Plenary Presentation to the Workshop on Frontiers of Extreme Computing:

Continuum Computer Architecture

Thomas Sterling

California Institute of Technology

and

Louisiana State University

October 25, 2005

Challenges to Extreme Computing

- Power consumption
- Exposure and exploitation of application Parallelism
- Memory “speed”
- Latency and clock propagation distance
- Chip I/O bandwidth
- Amdahl’s limits
- Reliability and error rates
- Programmability
- Cost of development (and applied research)
- Cost of manufacturing

Challenges to Computer Architecture

- Expose and exploit extreme fine-grain parallelism
  - Possibly multi-billion-way
  - Data structure-driven
- State storage takes up much more space than logic
  - 1:1 flops/byte ration infeasible
  - Memory access bandwidth critical
- Latency
  - can approach a million cycles
  - All actions are local
- Overhead for fine grain parallelism must be very small
  - or system can not scale
  - One consequence is that global barrier synchronization is untenable
- Reliability
  - Very high replication of elements
  - Uncertain fault distribution
  - Fault tolerance essential for good yield

Insidious Underlying Assumptions

- Near term incremental solutions will continue indefinitely
- COTS
  - Off the shelf conventional micros
  - High density dumb DRAMs
- Processor centric
  - Cache based
  - Program counter driven
  - > 100M gates
- Separate memory components
  - Can’t possibly afford to do anything else
- Multi-core
  - More of the same
- Message passing model of computation
  - Some vectors and threads thrown in as well
- ALUs are the precious resource
  - Processor and memory architectures designed around them
- Manual locality management
  - Programmer explicitly specified for latency avoidance

My own warped perspective

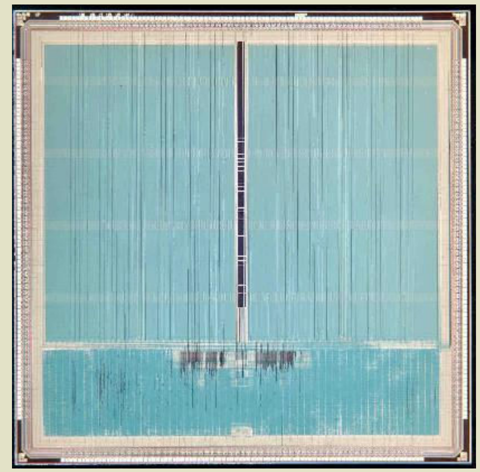
- Data access bandwidth is the precious resource
  - ALUs relegated to high availability devices (as opposed to high utilization)
- Conventional delineation of functionality is an artifact of ancient tradeoffs
  - e.g., Processor, Memory, Interconnect
  - Current strategy: Global parallel execution from local sequential devices
- Computation is about *continuations* and *state*
  - "continuation" specifies an environment and next action(s)
- Processors are one possible physical manifestation of a continuation
  - One continuation glued to each processor
  - Multithreading glues down a few continuations
- Barriers are bad
  - Over constrains parallel execution and destroys fine grain parallelism
- Meta-data determines control flow in some data intensive applications
- Control flow synchronization needs to be part of the meta-data
- Its some times better to move the continuation to the data than the data to the continuation
- Complexity of operation needs not be derived from complexity of design
  - Complex emergent global behavior may be a product of simple local rules of operation

