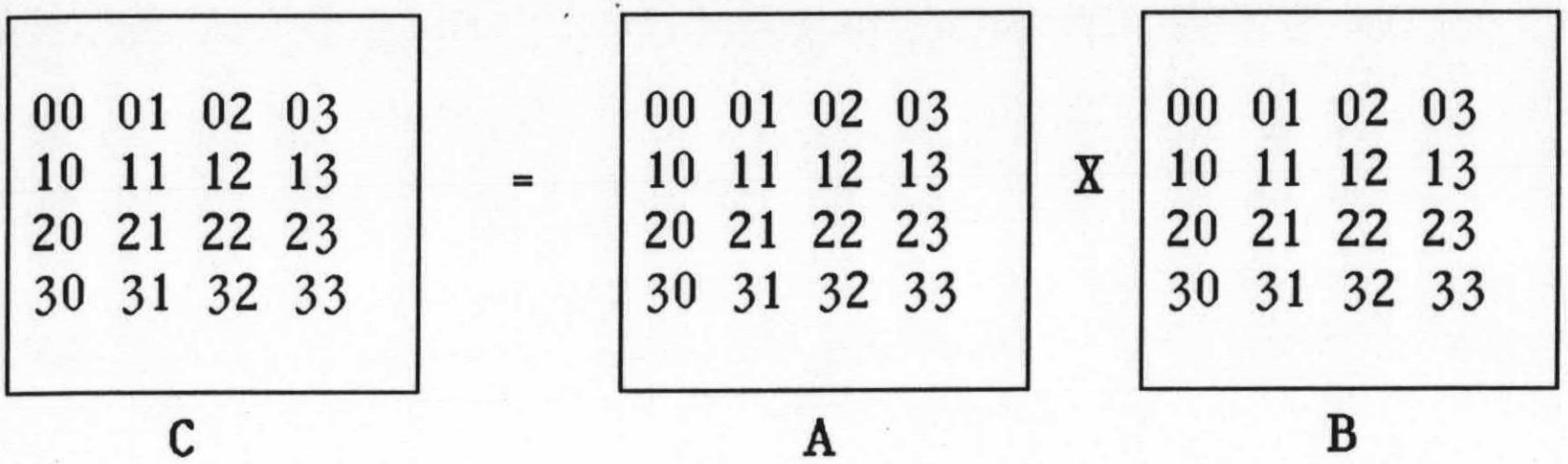
Matrix Multiplication on the Connection Machine

