Towards Petaflop Architectures

Application Emulators and Simulation

University of Maryland
Syracuse University
Rutgers University
University of California, Santa Barbara
Architectural Design using Application Emulators

• Characterize performance of important applications on future architectures
  – Assumptions
    • application belongs to a targeted application class
    • behavior of processor architecture/compiler interaction has been characterized for application class
      – project behavior of processor pipeline, cache using empirical characterizations from current architectures
      – assume that programmer and compiler will use known optimizations
      – not interested in “dusty deck” performance predictions
Driving Applications

- Sensor data and land cover characterization
- Visualization and analysis of very large microscopy datasets
- Bay and estuary simulation
- Circuit simulation
- Stealth aircraft design
- MSTAR
- Combustion simulation
- Data mining
- Data cube
Active Data Repository Design

Objectives

• Integrate and overlap a wide range of user-defined operations, in particular, order-independent operations with the basic retrieval functions

• **Support optimized associative access and processing of multiresolution and irregular persistent data structures**

• Targets *parallel and distributed architectures* that have been configured to support high I/O rates

• Applications -- *Titan*: Satellite sensor data; Virtual Microscope Server, Bay and Estuary Simulation
Sensor data processing -- data intensive applications

- Data products generated from disk based datasets
  - datasets are usually irregular
    - indexed by spatial location (position on earth, position of microscope stage)
- Spatial query used to specify iterator
  - computation carried out on data obtained from spatial query
  - computation aggregates data - resulting data product size significantly smaller than results of range query
Application Emulators

- Parameterized programs designed to mimic application computation and data movement patterns
- Focus is on memory hierarchy, computational details are abstracted
  - generally also abstract L1 and L2 cache
- Coarse grained, executable description of patterns of data movement and computation
- Generates type of dynamic task graph
Spatial Irregularity
AVHRR Level 1B NOAA-7 Satellite 16x16 IFOV blocks.
Example Projection Query

Output grid onto which interpolation is carried out

Specify portion of raw sensor data corresponding to some search criterion
Application Emulators

• Computation and data movement can be decomposed into a sequence of phases or *epochs* (loosely synchronous computational pattern)

• *Demand driven generation (POEMS terms)*
  – iterator specifies a number of independent computations
  – dependencies can exist within each iteration and between phases
Application Emulators

- Data product may itself be used or may be used as part of a more complex calculation
  - land cover classification
  - data assimilation
  - bay and estuary simulation
  - virtual microscope -- morphometry, 3D reconstruction
Current Application Emulators

• Scientific I/O intensive (Application Domains Supported by Active Data Repository)
  – Titan
    • Satellite data processing
  – Pathfinder
    • Satellite data processing
  – Virtual Microscope
    • Microscope image database server
    • data server (multiple simultaneous queries)
Current Applications Emulators

• Scientific irregular
  – Sparse Gaussian
  – Fast multipole method (Vortex dynamics)

• Database
  – Data Cube
  – Data Mining
  – External Sort
Simulators

- Suite of simulators - varying degrees of fidelity
- All simulators abstract pipelining
- Howsim - detailed architectural simulation using empirical and published device characteristics
- Petasim - rough analysis to account for costs of moving data between memory hierarchy levels
- Block level data driven simulators -
  - Data driven simulators -- Fastsim, Gigasim, Dumbsim -- trace chunks of data through retrieval, data movements, processing and storage (or output to network)
- Emulators coupled to simulators by incremental generation and consumption of work flow graphs
Coupling to Simulators
(Work flow Graphs)

Titan
- DISK READ
  - MESSAGE SEND
  - MESSAGE SEND
  - COMPUTE
  - MESSAGE RECEIVE
  - MESSAGE RECEIVE
  - COMPUTE

Pathfinder
- I/O nodes
  - DISK READ
  - MESSAGE SEND
  - MESSAGE SEND
  - MESSAGE RECEIVE
  - MESSAGE RECEIVE
  - COMPUTE
  - COMPUTE

Virtual Microscope
- DISK READ
  - COMPUTE
  - SEND
Compiler Support for Generating Application Emulators

- Used to generate work flow description used in petasim and block level simulators.
- Programmer uses knowledge of application domain to write computational component of application emulator.
  - Compiler generates code that, at runtime, produces work flow description.
    - Estimate computational costs and amount of data communicated between nodes in workflow graphs.
  - High level directives used to control granularity of work flow description.
General Overview of Performance Prediction Framework

**Applications**

- Petasim
  - *Initial Design*
  - *Simple hardware*

- Gigasim
  - *Large Scale Machines*
  - *Simple hardware*

- Fastsim
  - *Simple hardware*

- Dumbsim

**Emulators**

- Howsim
  - *Refinement*
  - *Detailed Hardware*

**Machine Configuration**

- *Initial Design*
- *Simple hardware*

- *Refinement*
- *Detailed Hardware*

**Applications**

- *Abstract, simple*
- *Hand-coded*

**Emulators**

- *Automated*

**Work flow graph, Dependency Graph Generation Tools**

**Execution Specification**
Evaluation of Candidate Architecture: Active Disks

- Disks with embedded CPU and memory
- Application-specific code executes on disk
- Processing partitioned: disk and host
  - Active disk performs bulk of the processing
  - Host coordinates/schedules/combines

- #CPUs increase with #Disks
- Processing power evolves with disks
Active Disk Architecture

- Processing power scales naturally with storage capacity
- Processing power evolves with storage

- Restructure apps
- Disk-resident code
  - bulk processing
  - disklet
- Host-resident code
  - coordination
  - communication
  - combination

Serial link

M  P

M  P

M  P

M  P
Experiments

• Compared algorithm-architecture combinations
  – current and future configurations

• Evaluated scalability
  – configuration: 4-32 disks
  – dataset size

• Evaluated impact of host upgrades