Architecture Innovation

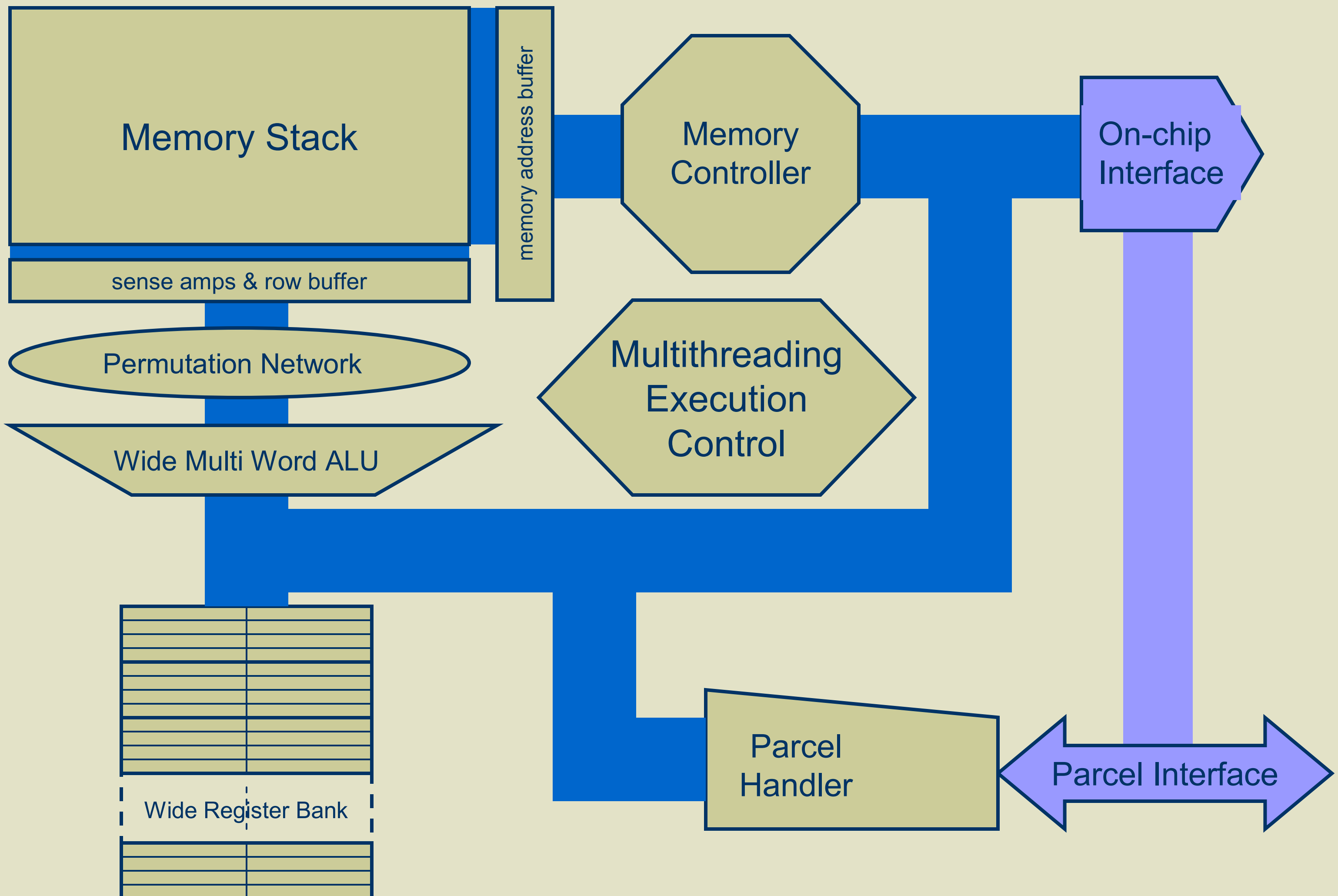
- Extreme memory bandwidth
- Active latency hiding
- Extreme parallelism
- Message-driven split-transaction computations (parcels)
- PIM
  - e.g. Kogge, Draper, Sterling, …
  - Very high memory bandwidth
  - Lower memory latency (on chip)
  - Higher execution parallelism (banks and row-wide)
- Streaming
  - Dally, Keckler, …
  - Very high functional parallelism
  - Low latency (between functional units)
  - Higher execution parallelism (high ALU density)

