

# Computer Systems Architecture II

## Chip-Multiprocessors: Applications and Architectures

CSE 561M

Prof. Patrick Crowley

# Plan for Today

- Announcements
  - Tuesday, final ONL help session
  - Tuesday, final milestone report to newsgroup
- Questions
- Project logistics
- Paper presentations and discussions

# Project Logistics

- Weekly Milestones
  - Milestone report due each week
  - Submit proposal Feb 26
  - Final presentation May 1
- Focus this week: Project wrap-up

## Project Milestones

<i>M0</i>	<i>Feb 26</i>	<i>Project Proposal</i>
<i>M1</i>	<i>Mar 4</i>	<i>Design</i>
<i>M2</i>	<i>Mar 18</i>	<i>Implementation 1</i>
<i>M3</i>	<i>Mar 25</i>	<i>Implementation 2</i>
<i>M4</i>	<i>Apr 1</i>	<i>Implementation 3</i>
<i>M5</i>	<i>Apr 8</i>	<i>Implementation 4</i>
<i>M6</i>	<i>Apr 15</i>	<i>Wrap-up, Prepare reports</i>
<i>M7</i>	<i>Apr 24</i>	<i>Reports &amp; code due via email</i>
<i>M8</i>	<i>May 1</i>	<i>Presentations/Last meeting</i>

# Project Reports

- Contents
  - Problem statement, description & background
  - Your design
  - Your implementation
    - Include characteristics of your code, such as code size, ME utilization, memory utilization, etc.
  - Experimental results
    - Demonstrate correct operation
    - Measure and explain the performance of your code
  - Future work (that you might do if you had time)
  - Division of labor amongst group members
- Length:  $5 \leq \# \text{ single-spaced pages} \leq 15$
- PDF is due via email on Thursday, April 24
  - Project code also due at that time

# Project Presentations

- Project Presentation
  - 15 minutes
  - All group members should contribute
  - Content:
    - Subset of project report
    - What you did, why you did it, and how well it worked
    - Plan a live ONL demo
- May 1, during our final exam slot

# Paper Presentations

- Each paper discussion will involve
  - A 15-minute **group** presentation
    - Summarize the paper's content and contributions
    - Relate it to your project (e.g., what if you had targeted that system for your project rather than the IXP?)
    - All group members should contribute
  - A 15 minute follow-up lecture
- Each group has an assigned paper & date
- Submit presentation via email by 2pm that day

# Paper Discussions

- Wulf and Harbison's Reflections in a Pool of Processors/An Experience Report on C.mmp/Hydra
  - Presenters: Erik Church, Paul Sebert
- Seitz's The Cosmic Cube
  - Presenters: Joe Clinch, Rich Hill
- Dean and Ghemawat's MapReduce: Simplified Data Processing on Large Clusters
  - Presenters: Mart Haitjema, Ritun Patney, Shakir James

