu.cn`

Phone: 62782530

<http://hpclab.cs.tsinghua.edu.cn/~duzh>

# 数据并行语言HPF

# 什么是数据并行与任务并行？

- 数据并行(SIMD)
  - 数据作为主要的处理对象
  - 相同的操作，不同的数据 肯定不能有同步操作
- 任务并行
  - 不同的任务作为主要的处理对象
  - 不同的操作，相同或者不同的数据

# 为什么数据并行问题是重要的？

- 科学与工程问题的划分（Geoffrey C.Fox）
  - 同步问题 40%      松同步问题36%
  - 无同步问题 7%      异步问题17%
- 数据并行问题 异步操作太少了
  - 同步问题 40%+松同步问题36%+无同步问题 7%=83%
- 数据并行语言的意义：有效、简洁地表达数据并行问题

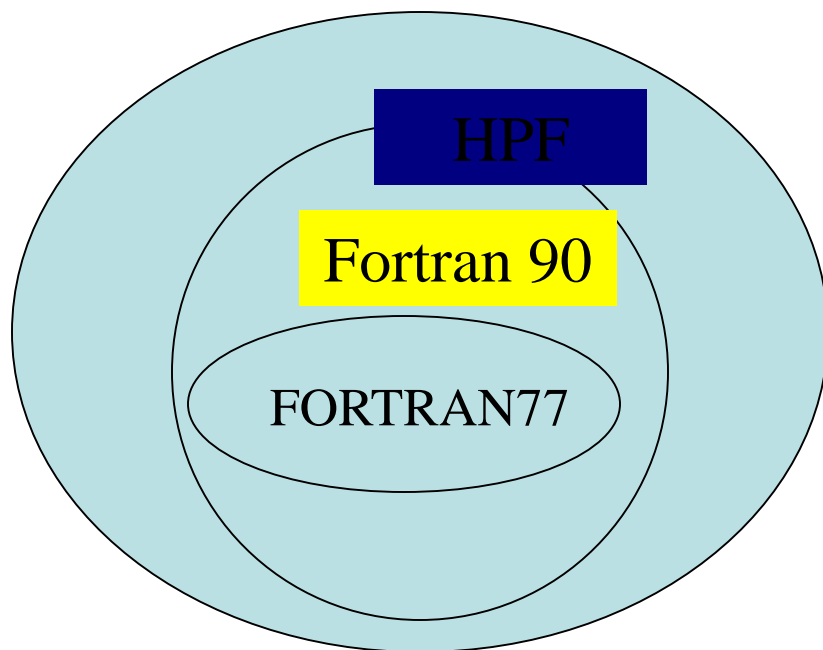
# 什么是HPF (High Performance Fortran)

- 一种数据并行语言
- 数据并行语言的工业标准

# HPF的产生

- 并行编程难（对高级并行语言的需求）
- 可行（早期数据并行语言的成功实践）
- 普遍（数据并行问题的普遍性）
- 容易（数据并行较任务划分并行更容易实现）

# HPF与其前面语言的关系



结论：所有FORTRAN77/Fortran90程序都可以作为特殊的HPF程序

HPF相当于是F90的扩展

# F90 对F77的扩充

- 数组运算

$A=B$

$A(20:1000)=B(10:990)+30$  vec + scalar

$C=A+C+D(:,1)$  切片



# HPF对F90的扩充

- 标注语句
  - 数据分配
    - !HPF\$ DISTRIBUTE A ONTO P
  - 执行语句
    - !HPF\$ INDEPENDENT DO
- 新的并行执行结构
  - FORALL(I=1:M,J=1:N) A(I,J)=B(I,J)

# HPF中体现并行的几个方面

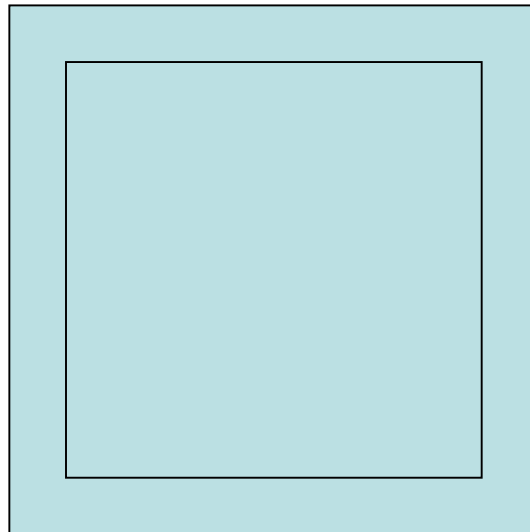
- 数据分布与映射
- 并行控制结构
- 内部函数与标准库
  - $\text{Sin}(A)+\text{Sin}(B)$