Concepts of the MIND Architecture

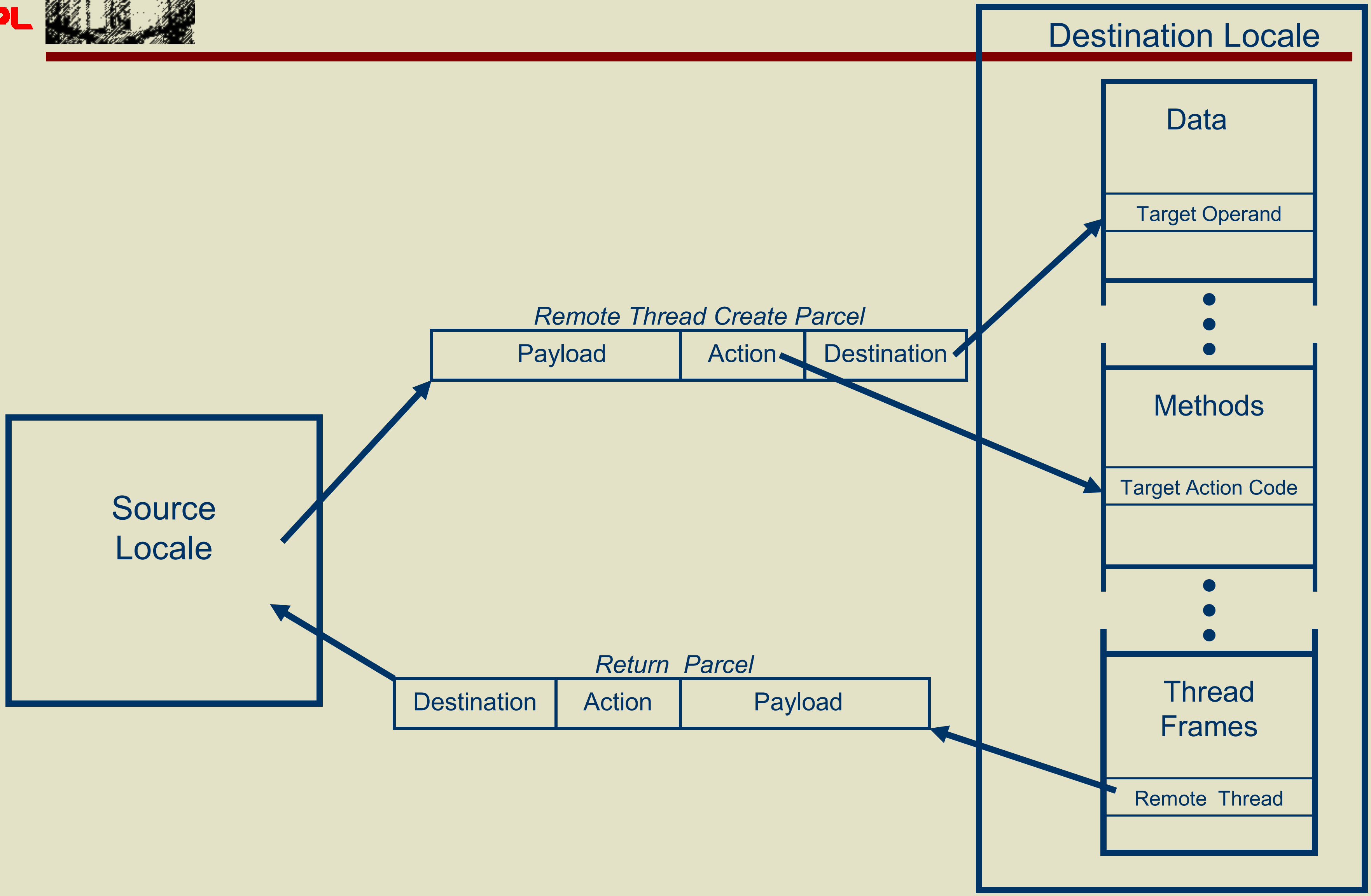
- Virtual to physical address translation in memory
  - Global distributed shared memory thru distributed directory table
  - Dynamic page migration
  - Wide registers serve as context sensitive TLB
- Multithreaded control
  - Unified dynamic mechanism for resource management
  - Latency hiding
  - Real time response
- Parcel active message-driven computing
  - Decoupled split-transaction execution
  - System wide latency hiding
  - Move work to data instead of data to work
- Parallel atomic struct processing
  - Exploits direct access to wide rows of memory banks for fine grain parallelism and guarded compound operations
  - Exploits parallelism for better performance
  - Enables very efficient mechanisms for synchronization
- Fault tolerance through graceful degradation
- Active power management