*Reflections in a Pool of  
Processors/An Experience  
Report on C.mmp/Hydra*

# What is C.mmp/Hydra?

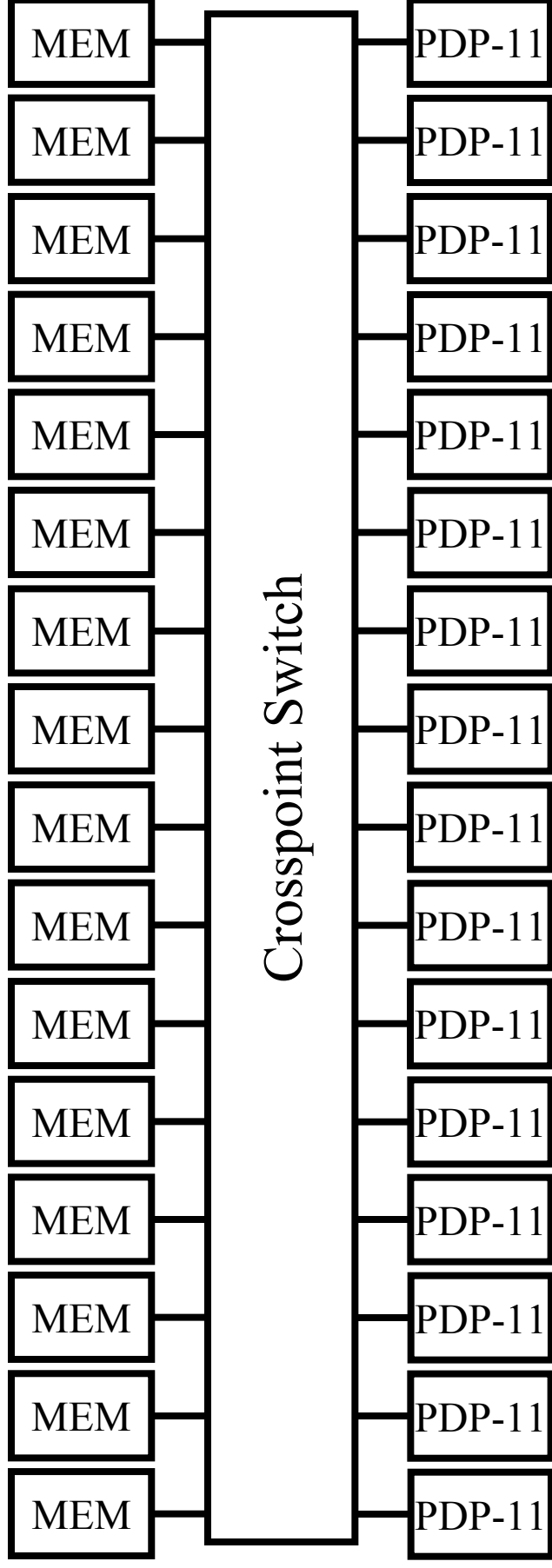
- C.mmp
  - CMU MultiMiniProcessor (built with PDP-11s)
  - No caches, uniform memory access from all CPUs
  - Goal: *Cost-effective* general-purpose shared-memory multiprocessor
- Hydra
  - Kernel-based OS for C.mmp
  - Goal: OS services can be built by users to enable experimentation

# What is a PDP-11?

- 16-bit computer from DEC
  - late '60s, early '70s
- CISC
- Tidbit: first machine with only a memory bus, no bus for I/O.
  - All I/O was handled through special memory addresses
- More info:
  - <http://en.wikipedia.org/wiki/PDP-11>

# C.mmp Organization

- 16 processors, 16 memory banks
- Total performance: 6 MIPS, 500 Mbps



# Hydra

- OS kernel on top of which OS services can be built, by users
- Protection and object reference based on *capabilities*
  - i.e., the only references available are capabilities, which encode the access rights of the holder

# Successes and Failures

- **Successes:** (1 of 5 are HW)
  - Cost-effective, symmetric MP
  - Hydra is user-extensible
  - Hydra is distributed
  - Reliability in software
  - Good Hydra engineering
- **Failures:** (3 of 5 are HW)
  - Hardware is not reliable
  - PDP-11 has small address space
  - C.mmp cannot be partitioned
  - Poor user interface
  - Poor project management

# A Cost-effective Multiprocessor

- Design goals
  - Speed
  - Simplicity
  - Off-the-shelf components
- System complaints
  - PDP-11 has too few addresses (64K)
  - High OS overhead
  - Memory contention

# Extensible OS

- The extension abstraction is the protected subsystem
  - Implemented over 20 subsystems (i.e., services)
- Verdict
  - Abstraction was good for fast implementation, revision and integration with existing system
  - Development environment poor
  - It was difficult to build schedulers outside the kernel

# The Distributed OS

- All Hydra instances on all processors are peers
  - no client/server relationships needed
- System tasks can run anywhere
  - Except for I/O tasks which must run on a processor attached to the particular device
- Provides a range of synchronization primitives
  - Fast locks
  - Slow semaphores