# HPF的数据对齐

为什么要数据对齐？(有效的数据分配)

A(16,16) B(14,14)

!HPF\$ ALIGN B(I,J) WITH A(I+1,J+1)



# HPF的并行执行结构FORALL

FORALL(I=1:M,J=1:N) A(I,J)=B(I,J)

FORALL(I=1:M-1,J=1:N) A(I+1,J)=A(I,J)

使用旧值

FORALL(I=1:M-1,J=1:N)

A(I+1,J)=A(I,J)

B(I,J)= A(I,J)

END FORALL

后面的语句依赖前面的语句

# HPF的并行执行结构

## INDEPENDENT DO

```
!HPF$ INDEPENDENT
```

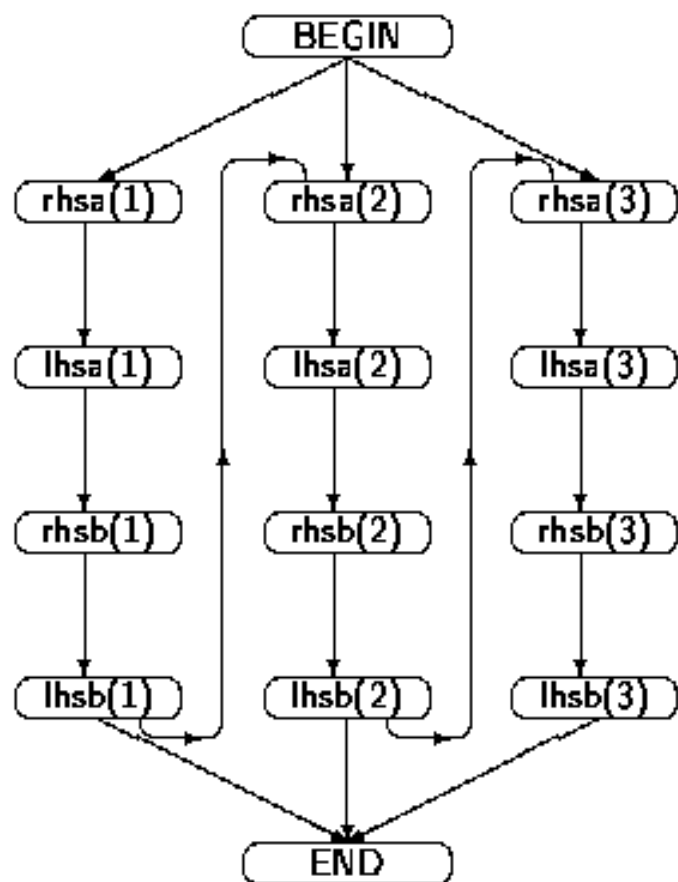
```
DO I=2,99
```

```
    A(I)=B(I-1)+B(I)+B(I+1)
```