Matrix Multiply: $c_{ij} = \sum_{k-1}^{n} a_{ik} b_{kj}$

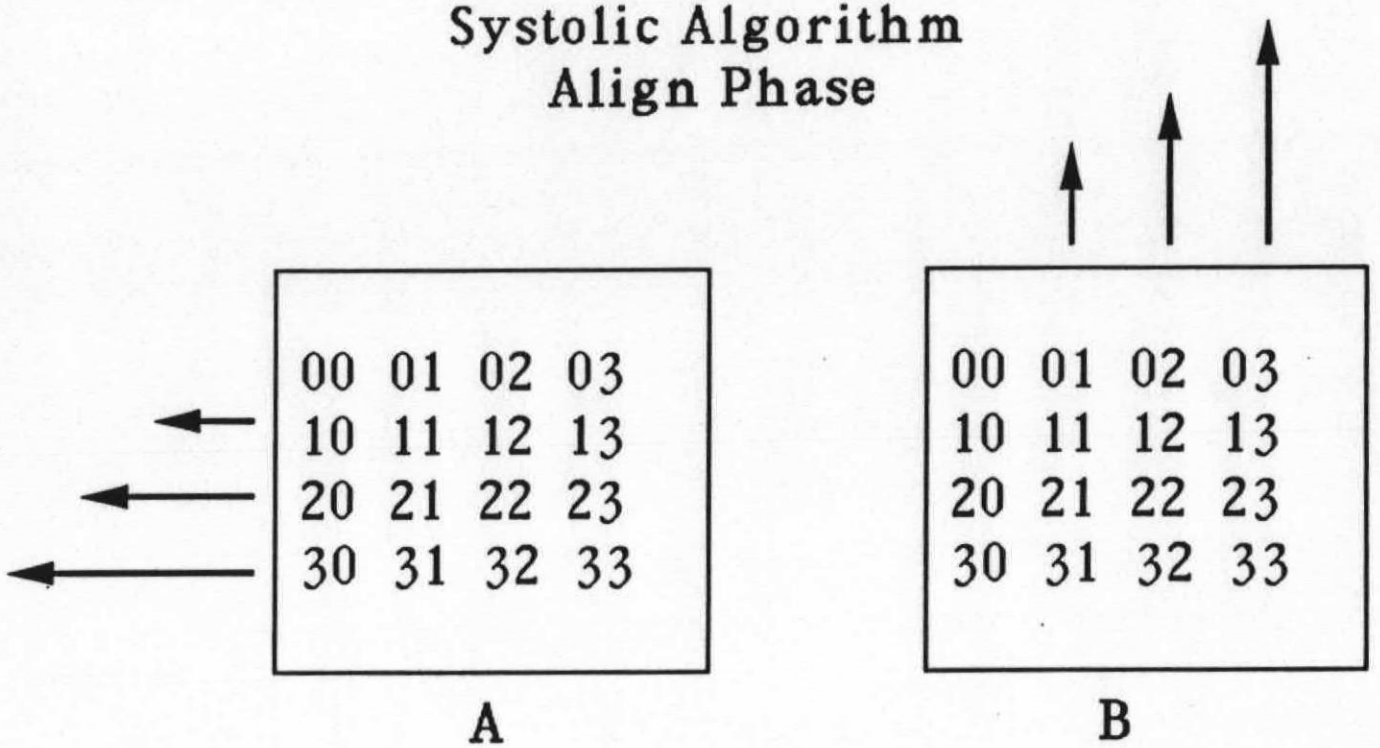


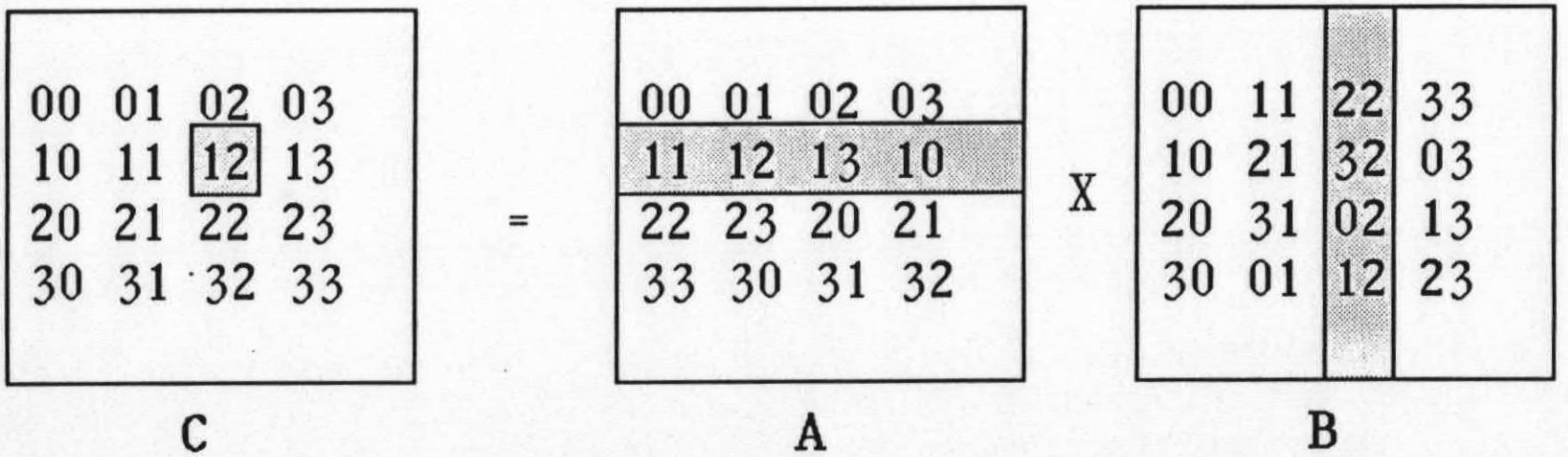
- Systolic "Cannon's" Algorithm
  - align and rotate phases

