Computer Architectures for Computational Physics

work done by

Computational Research and Technology Branch

and

Advanced Computational Concepts Group

Ames Research Center

The following slides describe the importance of having high performance number crunching and graphics capability. They also indicate the types of research and development underway at Ames Research Center to ensure that, in the near-term, Ames is a smart buyer and user, and in the long-term we know what the best possible solutions are for our number crunching and graphics needs.

The drivers for this research are real computational physics applications of interest to Ames and NASA. We are concerned with how to map the applications, how to develop the optimal system software and system architecture, and how to maximize the physics learned from the results of the calculations (which at the present time means graphics). We are utilizing a group of DEC and CRAY manufactured MIMD architectures, various simulation tools for larger MIMD architectures, and also plan to utilize various versions of the Hypercube architecture. To control flow we are looking at simulations and prototypes for the study of data flow and systolic architectures. At present, it is a competition between the three architectures to determine which one will hold the most promise for the early 1990s. Once we have discovered which one (or two) hold the promise we will concentrate our computer science R&D in that area.

The computer graphics R&D activities are directed at getting maximum information from our three-dimensional calculations by utilizing the real time manipulation of three-dimensional data on the Silicon Graphics IRIS Workstation. We are also working on new algorithms which will permit the display of experimental results, which are sparse and random, the same way we display computed results, which are dense and regular. This would permit the synergistic coupling of computational and experimental techniques.

Computer Architectures for Computational Physics

by

Computational Research and Technology

and

Advanced Computational Concepts

presented by

K. G. Stevens, Jr.

Ames Research Center

Related Research and Development

More than 50 academic projects to build prototype systems

Many start—up and established companies developing SIMD, MIMD, and Systolic architectures

Several Government Agencies including DARPA, DoE, and NSA are into architecture studies

Rapid growth in computer graphics hardware by start—ups and established companies

How the Research at Ames is Different

Directed towards the computational physics applications of interest to NASA and Ames

Total system approach including hardware, software, applications, peripherals, and the user interface

Complete application programs are the target

Existing, Emerging, and Future designs are studied

OBJECTIVE

Conduct Research Which Will Have Benefit to Computational and Experimental Physics Research

- Computer Architecture
  - Short—term: How do we use what we have and what should we buy?
  - Long—term: What are the best architectures possible?
- Computer Graphics
  - Develop new algorithms and software to exploit computer graphics for experimental and computational physics

Technical Approach

Start with "real" complete applications

Map them onto architectures of interest

Predict performance via analysis, simulation, emulation and/or execution

Compare with other architectures and consider performance improving modifications

Determine the user interface implications —— programming languages, debuggers, environments, graphics packages, etc.

Areas of emphasis

- Architectures for "Number Crunching"
  - SIMD
  - MIMD
  - Data Flow
  - Systolic Arrays
- Computer Graphics

Algorithms of Interest

TWING

Conservative Full Potential Equation

(Implicit, Approximate Factorization Algorithm)

AIR3D

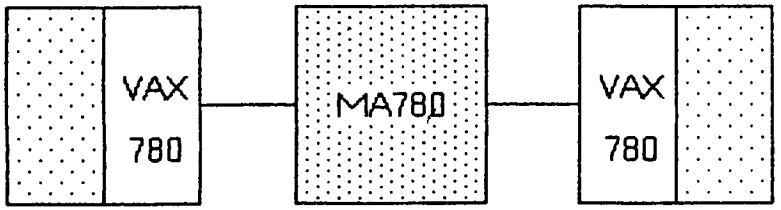
Reynolds—Averaged Navier Stokes

(Implicit, Approximate Factorization Algorithm)