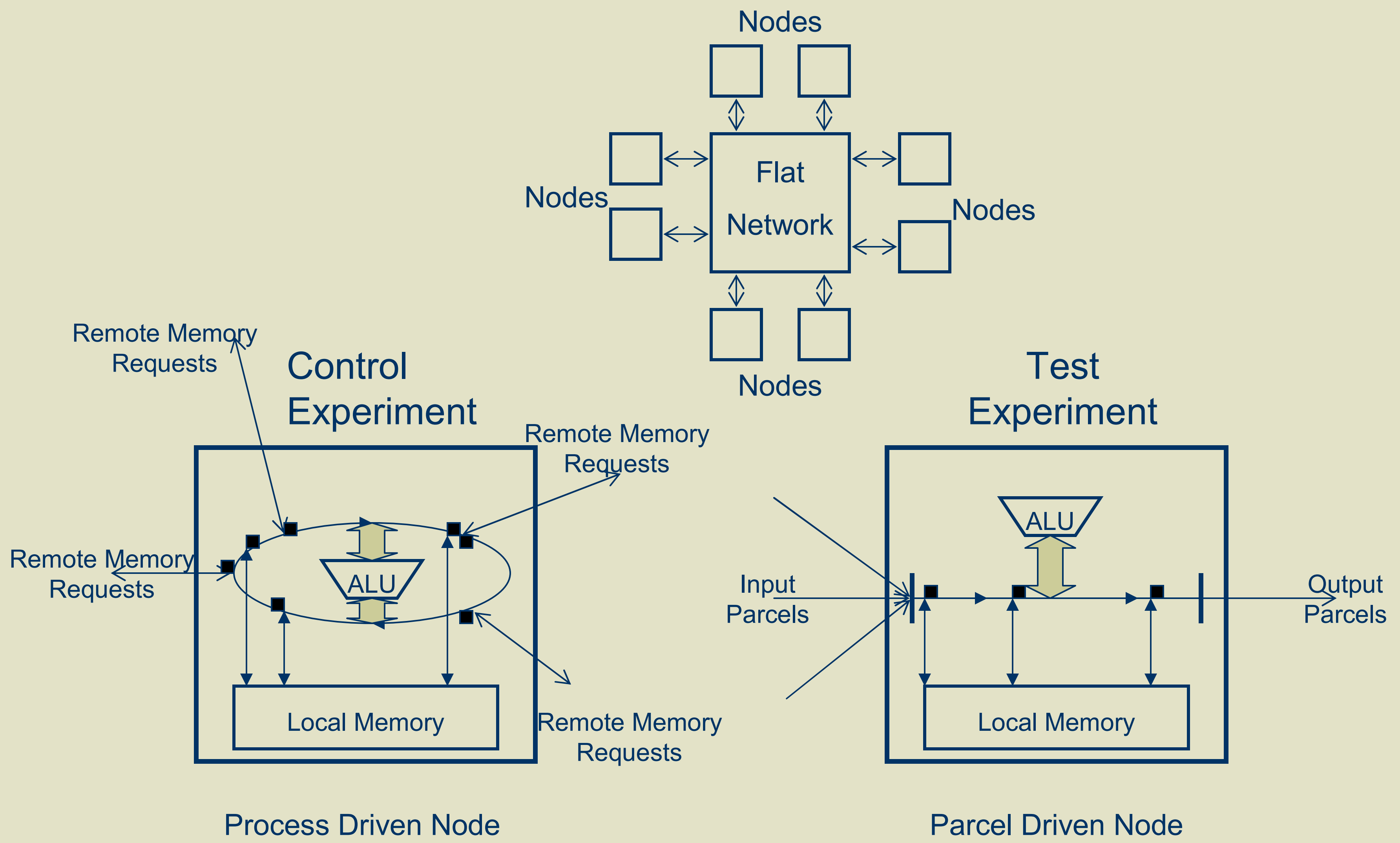
MIND Node


Memory Stack
memory address buffer
sense amps & row buffer
Permutation Network
Wide Multi Word ALU
Wide Register Bank
Memory Controller
Multithreading Execution Control
On-chip Interface
Parcel Handler
Parcel Interface