# Reliability

- Goal: provide software error detection and recovery, regardless of error source
- Error detection
  - HW: parity checking on all memory transactions
  - SW: redundant representations, etc.
- Error recovery
  - *Suspect-monitor* paradigm
  - When one processor gets hung, another re-boots it

# Hardware Reliability

- Mean-time-between-failures for C.mmp/Hydra: 2-6 hours
  - 2/3 of failures were 100% HW
- Culprits
  - PDP-11 UNIBUS
  - Drums (memories)
  - Crosspoint switch can be hung by misbehaving clients
  - HW group wrote poor diagnostic code

# Address Space

- Only 64K addresses
- Consequences
  - OS had to help extend memory size
  - Programs/data must be fractured
- Realizations
  - Users did not write small program pieces
  - OS-managed paging was expensive

# System Partitioning

- Idea: disjoint subsystems will enable concurrent usage among: users, diagnostics, maintenance, upgrades, etc.
- Infeasible on C.mmp/Hydra
  - Although some maintenance can be partitioned
  - Not enough I/O devices to go around
  - Design does not account for the sharing or communication of capabilities between partitions

# Human Interface Engineering

- User interface did not receive much attention
- Consequences
  - Long learning process
  - System lacked conventional conveniences and languages
  - However, the Command Language was a big hit (allowed complete access to Hydra environment)

# Address Space Problem

- Program run on several PDP-based machines
- Dynamic vs. static
  - Paging is never needed, but dynamic version checks anyway