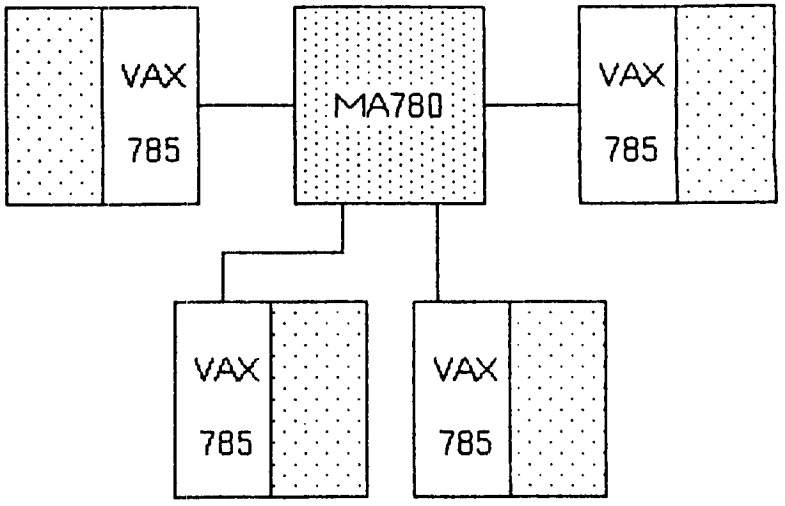
LES

Large Eddy Simulation Utilizing Spectral Methods

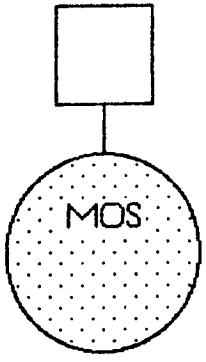
Architectures



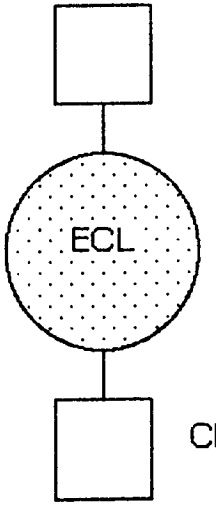
Dual VAX 11/780



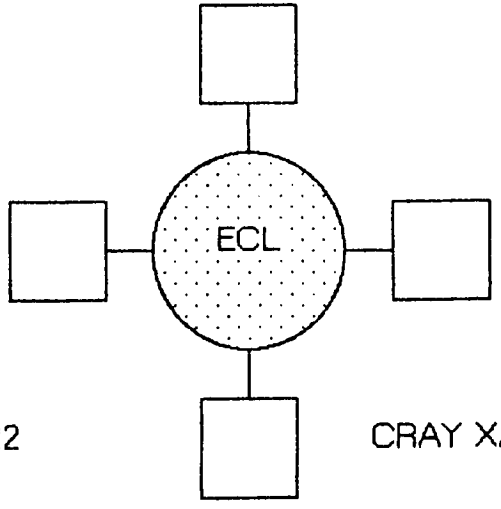
Quad VAX 11/785



CRAY X/MP-12



CRAY X/MP-22



CRAY X/MP-48

Performance of Multitasking on the CRAY X/MP