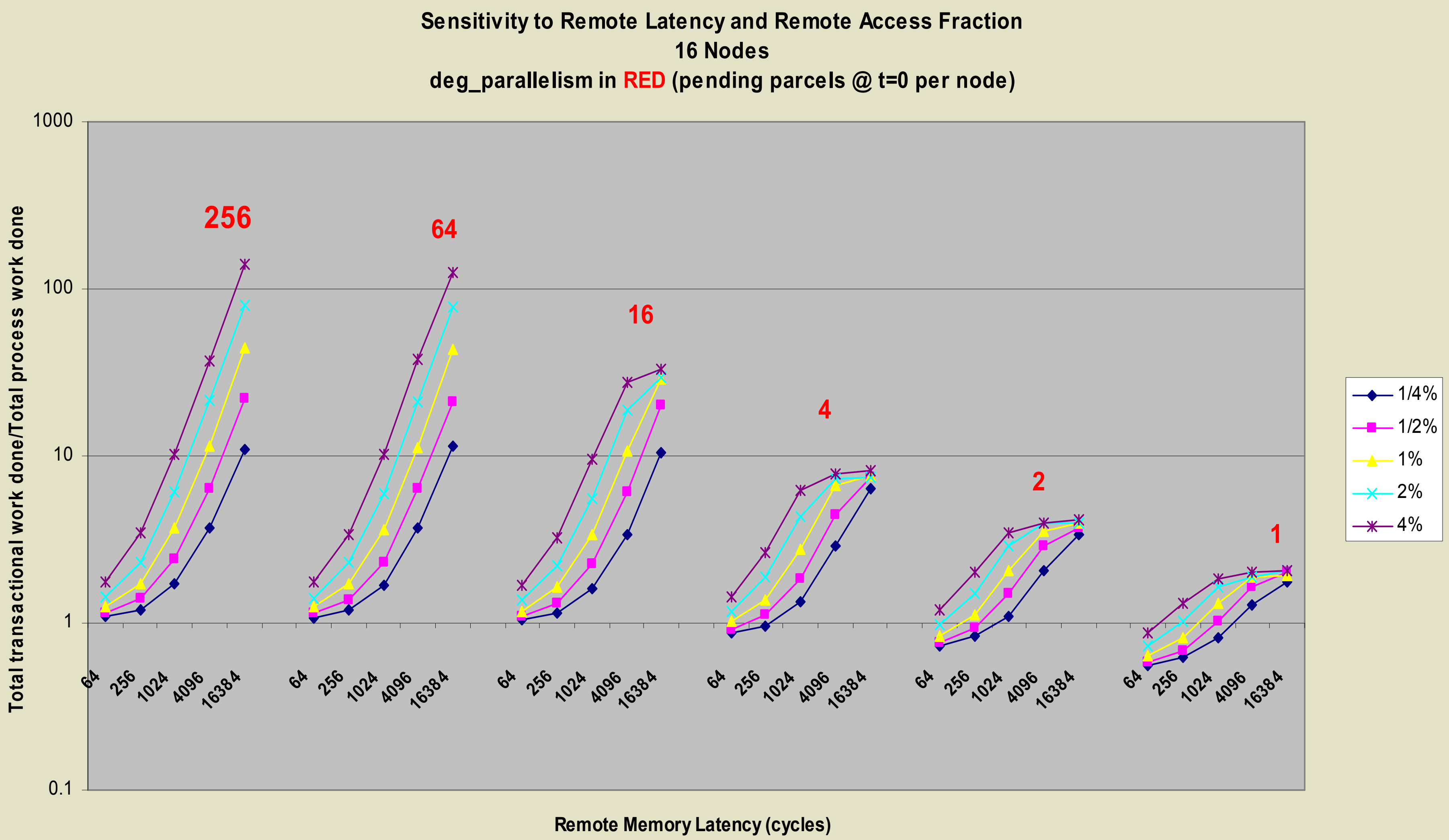
Parcels for remote threads



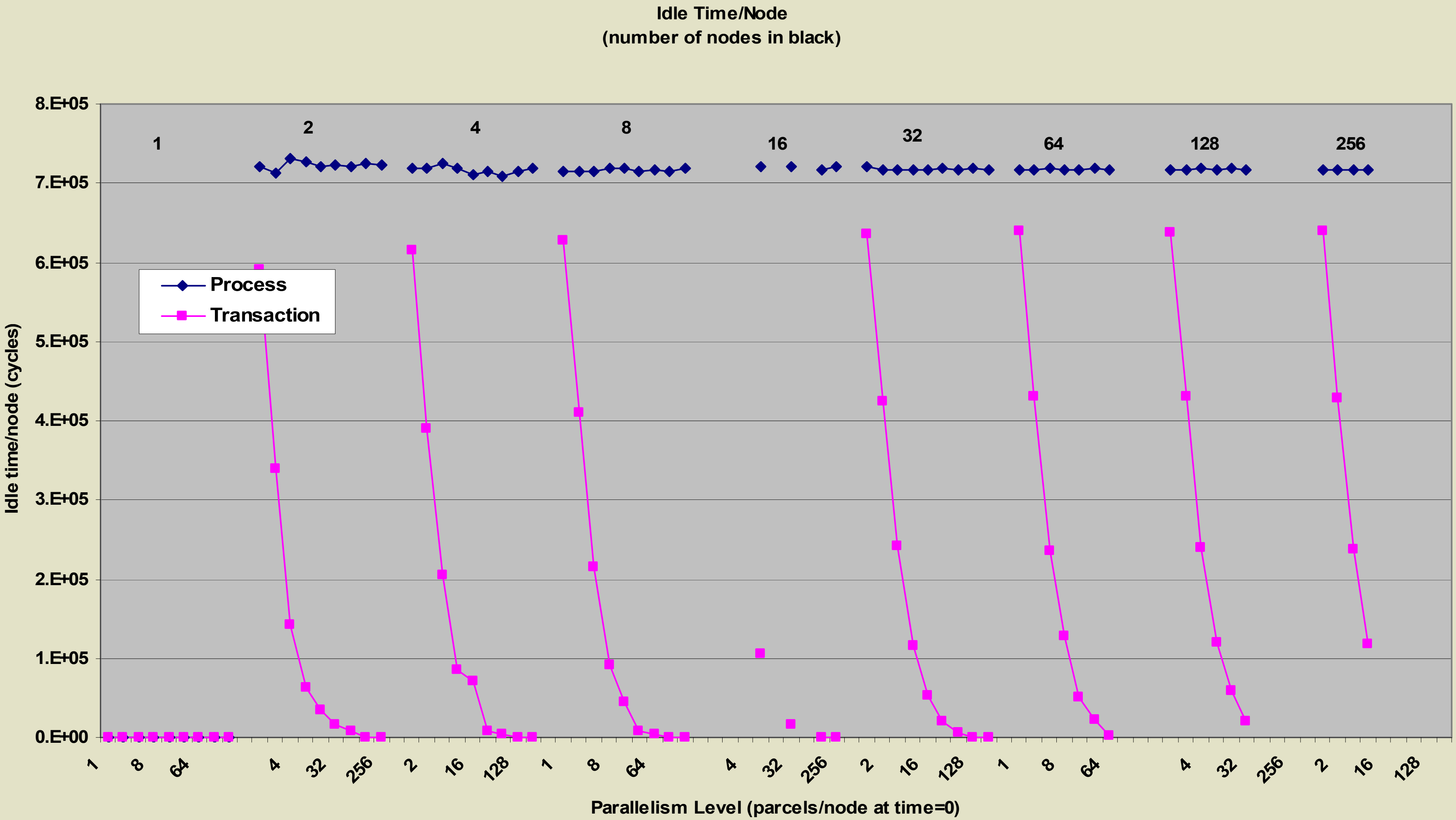
Parcel Simulation Latency Hiding Experiment



