Processor Characteristics

Mainframe

$10^8$ transistors

Intermediate

$10^6$ transistors

Systolic Array/Macromodule

$10^4$ transistors

Memory (10 Mb)
Comparison of Three Types of Processors

Problem: conversion of a 1000x1000 matrix to upper triangular (no pivoting).

<table>
<thead>
<tr>
<th>processor</th>
<th>sequential steps</th>
<th>time/step</th>
<th>total time</th>
<th>size</th>
<th>#chips</th>
<th>cost</th>
<th>cost*time</th>
</tr>
</thead>
<tbody>
<tr>
<td>mainframe</td>
<td>10⁹</td>
<td>1 uS</td>
<td>10³ sec</td>
<td>1 Mb</td>
<td></td>
<td>$200K</td>
<td>2x10⁸</td>
</tr>
<tr>
<td>10x10 NNCP</td>
<td>10⁷</td>
<td>10 uS</td>
<td>100 sec</td>
<td>100 bds</td>
<td></td>
<td>$200K</td>
<td>2x10⁷</td>
</tr>
<tr>
<td>1000x1000 systolic</td>
<td>10³</td>
<td>100 uS</td>
<td>.1 sec</td>
<td>10⁶ chips/10⁴ boards</td>
<td></td>
<td>$20M</td>
<td>2x10⁶</td>
</tr>
</tbody>
</table>
Software Development Plan

Hardware Machine Model

User Programming

Phase 1 Virtual Machine Model

Rigid CSP

Phase 2 Virtual Machine Model

CSP Extensions

Phase 3 Virtual Machine Model

?
Hardware Machine Model

CPU

$10^6$ bits memory

interface 0

message ready

buffer empty

interface 1

other interfaces
Phase 1 Machine Model

Features: Storage Allocator  
Process Scheduler  
Interprocess Communication  
Physical I/O
Phase 2 Machine Model

Features:
- Message Routing
- Debug Aids
- Diagnostics
Phase 3 Machine Model

Features:
- Dynamic Process Movement
- Global Pointers
Engineering/Science Applications

(Except architecture research)

* PDEs
* Matrix operations
* Sparse matrix operations
* Computer chess

1/2 Simulation (circuits)
  Design rule checking
  Weather prediction
  Oil exploration
"The Deal"

**You get:** 10:1 increase in available computing resource.

**Cost:** Different programming style.

**Computer Science gets:** Algorithms research and experience in concurrent computing for only the cost of guidance.
Spatial Decomposition of Problems

**PDE:**
- Conventional spatial
  - Update lattice points sequentially

**DRC:**
- Conventional spatial
  - For every pair of polygons
    - Check all design rules
  - (1) Sort self
    - (2) Make sure neighbors are far enough away
Limitations

Programming styles fall into two categories:
(1) those that allow general abstraction, and
(2) those that are implemented and are efficient.

(mutually exclusive)
Project Schedule

Hardware
  DeBenedictis
  Seitz
  Athas
  Cavallero

PDE's
  Fox
  Brooks
  Otto
  Martin

Lisp
  Athas

Cope
  Lang (until he leaves)

Actors

Volunteers??
Motivations For Homogeneous Machine

1) User community exists before the machine is built.
2) Cost/Performance of the machine will be considerably less than existing machines (performance leverage).
System Construction

System Step 1: Nearest Neighbor Communication. (Read/Write to any physical link, and guards.)
* Loader.

User Step 2: Systolic Applications.
System Step 2: Storage Allocator. (Unix style MALLOC/FREE.)
* Processes.
* Internal Ports.
* External Ports (Message Routing.)

User Step 3: CSP Applications.
System Step 3: Process Movement between CPUs.
* Global Pointers.

User Step 4: Object Oriented Applications
Actor Systems.
Hardware Diagnostics

Self replicating diagnostic programs for CPUs and I/O links.
Software Diagnostics

* Provide some secure software, either in ROM or by using a supervisor/user mode feature. Software includes:
  * I/O primitives capable of distinguishing monitor messages from user messages.
  * Debug software. All that is required is deposit/examine memory and process control.
* Provide a debug program in a host with a user interface.
A process is an independent program except:

1. Port reference data type.
2. "Create port" capability (monitor call).
3. Knows the names of other processes (like knowing how to invoke another procedure in PASCAL).
4. "Create process" capability.
5. Can send and receive messages via a port reference.
Applications to Systolic Programming

Given a Systolic Program consisting of:

Topology:

Program A:
Begin ~ End

Program B:

Program C:

Implementation:

"Master Setup Program" uses the topology specification to create ports and processes.
Applications to CSP Programming

CSP has a nearly static topology (only arrays of processes are allowed), but allows communications between processes, not ports:

\[
P_2! X(i); \quad P_1? X(j); \quad P_2? Y(j); \quad P_1! Y(8)
\]

Implementation:

Master setup program does following:
Actor / Object Oriented
Hardware Design Considerations

Minimize \[
\frac{Cost}{Performance}
\]

Performance Measures:

- Maximum size of problem \(\Leftrightarrow\) Total memory in all processors.
- Speed of solving problem \(\Leftrightarrow\) \# of CPUs available to user.

   (note about overhead, routing processors.)

CIT 6-cube:

- 64 CPUs, 1/2 mb /CPU, 1/4 VAX/cpu (8/10 MHz)

\[\Rightarrow 8 VAX \& 32 VAX\]

Designable Parameters:

- Size of memory
- CPU speed / communication time
Size of Memory

We choose: 65K words (16 65K RAM chip)
Too big?? 65K bytes better??

Reasons:

Each CPU must be large enough to hold:

- operating system (8K bytes)
- user program (8K bytes) (modular)
- user data (48K bytes) (smaller-the-better)

Experience: HP people always want more memory, but after
some persuasion they will settle for more CPUs with
equal memory.

Exceptions: Some real programs can run into substantial amounts
of code. It remains to be seen if this code can always be
modularized. Actor-operating systems are presently huge.

Psychology: We want to encourage concurrent programming. With
large amounts of memory naive users adopt a "multiple Von-
Neuman" style. By keeping the memory at the lower limits of
the useful range, we will coerce users to concurrent programming
sooner.
**CPU Speed / Communication Time**

We choose: 20 µs 164 bit message transfer
20 µs / floating point ... about equal

To be competitive with conventional VNMs we must have a factor of 10 cost/performance advantage. We can only maintain this advantage by skimping on the communication.

<table>
<thead>
<tr>
<th>Application</th>
<th>Communication Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systolic</td>
<td>&lt;10%</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Actors</td>
<td>Embarrassingly high</td>
</tr>
</tbody>
</table>

**COMMUNICATION CHIP!!**