- "Blocked" Data Movement
  - minimizes communication costs

- Optimum use of Floating Point Units
  - micro-coded local matrix multiply

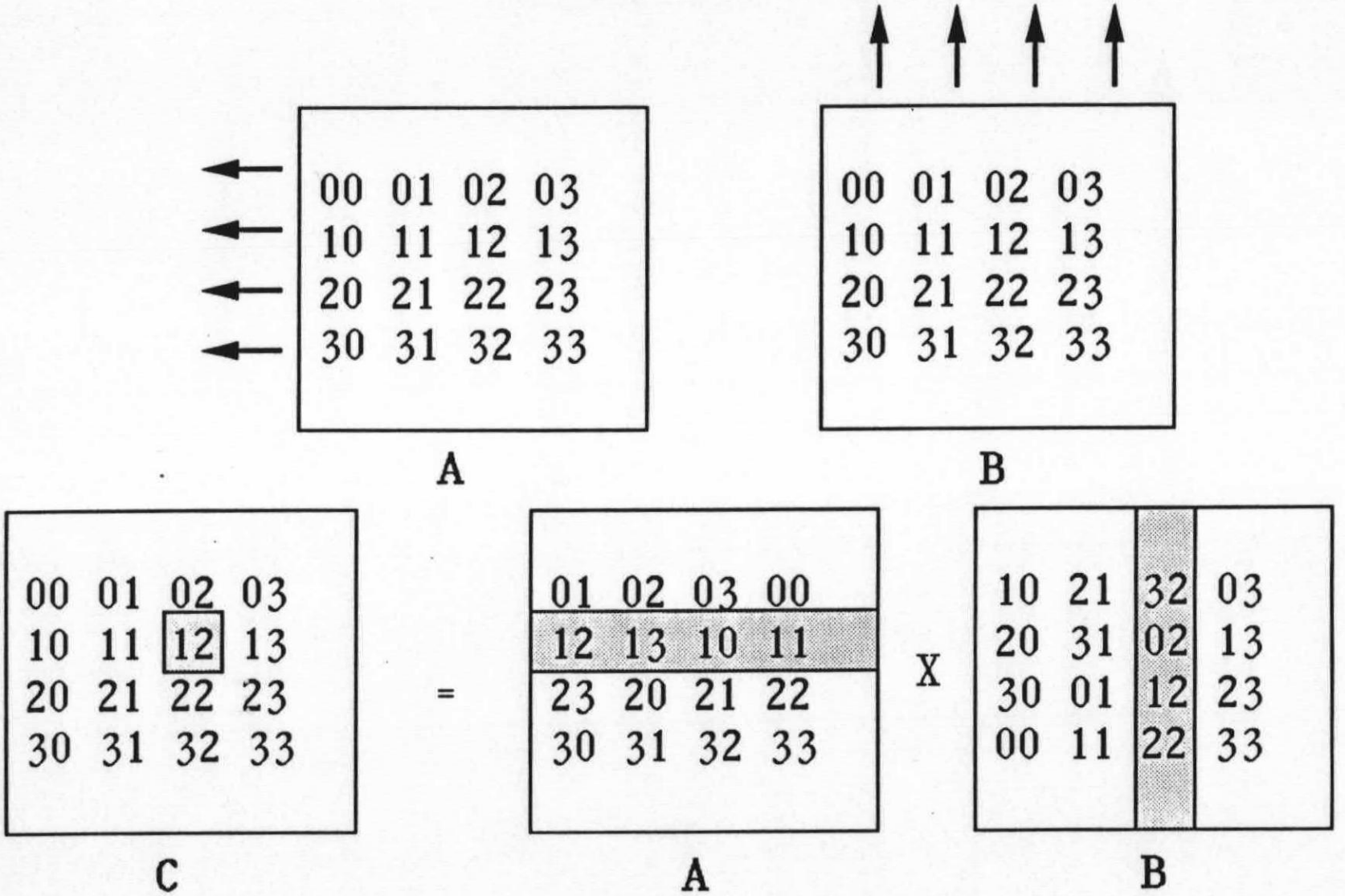
Systolic Algorithm
Align Phase





- In A row i moves i to left - In B column j moves j up
- Aligns rows of A and columns of B so inner indices match

Systolic Algorithm
Rotate Phase



- In A all rows move 1 to left - In B all columns move 1 up
- Multiply, add and rotate n times

Systolic - Blocked Algorithm
Conforming blocks

n X m		m X p		n X p
00 01 02 03	04 05 06 07	00 01	02 03	00 01	02 03
10 11 12 13	14 15 16 17	10 11	12 13	10 11	12 13
20 21 22 23	24 25 26 27	20 21	22 23	20 21	22 23
30 31 32 33	34 35 36 37	30 31	32 33	30 31	32 33
		40 41	42 43		
		50 51	52 53		
		60 61	62 63		
		70 71	72 73		