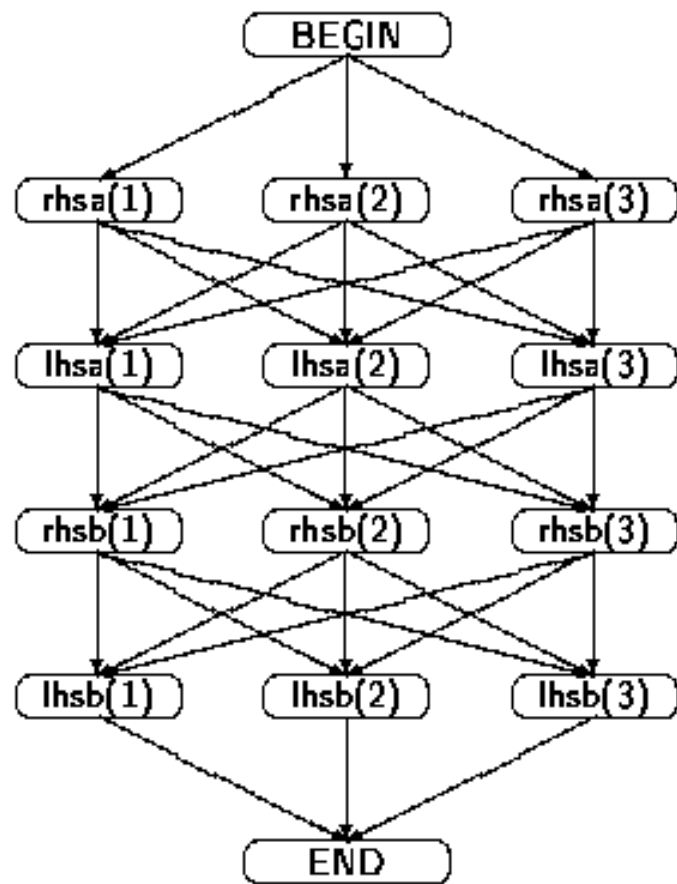
```
END DO
```

# 图示

```
DO i = 1, 3
  lhsa(i) = rhsa(i)
  lhsb(i) = rhsb(i)
END DO
```



```
FORALL ( i = 1:3 )
  lhsa(i) = rhsa(i)
  lhsb(i) = rhsb(i)
END FORALL
```



# 图示

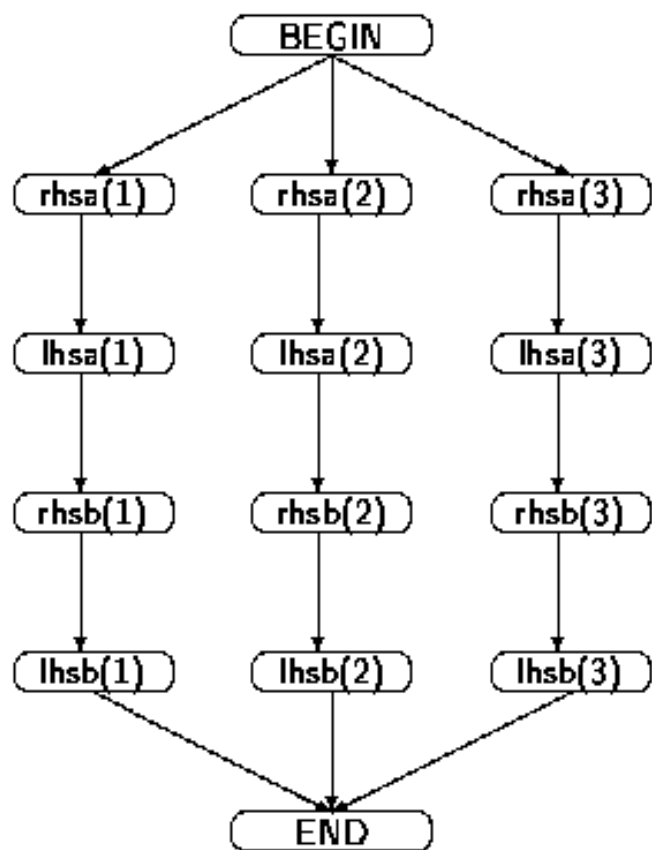
!HPF\$ INDEPENDENT

DO i = 1, 3

  lhsa(i) = rhsa(i)

  lhsb(i) = rhsb(i)