LES with 100 iterations on a 32^3 Mesh

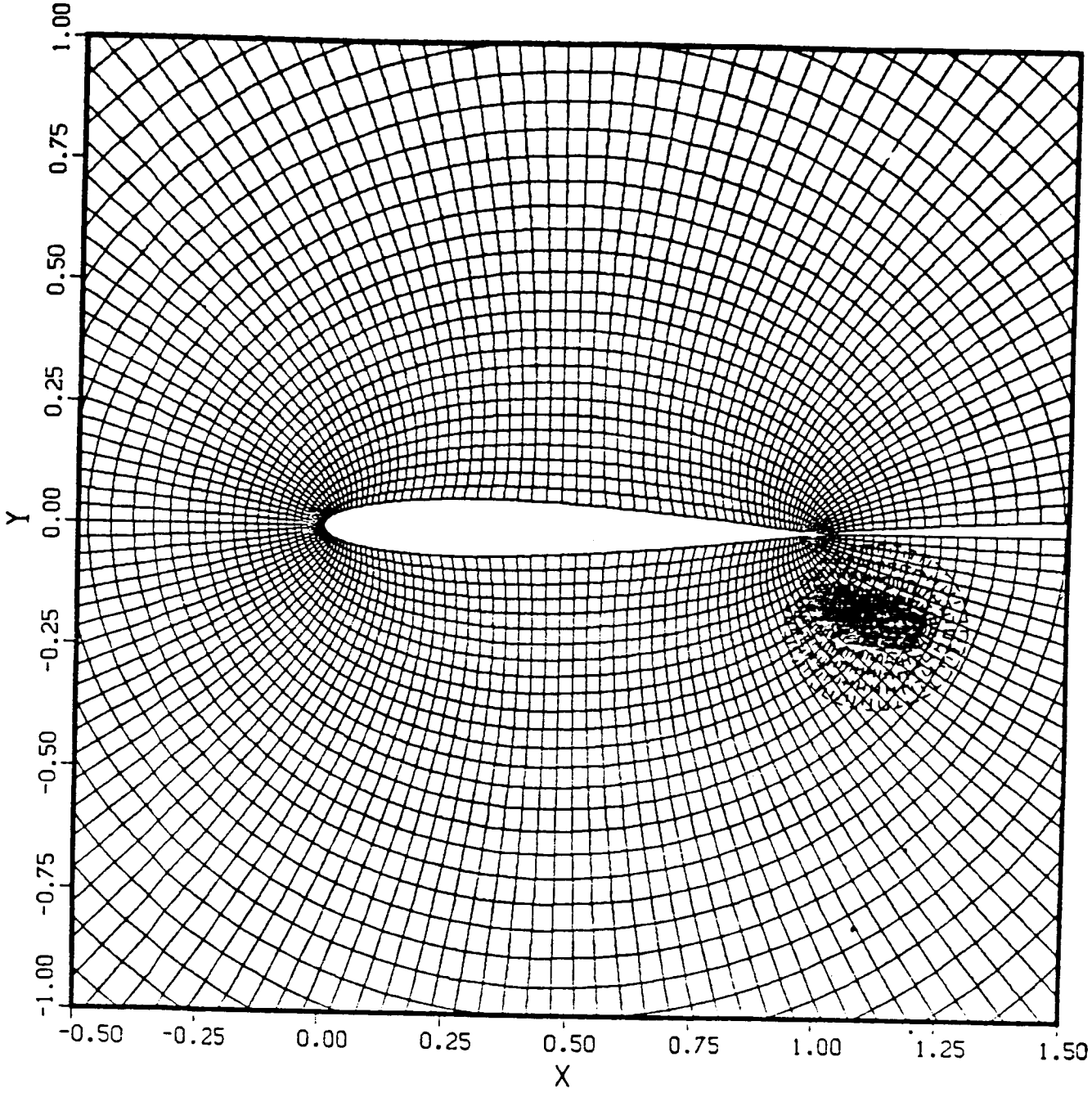
CPU Times in Seconds/Speed up

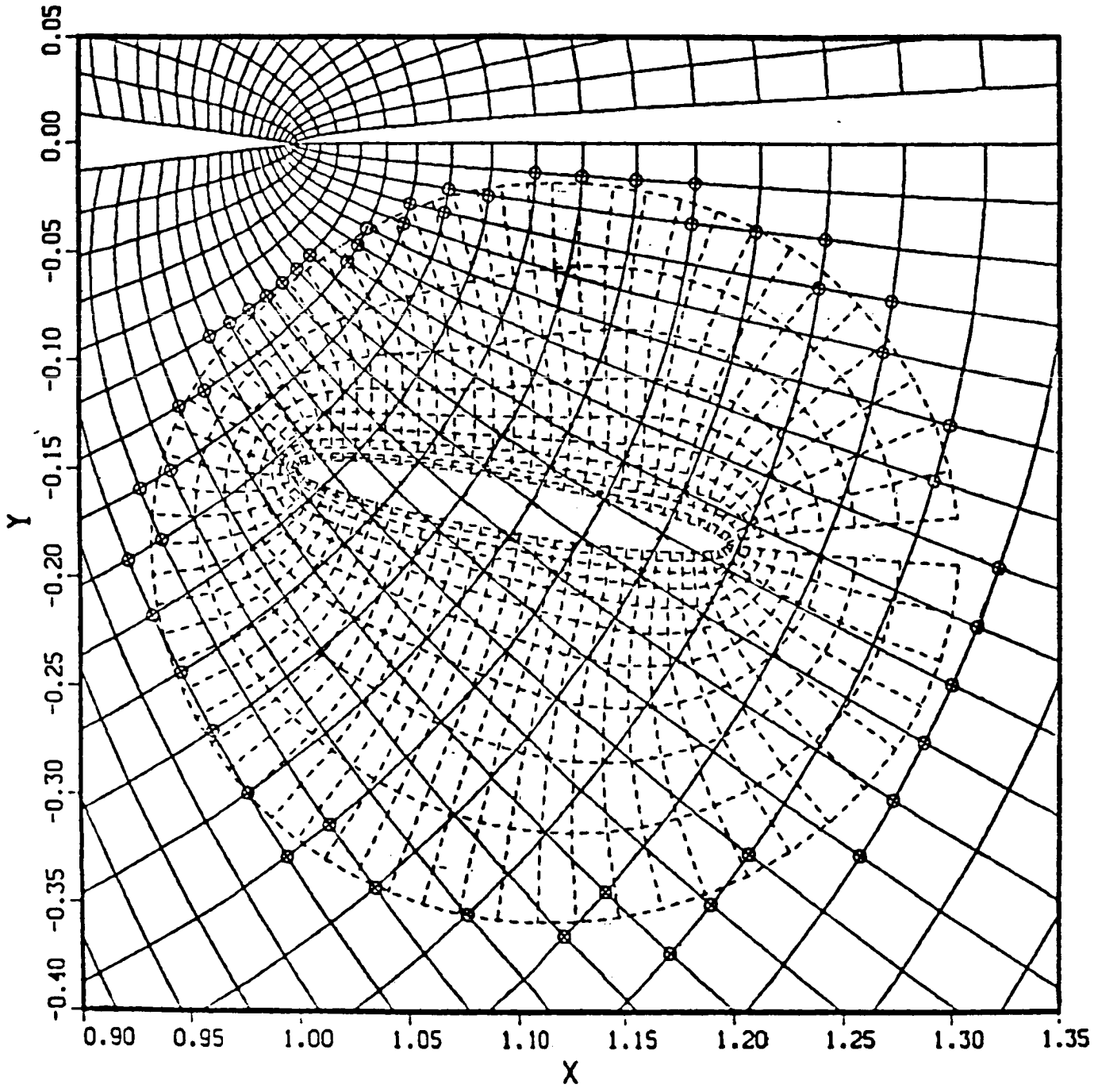
Mode	Loop	Bombard	LES	None
Static	1.85 / 31.1	2.13 / 35.9	1.98 / 33.2	* / *
Stack	1.85 / 31.1	2.15 / 36.2	1.99 / 33.5	* / *
Mtsk–1	1.86 / 31.1	2.16 / 36.3	2.00 / 33.6	* / *
Mtsk–2	* / *	* / *	* / *	1.96 / 33.0