X =

00 01 02 03	X	00 01	=	00 01
10 11 12 13		10 11		10 11
		20 21		
		30 31		

- Perform local matrix multiply on entire block
- Local matrix multiply uses micro-coded SAXPY

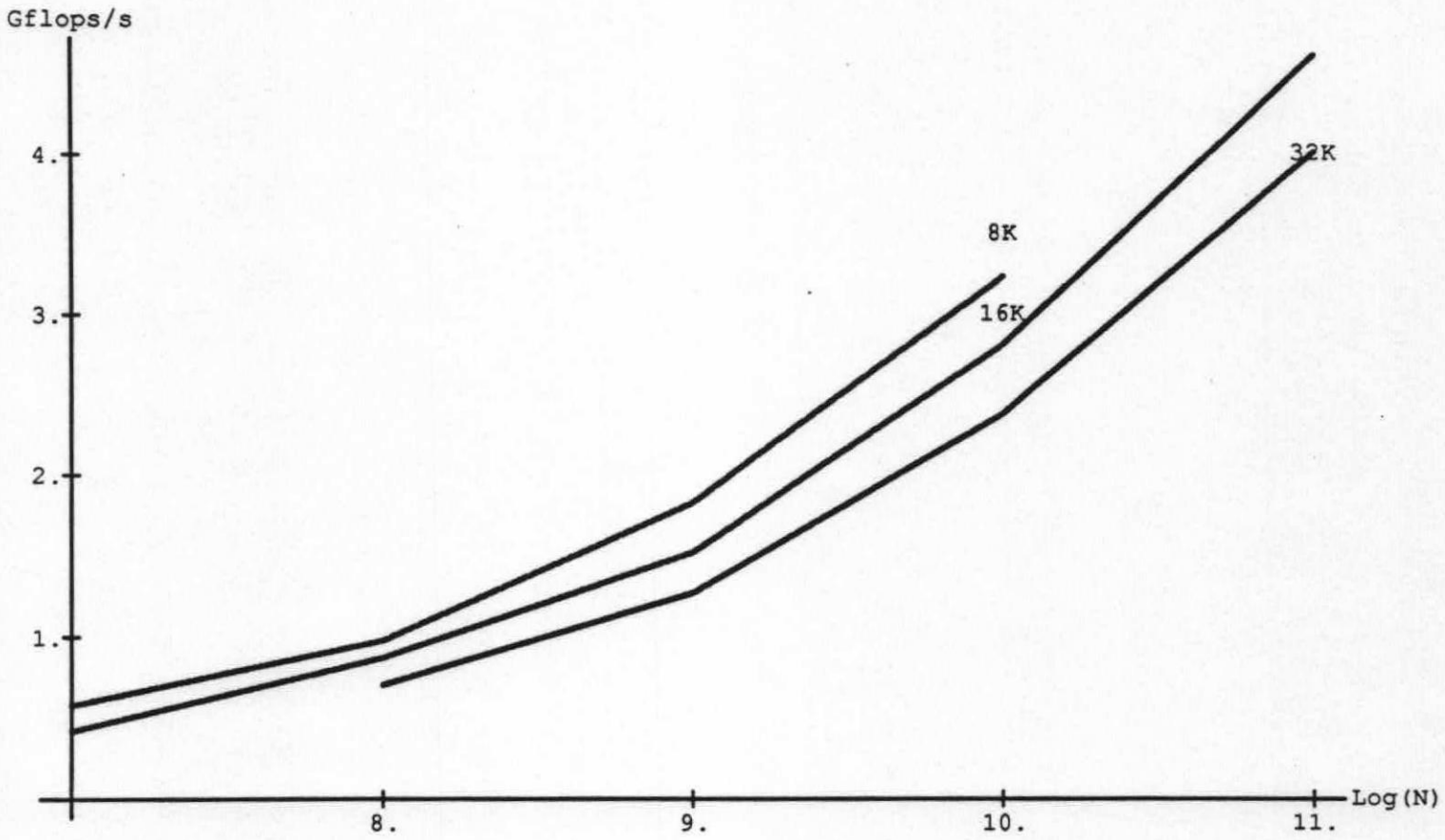
Systolic - Blocked Algorithm
Non-conforming blocks

n X m		m X p		n X p
00 01 02 03 04 05 06 07 10 11 12 13 14 15 16 17 20 21 22 23 24 25 26 27 30 31 32 33 34 35 36 37	X	00 01 02 03 10 11 12 13 20 21 22 23 30 31 32 33 40 41 42 43 50 51 52 53 60 61 62 63 70 71 72 73	=	00 01 02 03 10 11 12 13 20 21 22 23 30 31 32 33