Latency Hiding with Parcels
with respect to System Diameter in cycles



Latency Hiding with Parcels
Idle Time with respect to Degree of Parallelism



Metric of Physical Locality, τ

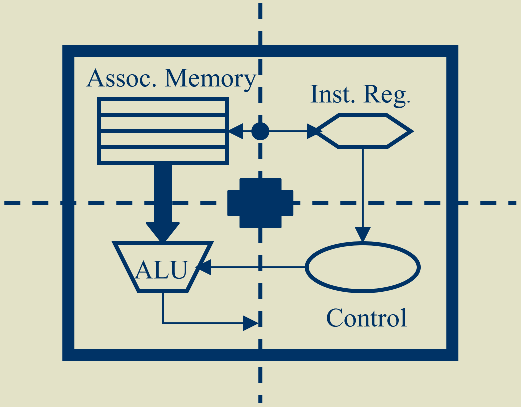
- Locality of operation dependent on amount of logic and state that can be accessed round-trip within a single clock cycle
- Define τ as ratio of number of elements (e.g., gates, transistors) per chip to the number of elements accessible within a single clock cycle
- Not just a speed of light issue
- Also involves propagation through sequence of elements
- When I was an undergrad, $\tau = 1$
- Today, $\tau < 10$
- For SFQ at 100 GHz, $100 < \tau < 1000$
- At nano-scale, $\tau > 100,000$

Continuum Computer Architecture Fundamental Concepts

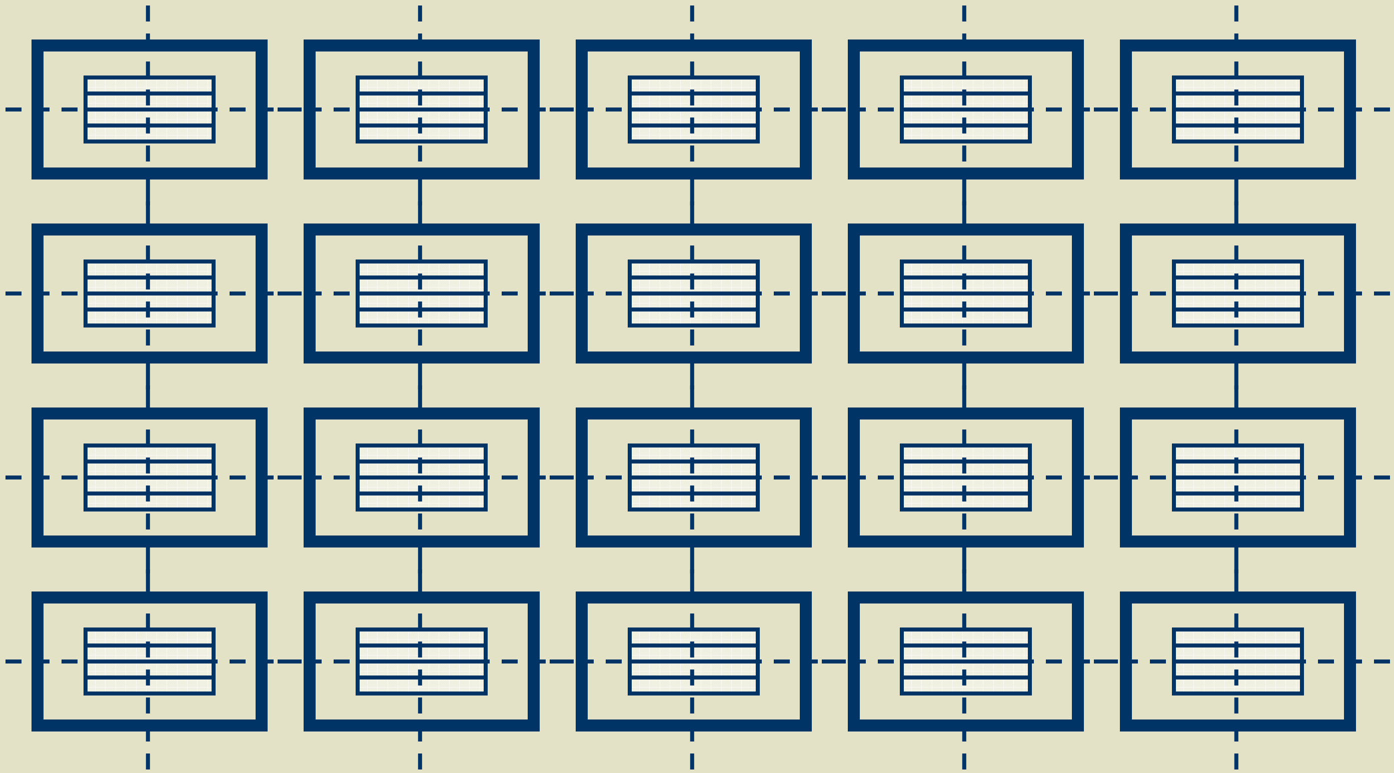
- General purpose cellular architecture
- Global fine-grain cellular structure (*Simultac*)
  - 2.5 or 3-D mesh
  - Blocks interact nearest neighbor
- Merge functionality into a single <u>simple</u> block (*Fonton*)
  - Synergism among fontons yields emergent global behavior of general parallel computing model
  - Communications nearest neighbor
  - Memory, all register with associative tags for names and type specification
  - Data/instruction structures distributed across fontons – virtual addressing
  - Logic performs basic operation on local data
- Dynamic adaptive and distributed resource management