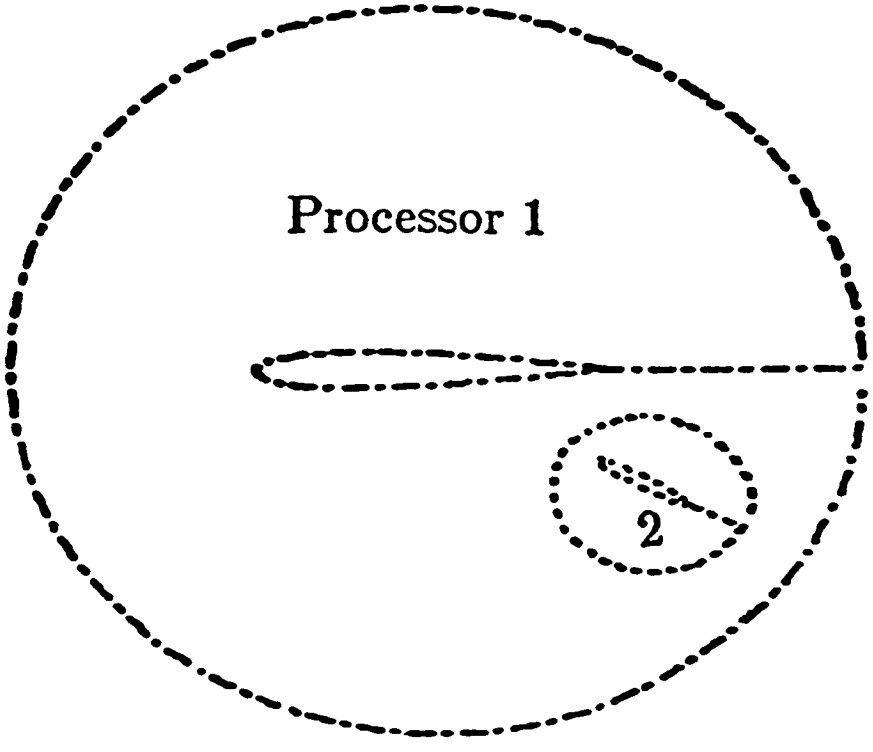
Time in Seconds	
Task 1	16.8
Task 2	16.2
Total	33.0