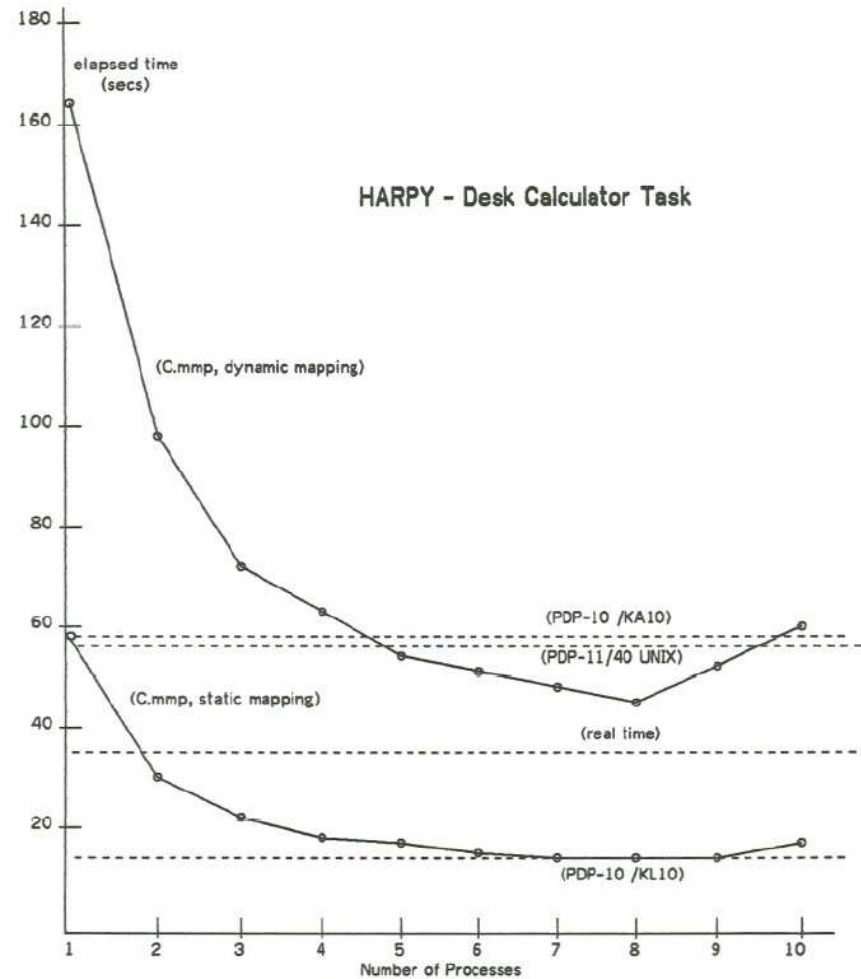


Figure 1—A look at the small address problem

# Project Management

- Medium-sized team, 15 or so
- Failures
  - Too few people, wrong personnel
  - Novel aspects of system were over-emphasized, essential (and mundane) tasks were neglected
  - Informal management style felt right, but, in fact, led to problems in specifications, documentation and coding standards
    - Hydra team did not use C.mmp for development
    - Lack of specs prohibited parallel HW/SW development

# Intellectual Digestion

- What purpose does an OS serve?
- Does the IXP have an OS? Does it need one?
- Could you build something like the C.mmp in your spare time? How much would it cost?

# Purpose of OS

- To provide
  - Common services
  - Rights-based access to resources
  - Protection between different programs
  - Resource sharing to applications
  - A record of resource usage and system activity
  - A higher-level implementation target for applications

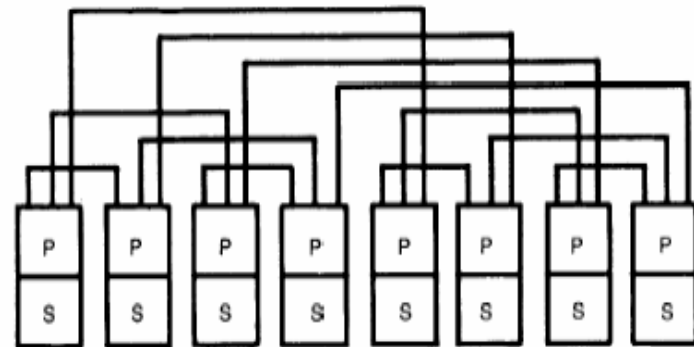
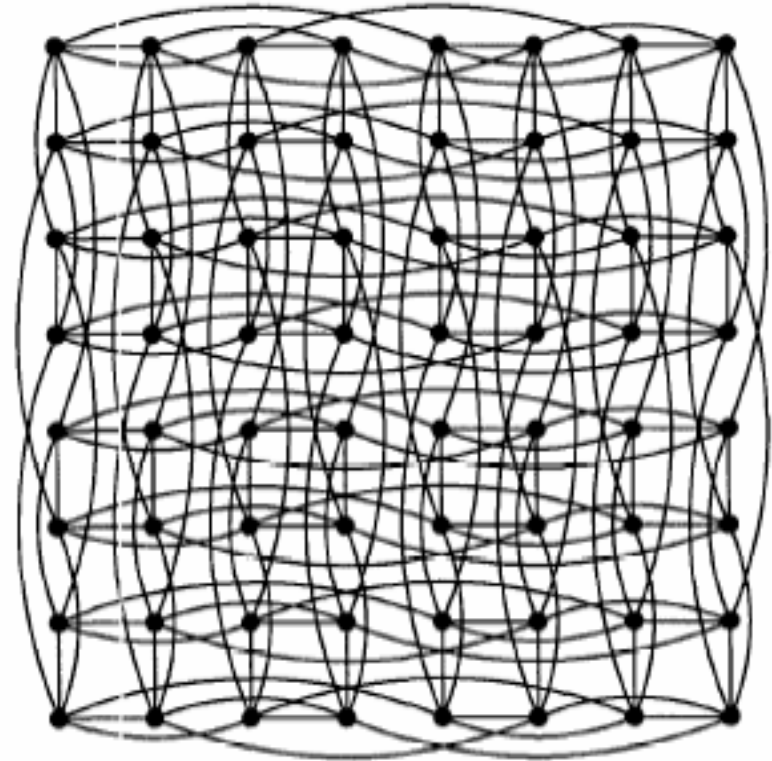
# OS on the IXP

- OS existence
  - XScale certainly has one
  - MEs do not
    - Although, certain tasks are supported in HW, such as synchronization and thread scheduling
- OS demand
  - Do MEs need separation?
  - Do MEs need to share resources?
  - Do MEs need to protect resources?
- Put another way, how would you use the IXP if it had this support?