END DO



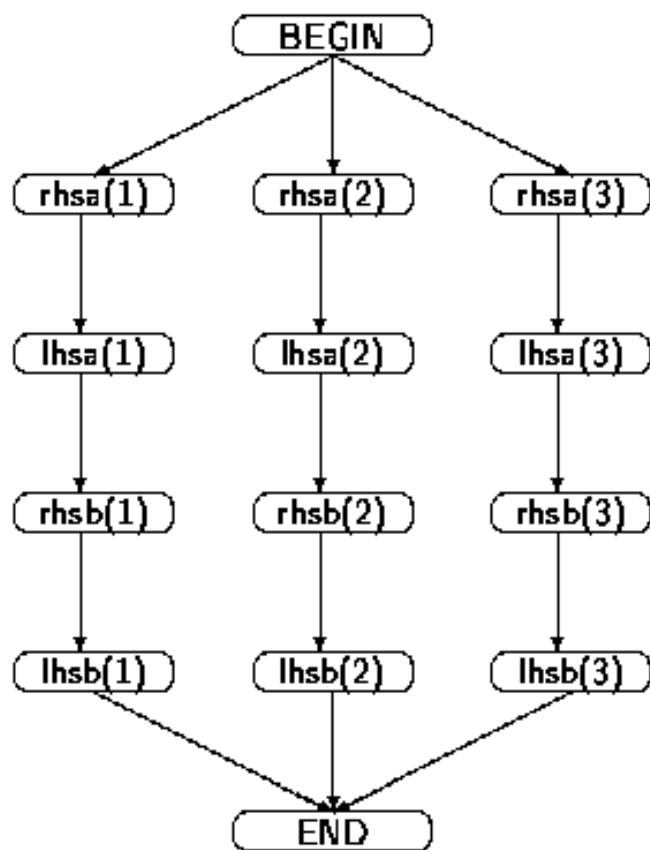
!HPF\$ INDEPENDENT

FORALL ( i = 1:3 )

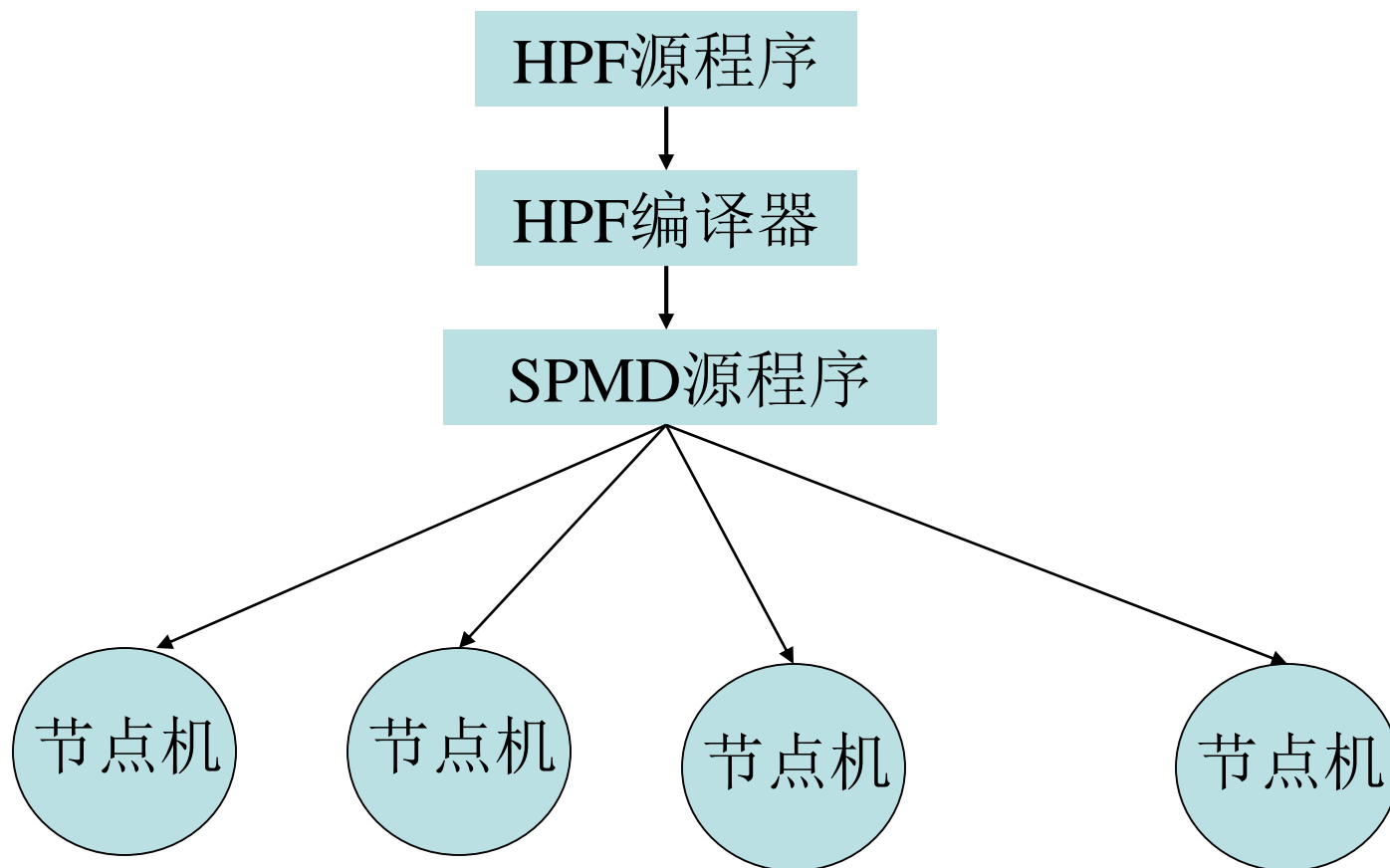
  lhsa(i) = rhsa(i)

  lhsb(i) = rhsb(i)