Performance of Multitasking on the Dual VAX 11/780

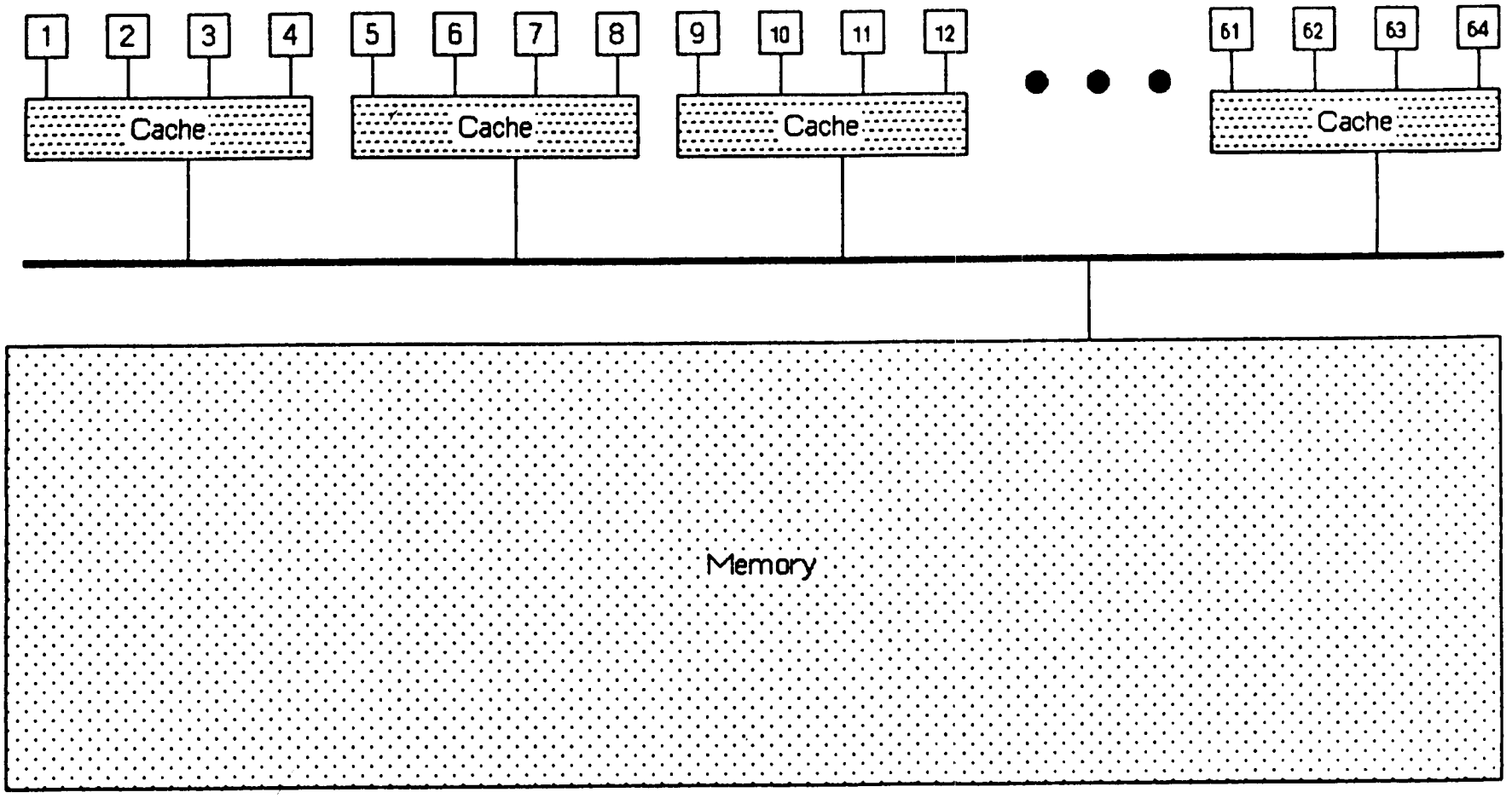
Code	Speedup
Twing	1.55
AIR3D	1.85
LES	1.98


1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00
Y
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
X


CHIMERA GRID
0.05
0.00
-0.05
-0.10
-0.15
-0.20
-0.25
-0.30
-0.35
-0.40
Y
0.90
0.95
1.00
1.05
1.10
1.15
1.20
1.25
1.30
1.35
X


Processor 1
2

PPA Architecture



Circuit-switched Network Simulation

Motivation and Objectives

- Understand performance of networks which could be used to build high-performance parallel architectures
- Use a real application (LES) from Ames to generate data for this study
- Understand how a real CFD problem could map onto a large MIMD architecture