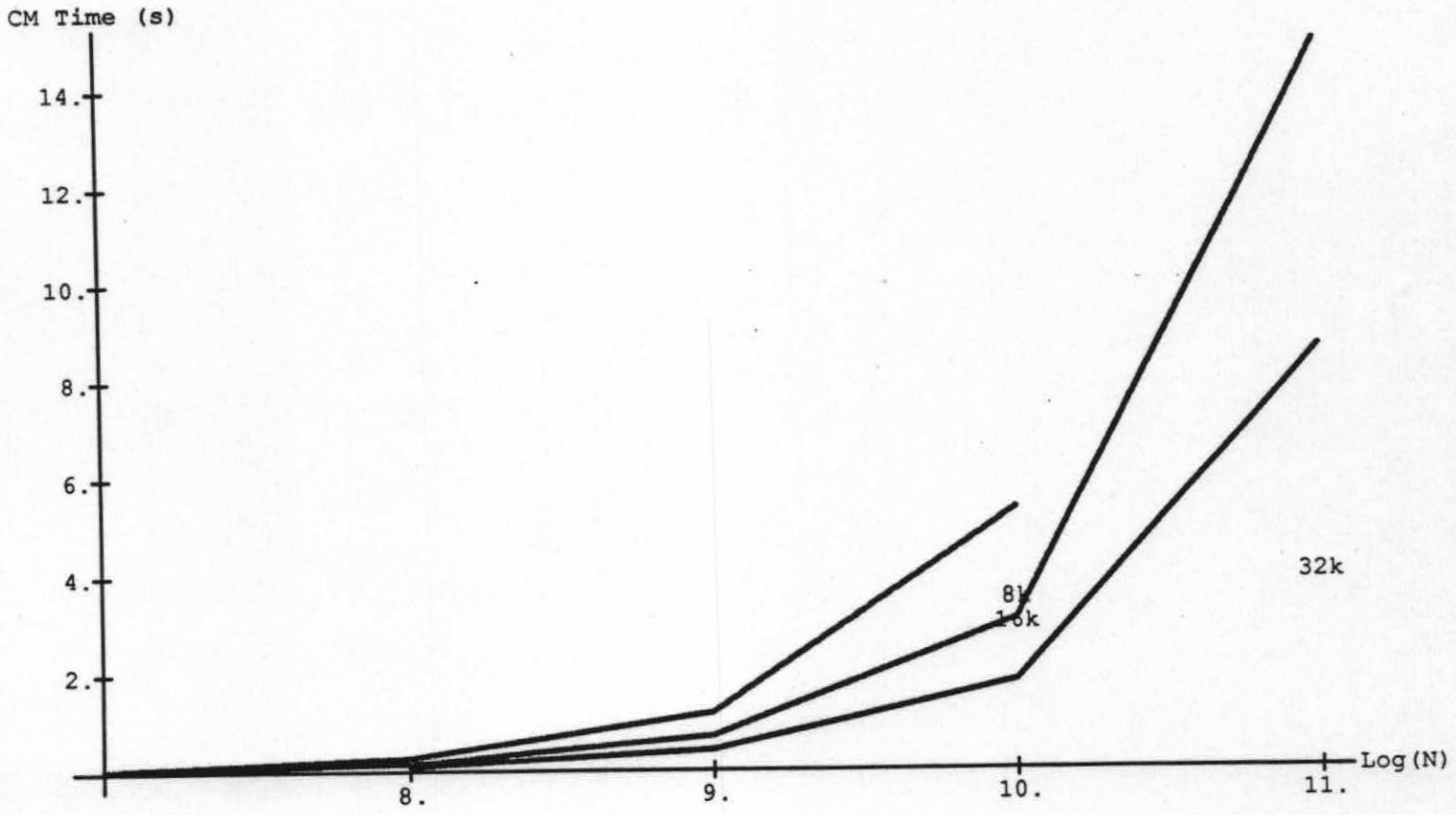
00 01 02 03 04 05 06 07	X	00 01 02 03 10 11 12 13	=	00 01 10 11

- Operate only on conforming part of blocks
- Rotate A and B unequally to allow use of all data

Performance of the matrix multiplication routine for square matrices of size N as a function of the size of the Connection Machine.



Performance of the matrix multiplication routine for square matrices of size N as a function of the size of the Connection Machine.



Peak Performance of the sprint node matrix multiplication kernel.