END FORALL



# HPF编译器的概念模型





# HPF编译器的主要功能

- 构造中间表示
- 转换数据映射为数据划分
- 并行结构的标量化结点程序映射
- 串行结构向结点程序的映射
- 对消息通信的处理及对运行支持系统的调用

# 已有的HPF编译器

- pghpf: Portland group Inc.
- XL HPF: IBM
- shpf: Southampton
- pkhpf: Peking University

# HPF编译器的主要组成

- 编译时分析
- 运行时系统

# 总结HPF的问题

- 表达能力
  - 表达复杂问题的能力有限
  - 表达复杂并行结构与程序结构的能力
- 高效编译器的性能
  - 完全的HPF编译器很难开发
  - 如何提高性能

# 与其它具有相同功能的 并行编程语言的对比

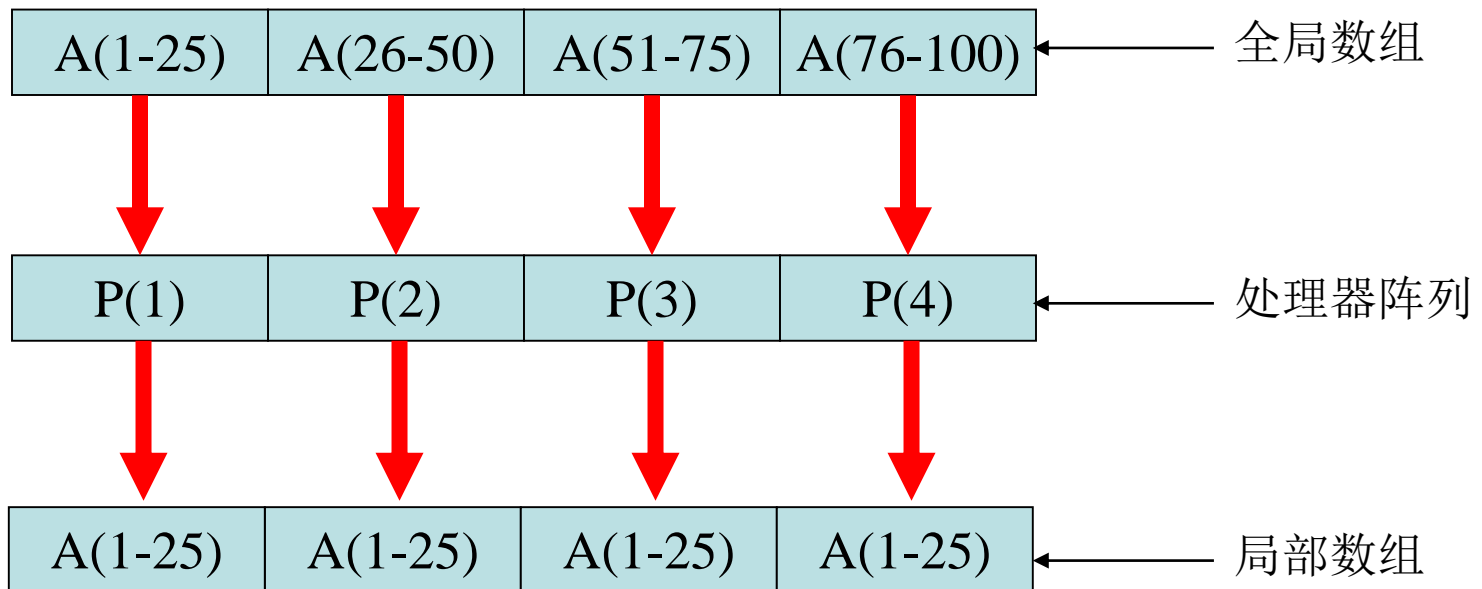
- 为什么使用消息传递语言
  - 代码高效
  - 灵活
- 为什么不使用消息传递语言
  - 难，编程级别低
- 为什么使用数据并行语言
  - 容易，快，编程级别高
- 为什么不使用数据并行语言
  - 没有高效、完全的编译器
  - 对混合、复杂问题的表达能力差