The Model

- A circuit switched Omega network serving multiple processors connected to multiple modules of a shared memory
- Queues of requests exist at each processor port and are served one at a time

Construction of the Simulator

- Discrete event simluation facility of SLAM driven by FORTRAN subroutines
- Statistics collected on service times

Bandwidth of Network for Various Cases

Three cases:

- Real data from a CFD code (LES)
- Random data
- Infinite vectors with p=1

Total Bandwidth in MW/sec.				
n	MAX	Random	Vectors	Actual
8	36	12.5	5.52	5.75
16	67	12.2	5.60	5.62
32	123	5.12	5.12	5.24
64	229	5.76	4.16	4.36

For comparison look at Crays:

Maximum Bandwidth in MW/sec.	
Machine	Bandwidth
Cray 1	80
Cray X-MP	631
512*512	1500

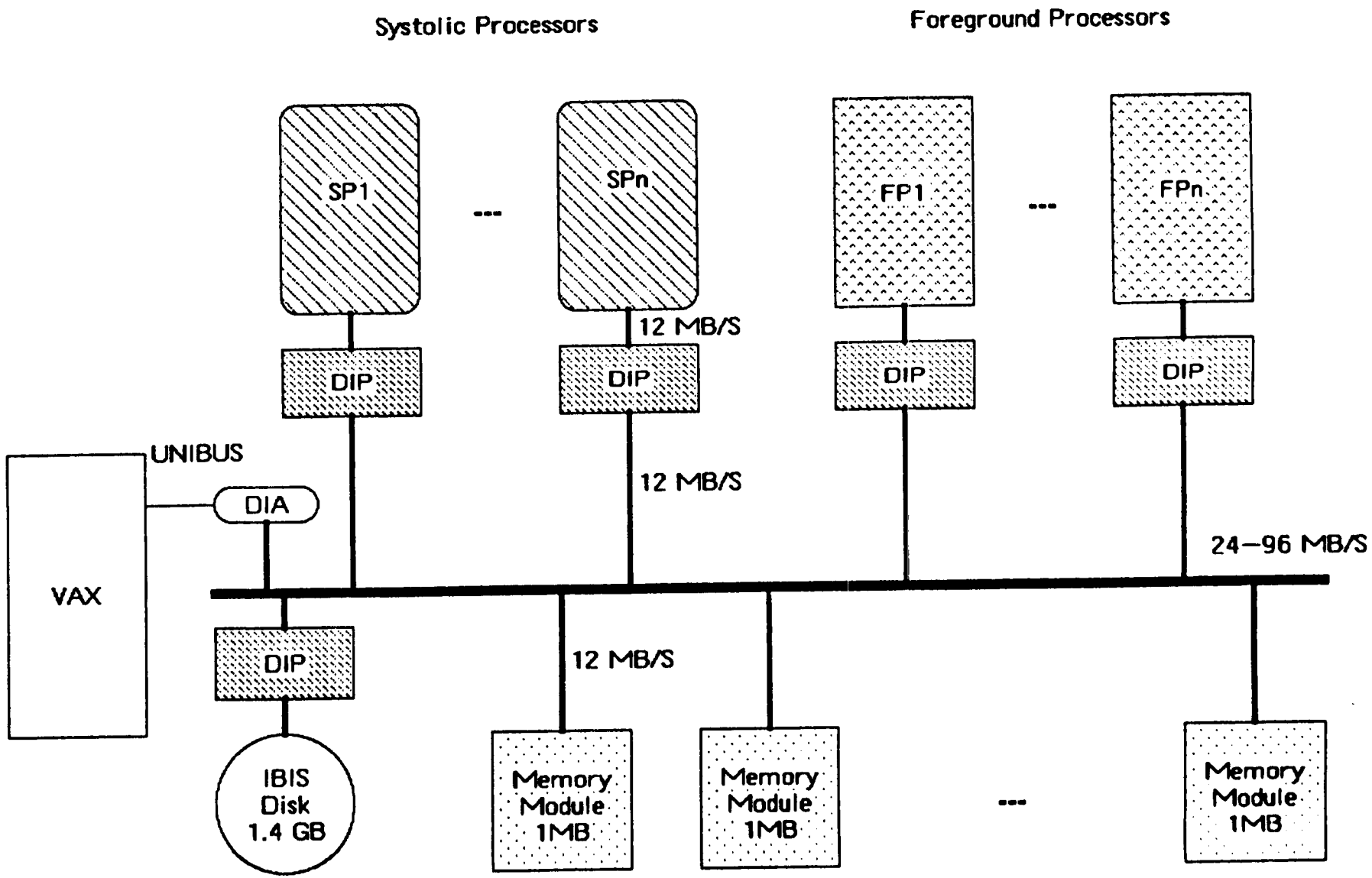
Conclusions from the Network Simulation

Modelling network traffic with streams of random data can be very misleading since actual codes exhibit a very different behavior