CCA Structure: *Fonton*

- Small block of fully associative tagged memory/registers
- Basic logical and arithmetic unit
- Instruction register directs control to set data paths
- Nearest neighbor communications with switching
- PRECISE binary instruction set compressed encoding



CCA Structure: Distributed Associative Memory



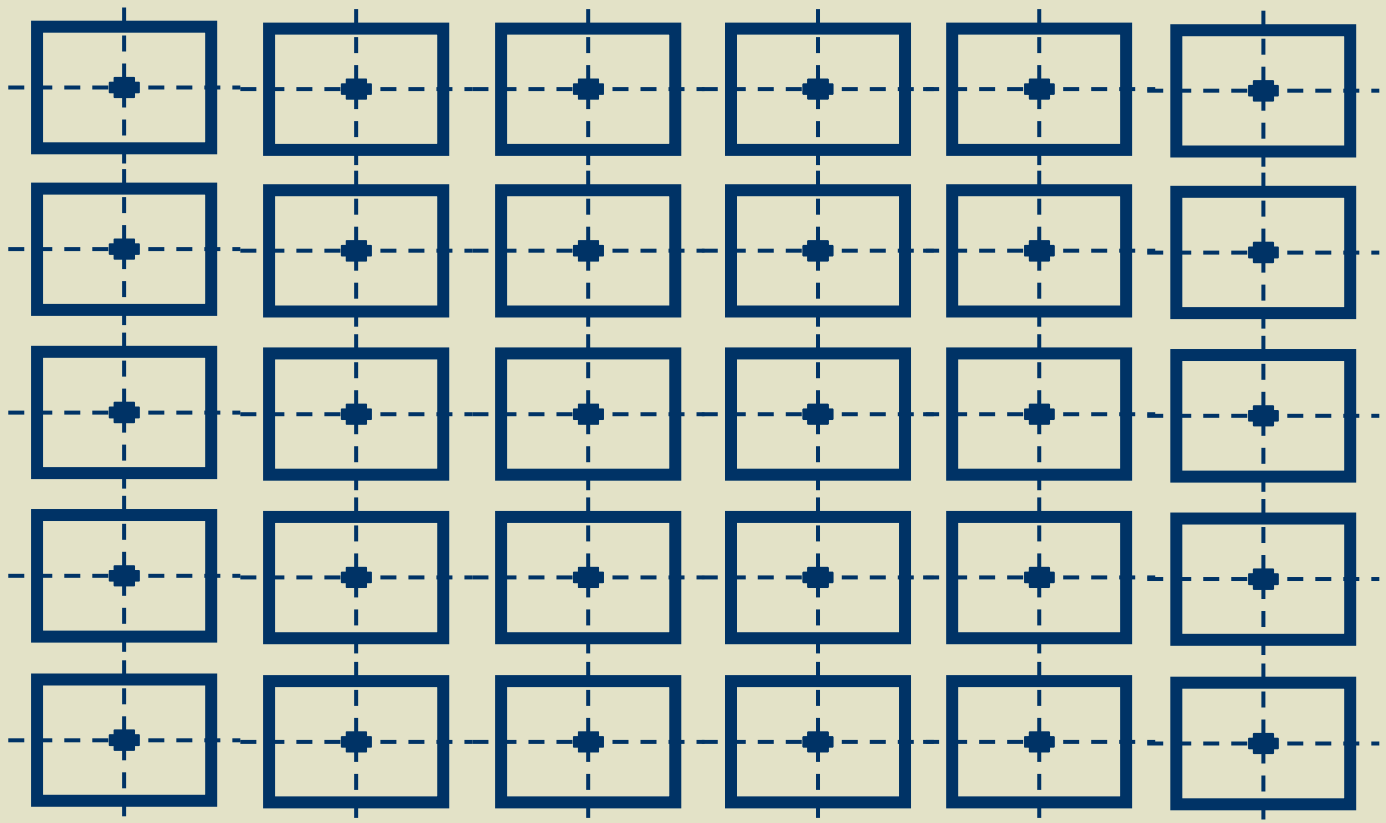
Data Organization and Management

- Three classes of data ensembles
  - scalar values; stored in a single fonton
  - complex; records of some small number of values, which if small can fit on a single fonton, or in adjacent fontons
  - compound; distributed across fontons and coordinated by links
- Data migration
  - objects are copied to adjacent fontons
  - copying exploits fine grain data parallelism, even for irregular data structures
  - objects may transit by means of wormhole routing
- Data objects are virtual named
  - With tags for associative search and typing