# HPF的精华：数据分布

- $CD(J,K)=(J+K-1)/K$
- $CR(J,K)=J-K*CD(J,K)$
- 设数组大小为d,处理器个数为p

# BLOCK

- 数组大小 $A(d)$ , 处理器个数 $p$ , 块大小 $NB=CD(d,p)$



# 例子

```
!HPF$ PROCESSORS SEDECIM(16)  
      REAL CENTURY(100)
```

Distributing the array BLOCK (which in this case would mean the same as BLOCK(7)):

```
!HPF$ DISTRIBUTE CENTURY(BLOCK) ONTO SEDECIM
```

results in this mapping of array elements onto abstract processors:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	8	15	22	29	36	43	50	57	64	71	78	85	92	99	
2	2	9	16	23	30	37	44	51	58	65	72	79	86	93	100	
3	3	10	17	24	31	38	45	52	59	66	73	80	87	94		
4	4	11	18	25	32	39	46	53	60	67	74	81	88	95		
5	5	12	19	26	33	40	47	54	61	68	75	82	89	96		
6	6	13	20	27	34	41	48	55	62	69	76	83	90	97		
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98		



# BLOCK(m)

- $m \geq \text{CD}(d, p)$

# 例子

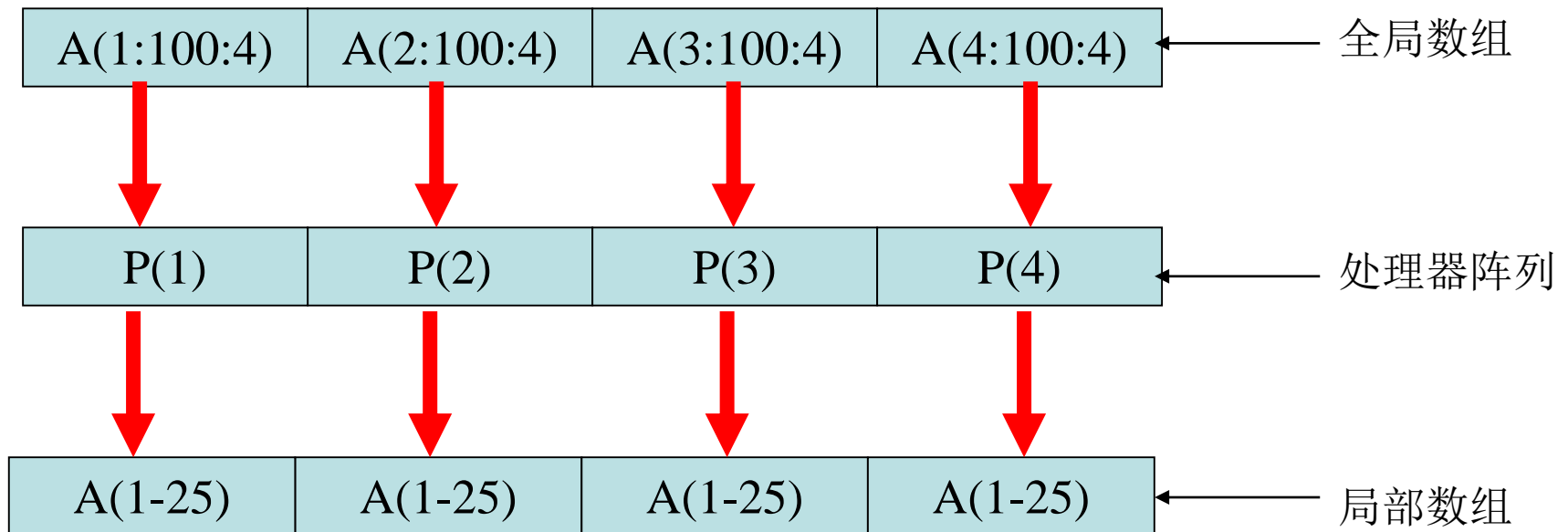
`!HPF$ DISTRIBUTE CENTURY(BLOCK(8)) ONTO SEDECIM`

results in this mapping of array elements onto abstract processors:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	9	17	25	33	41	49	57	65	73	81	89	97			
2	10	18	26	34	42	50	58	66	74	82	90	98			
3	11	19	27	35	43	51	59	67	75	83	91	99			
4	12	20	28	36	44	52	60	68	76	84	92	100			
5	13	21	29	37	45	53	61	69	77	85	93				
6	14	22	30	38	46	54	62	70	78	86	94				
7	15	23	31	39	47	55	63	71	79	87	95				
8	16	24	32	40	48	56	64	72	80	88	96				