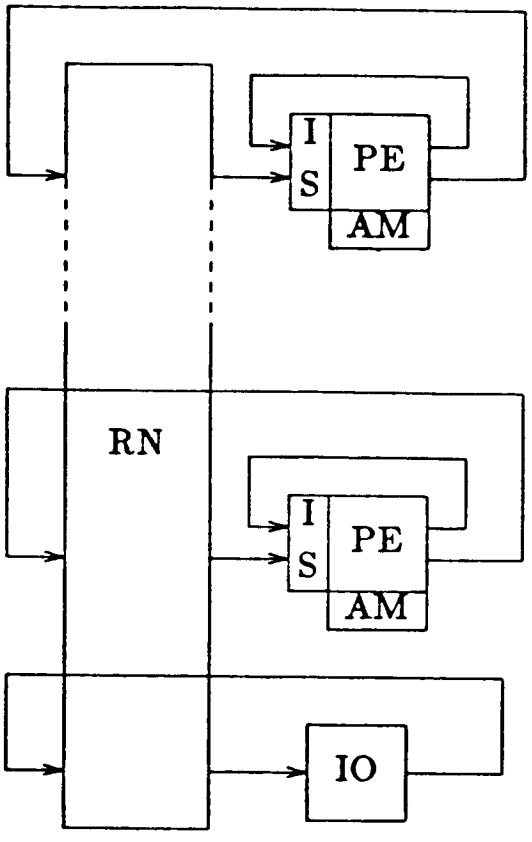
The bandwidth of the network does not increase linearly with the number of ports

A circuit-switched network such as this is far too slow to be useful for building high-performance MIMD architectures

System Architecture of a Systolic Attached Processor



Static Data Flow Machine Architecture



RN Routing Network. 512 by 512, 16 bit data paths, operates at > 5MHz, average rate of transmitting FP packets 0.25 MHz from a single PE to another.

PE Processing Elements. 5 to 8 MFLOPS with two 1.25 to 2 MFLOP multipliers. 256 PE's in the system.

IS Instruction Store. 1024 cells for FP instructions, 1024 for others.

AM Array Memory. Size not fully determined. At least 256K 64 bit words per PE.

IO Input Output. Includes mass memory, host processor, and display systems. 256 paths through the RN are reserved for IO.

Status of Data Flow Simulator

Design of simulator complete.

Coding of simulator begun.

Coding being done in PASCAL, and problems encountered with CRAY compiler

Input Codes are being developed

Questions to be Answered by the Simulator

Are previous performance predictions realistic?

What is the load on the routing network? Can the network handle it?

How much instruction memory and array memory is needed?

What is the effect of adding more processors?

What is the best way to distribute instructions across the processming elements?

GRAPHICS RESEARCH AND DEVELOPMENT

PURPOSE: PROVIDE FOR GREATER USER PRODUCTIVITY BY ENABLING VISUALIZATION OF 3 DIMENSIONAL EXPERIMENTS AND SIMULATIONS (EG. VISUALIZATION OF FLUID FLOW IN THREE DIMENSIONS)

RESULTS:

Established consortium agreement with Robert Barnhill (Utah) to develop algorithms for generating smooth contours from sparse random data such as those from wind tunnel tests

Developed State-of-the-Art Three-Dimensional graphics program for the Silicon Graphics IRIS terminals and demonstrated its use for several computational physics applications

Organization of Data Flow Simulator

DRIVER: Defines Characteristics of the architecture to be simulated, e.g. network characteristics number of processing elements, number and type of functional units in processing elements, etc.

TRANSLATOR: Takes code written in intermediate Data Flow Language (IF1) and translates it to input for the simulator (LLNL supplying SISAL to IF1 front end)

SIMULATOR: Perfroms actual simulation

DRIVER will run on VAX to allow interactive use. TRANSLATOR and SIMULATOR will run on Cray because of length of run.