Aries

High Level Architecture
And
System Implementation
Large-Scale System Challenges

- Large-scale systems: 1K to 1M+ processors
  - 100M+ components
  - 1M processor chips would keep TSMC’s $2.4 billion Fab-6 busy for one month solid at full tilt, given good yields
  - one high-end automated chip-shooter would need about 700 days @ 24/7 to just place the parts on a machine with 1M Aries chips
- “small problems” become significant challenges:
  - Assembly
  - Integration and Configuration
  - Testing
  - Reliability
  - Cooling
  - Speed of Light
- Compare to Amorphous Computing
Better Living Through Architecture

- Simple modules with a common interface
- Self-checking and correcting capabilities in each submodule
- Easy customization for varying bandwidth/processing requirements (scalability)
- Fully distributed processing, storage, power conversion and cooling capabilities
  - Design-out catastrophic single-point failures
A Sketch

Network

optimal bisection
BW is dependent upon
user application

P-node

P-node

P-node

6.911
Box Architecture

• Three components
  – Processor boxes
  – Network boxes
  – Backplanes (racks)
• Vary mix of processor to network boxes to meet bandwidth requirements
• Flexible network box architecture to allow various topologies and paranoia levels
• All components hot-swappable
• Field-serviceability is simplified
Basic Box Architecture
Rack Numbers

• Some ball-park numbers for a single fully-populated processor rack (both sides):
  – 1024 processors
  – 2G active RAM
  – 64 G of disk buffer RAM
  – 2 TB of disk
  – 40 kW power dissipation
  – Backplane bisection BW ~400 GB/s
Network Topologies

• Fat tree topology with channel capacity scaling
  – Implemented as METRO-style multibutterfly networks with path-expansion
  – No single point of failure
  – Ref: “Fat-Tree Routing for Transit” by Andre Dehon

• Circuit-switched network assumptions
  – Time to connect is short compared to message transmission time
  – As time to connect gets longer, may need a hybrid packet-based or wormhole approach
Distributed Architecture for Fault-Tolerance

- RAID array per processor node
  - Higher bandwidth, more fault-tolerance
  - RAID ring topology (shared drives between adjacent modules)

- Power conversion is distributed
  - Redundant high-voltage distribution bus with local down-converters

- Cooling is distributed
  - Pump and heat exchanger in each module with redundant coolant distribution and coolant stores
The Cooling Challenge

- Removing heat faster lets us make more of it
- Stable die temperatures $\Rightarrow$ more performance margin
- Estimated 200W per Aries chip peak dissipation
  - Spray cooling or microchannels
  - Doesn’t count support chips (external DRAM, glue)
- Must keep physical size down to reduce speed-of-light impact
- Each box will dissipate in excess of 1 kW
  - liquid cooling is mandatory
- Either sealed-box (harder to service) method or gravity-assisted falling liquid film (messy)
Node Architecture

Bandwidths, error reporting

48 VDC

local regulators

coolant

heater exchanger

router

monitor processor

I/O and storage chip

2 GB buffer

RAID ring ports

distributed RAID-5 Array
/about 60 GB/node

about 16 MB PIM per chip @ 8 procs

6.911
Mega-Processor Machine

• Previous design good to about 10k-50k processors
• 1 M+ processors requires a different approach
  – full bisection BW would be about 10TB/s--a wire bundle about 14 meters (40 ft) in diameter
  – thinning by 1000 gives a bundle about 1ft in diameter
• Challenges in shear mass, connectivity, power, and assembly
• Must be a fully 3-D design
Wedge Assembly

![Diagram of Wedge Assembly]

- Cooling channels
- "Wedge" front view
- "Wedge" side view

6.911
Computronium

- Reflective routing for shortcut routing
- Spherical structure
- Router "core"
- Lasers for free-space interconnect
- Very large cable bundles
- Spherical wave RF clock distribution
- Supports carry power and coolant (perhaps more feasible if implemented in zero-G)
- More shells of computation if necessary
- "Wedge"
Near-Term Work

• Extensive simulations are necessary
  – Study processor to memory ratio tradeoffs
  – Measure bandwidth, latency impact upon performance
  – Test basic fault-tolerant principles

• Prototype hardware system
  – Built out of FPGAs and COTS components
  – Provide maximum flexibility in a MP hardware simulation environment
  – Run realistic benchmarks and gather stats, design data for the second system
Moore Board

- Processor and network simulation
- Implement test ISAs, architectures (DAE)
  - Two 1.5 million gate FPGAs with 49kB embedded SRAM
  - PC-100 interface for fast interface to diagnostic host
    - Disadvantage: no interrupts
  - On-board 256-bit wide, fast SRAM to emulate caches or main memory
- Leverage Moore’s law to make a decent implementation
  - Use SRAM-based technology that scales w/time—wait two years and you’ve got a board that’s got 2x the performance
Moore Board

256k x 128 bits ZBT SSRAM@200 MHz

Processor FPGA

3.2 GB/s

256k x 128 bits ZBT SSRAM

3.2 GB/s

programmable clock chip (up to 100 MHz)

displays and temperature sensing

Data I/F FPGA

2-4 GB/s, 10 - 20ns L

expansion (LVDS, 28 pairs)

PC-100 interface (0.8 GB/s, ~100 ns latency to processor)
FINI Board

- Flexible Integrated Network Interface board
  - Test platform for COTS network chips
  - Trial implementations of network protocols
  - Test out key physical design ideas

- 4 Virtex FPGAs plus DS90C387/DS90CF388 LVDS interface chips
  - SSRAM buffers to hold test data patterns
  - Programmable clock
Node Simulator

• Combine the findings from Moore and FINI
  – Add hard drives, I/O processor

• Add cooling apparatus
  – Heat exchangers
  – Microchannel-style heatsinks
    • Use silicon blanks bonded to active die to reduce chance of damage to active, valuable die

• Provide adequate mechanisms for monitoring functional unit loading and congestion
On-Chip Architecture

• A different world
  – Wires and signaling can be made much more reliable
  – Much higher density of wire to logic ratio available
  – Wire delays shorter due to shorter on-die distances
  – Integrated fault detection/correction strategies

• JP’s domain