# *Cosmic Cube*

# Cosmic Cube

- Large-scale message-passing multicomputer designed for scientific applications
- Does not use shared memory



# Cosmic Cube



The nodes are packaged as one circuit board per node in the long card frame on the bench top. The six communication channels from each node are wired in a binary 8-cube on the backplane on the underside of the card frame. The separate

units on the shelf above the long 8-cube box are the power supply and an "intermediate host" (IH) that connects through a communication channel to node 0 in the cube.

FIGURE 6. The 64-Node Cosmic Cube in Operation

# Interconnect

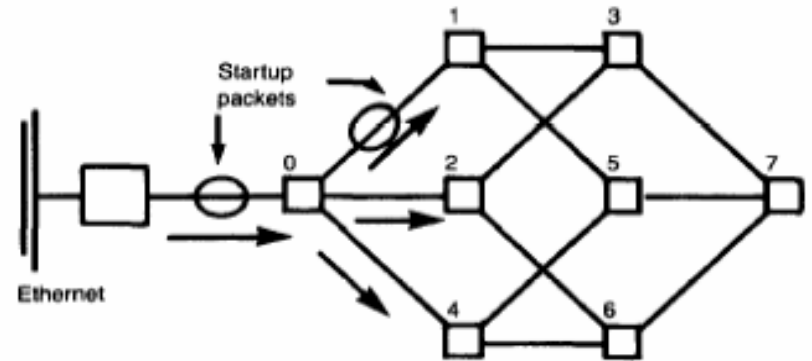
- Six-dimensional hypercube
  - Binary, 6-cube
  - $N = k^n$ , k-ary, n-cube: n dimensions, k nodes per
    - When  $k=2$ , it's a hypercube
  - $64 = 2^6$
- Direct point-to-point links, no switches
- Scales to many nodes
- Question
  - When  $N=64$ , what is the maximum distance between any two nodes?

# Programming Model

- Communicating sequential processes
- OS provides send/receive primitives
- Process distribution directed by programmer; programmer maps solution onto network
- Good fit for modeling physical systems, where physically distinct computations exist

# Hardware

- Node
  - Intel 8086/8057, 5-6 MHz
  - 128KB RAM
  - 8KB read-only initialization memory
  - 2 Mbps links
- Cube would be attached to a host
- Note: far more reliable than C.mmp/Hydra
- Seitz considered this to be a simulation of a future single-chip node



# Intellectual Digestion

- Is the interconnect strategy described scalable? Explain.
- Is the cosmic cube programming model more or less complex than that of the IXP? Explain.
- What aspect of the Cosmic Cube was the most interesting to you? Is it also the most significant aspect?
- Consider how you would have solved your IXP project on the Cosmic Cube. Comment on design, performance, and ease of programming.

# Assignment

- Tuesday
  - Milestone 6: Project completion
    - Use the template in this presentation and post your milestone report to the newsgroup. Bring a hardcopy to class.
- Thursday (submit PPT by 2pm)
  - Smith's Architecture and Applications of the HEP Multiprocessor Computer System
    - Presenters: Steve Nann, Tam Vu Ngoc, Dan Vianello
  - Dennis and Misunas' A Preliminary Architecture for a Basic Data-Flow Processor
    - Presenters: Steve Prochazka, Mark Dunn
  - Shriraman et al.'s Flexible Decoupled Transactional Memory Support
    - Presenters: Eitan Marder-Eppstein, Stu Glaser