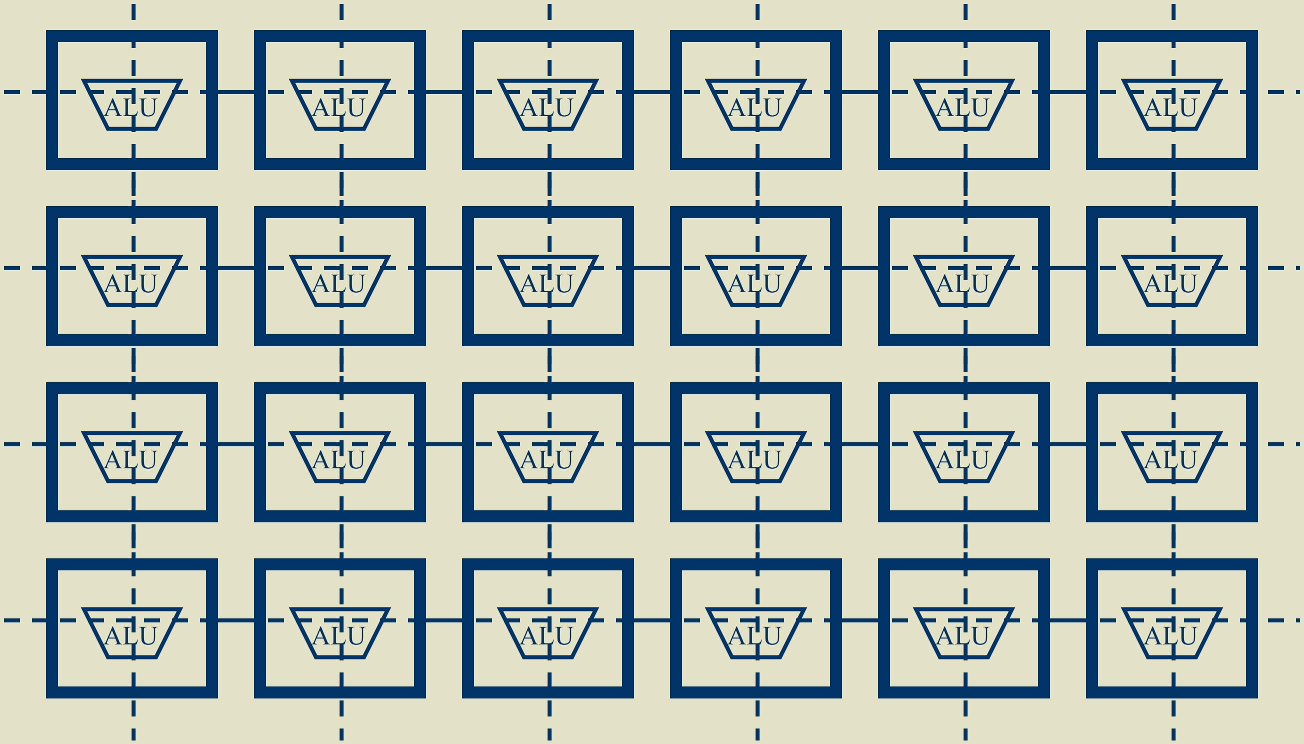
Address Translation

- Distributed associative mapping
- Data carries virtual tags
- Requests are directed in 2D/3D
- Requests search sequence of fontons for reference
- Reference match either locates operand or a new directive (“crumbs”).
- Reference tree of links using virtual links and directors
- Objects can be nailed down
- Directory table provides global naming
- “Shock-wave” searches for unknown positions

CCA Structure: Mesh Network



CCA Structure: Gate Array



Instruction Streams and Execution

- Instruction streams are just another data structure
- Can be static in place or migrate through successive fontons
- They are tagged as instructions and carry their target environment id for unique instantiation
- Data can move across instruction sequence, synthesizing the equivalent of a programmable pipeline
- Instruction sequence can move across a stored data sequence synthesizing the equivalent of vector execution

Principal Mechanisms

- Virtual address translation and object locating
- Cellular client-server relationship
- Resource allocation (load balancing) by diffusion (adjacent copying)
- Data objects & structures distributed across fields of fontons
- Vectors are mobile data structures (*workhole routing* type I)
  - Can move across software pipelines with separate instructions in each fonton pipeline stage
- Instruction threads are mobile data structures
  - Can move across ephemeral vector register with separate datum in each fonton of register bank ("Parcels")
- "*Futures*" coordinate time synchronization and space utilization
- Irregular data structure pointers direct n-furcating
- Fault isolation (reconfigurable?, at least On-Off)
- Asynchronous interfaces

Summary

- Extremes in technology scale and clock rate will demand innovations in architecture
- In the limit, locality of action will dominate operation and bandwidth of storage access will determine sustained performance
- Fine grain cellular structures provide highest storage access bandwidths and highest peak performance
- Continuum Computer Architecture (CCA) exploits fine grain cellular structure for general purpose parallel processing
- CCA may provide convergent architecture for nano-scale and ultra high clock rate technologies at the end of Moore’s Law