# CYCLIC

- 数组大小 $A(d)$ , 处理器个数 $p$ , 块大小 $NB=CD(d,p)$



# 例子

!HPF\$ DISTRIBUTE CENTURY(CYCLIC) ONTO SEDECIM

results in this mapping of array elements onto abstract processors:

[illegible]

CYCLIC(m)

# 例子

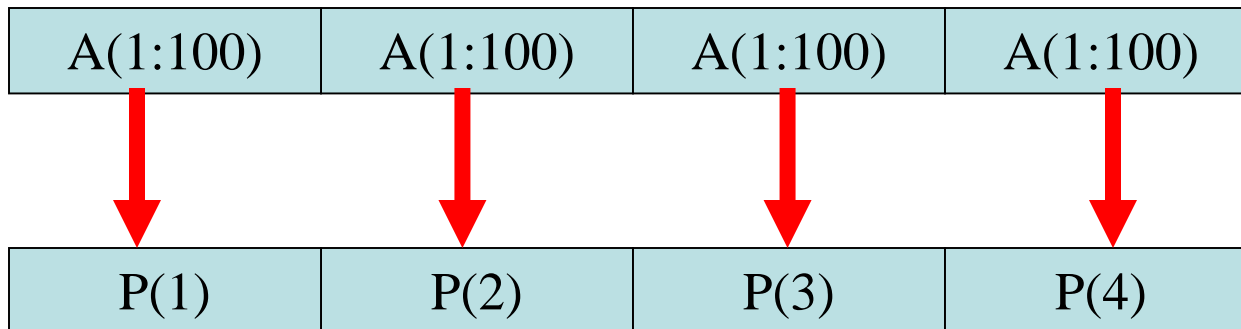
```
!HPF$ DISTRIBUTE CENTURY(CYCLIC(3)) ONTO SEDECIM
```

results in this mapping of array elements onto abstract processors:

[illegible]

(\*)

- 数组大小 $A(d)$ , 处理器个数 $p$ , 块大小 $NB=d$

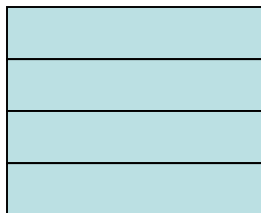


# HPF数据分布形式的组合

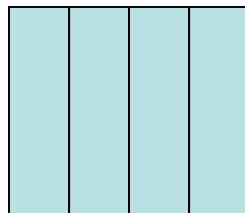
- 两维
  - (block,\*) (cyclic,\*) (block,cyclic)
  - (\*,block) (\*,cyclic) (cyclic,block)
  - (block,block) (cyclic,cyclic)
- 三维



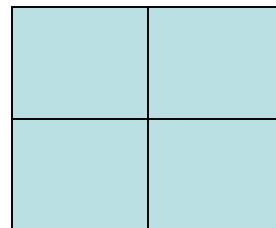
# 例子



(block,\*)



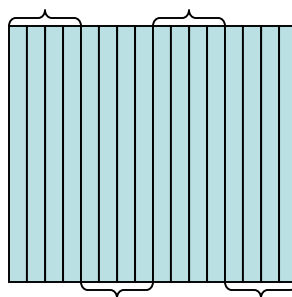
(\*,block)



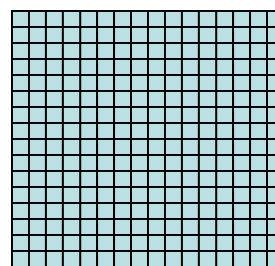
(block,block)



(cyclic,\*)



(\*,cyclic)



(cyclic,cyclic)

# 练习

- 已知数组 $A[1:N]$
- 将该数组分别按照 $BLOCK(M)$ ,  $CYCLIC(M)$ , \* 三种形式分布到 $P$ 个处理器上
- 求(从1开始编号)
- (1) 全局元素 $A(J)$ 被分配到的处理器编号 $I=?$   
(1- $P$ )
- (2) 全局元素 $A(J)$ 在其被分配的处理器 $P(I)$ 上对应的局部元素的下标 $A(Ij)$ 的 $Ij=?$  (1-?)
- (3) 第 $i$ 个处理器上分配到的元素的个数 $Ni=?$