Q: What Does the Future Hold for “Parallel” Languages?

A: Examine History

Frederica Darema
Senior Science and Technology Advisor
Director, Next Generation Software Program
NSF
Application and Platform Directions

- Applications are becoming:
  - more complex, multi-modal aspects of the application system, multiple developers, multiple languages, application modules distributed, data distributed

- Platforms
  - *in the past*: Vector Processors, SIMD MPPs, Distributed Memory MPs, and Shared Memory MPs
  - *are evolving into*:
    - Distributed Platforms: GRIDS and GiBS (Grids-in-a-Box)
      - Heterogeneous Computers and Networks
    - architecture (compute & network) and node power (supernodes, PCs)
    - Multiple levels of memory hierarchy, with
      - Latencies: variable (internode, intranode)
      - Bandwidths: different for different links; different based on traffic
The Emergence of Grid Computing in the 90's
coordinated problem solving
on dynamic and heterogeneous resource assemblies

Example: “Telescience Grid”, Courtesy of Ellisman & Berman /UCSD&NPACI
Adaptive Software Project for Fluids and Crack Propagation Pipe Workflow

MiniCAD

Model_{t} → Surface Mesher → Surface Mesh_{t} → Generalized Mesher

Mechanical

T_{s}/P_{s} → Fluid/Thermo → Fluid Mesh_{t}

T4 Solid Mesh_{t} → T4→T10

JMesh

T10 Solid Mesh_{t} → Model_{t+1}

Disps_{t} → Client: Crack Initiation → Initial Flaw Params → Crack Insertion → Crack Extension

Viz → Fracture Mechanics → Growth Params_{1}
Programming Parallel Computers

- In ’83 the SPMD Model (private+shared memory machines)
  - Some used it as Data Parallel Model, SPMD more general
- Mid-late 80’s several “message-passing” machines
  - PVM
    - Parallel Computing Forum, IBM Parallel Fortran
- Early 90’s Scalable MPPs:
  - PVM, MPI, and multitude of implementations thereof
  - HPF
- Later also DSM and SMP Clusters
  - Threads, Titanium, Split-C, UPC, GA, Co-Array Fortran, Earth, OpenMP, Charm++, STAPL, ...
## Parallel and Distributed Computing

### Dynamic Analysis Situation

- **Launch Application(s)**

### Platform

<table>
<thead>
<tr>
<th>Platform</th>
<th>Programming Model</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network of Workstations (NOW)</td>
<td>• Message passing</td>
<td>• Inefficient load-balancing</td>
</tr>
<tr>
<td>Symmetric Multiprocessor (SMP)</td>
<td>• Static partition</td>
<td></td>
</tr>
<tr>
<td>Cluster of SMPs</td>
<td>• Shared queue</td>
<td>• Application “re-write” required</td>
</tr>
<tr>
<td></td>
<td>• Dynamic allocation of work</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Message-passing across SMPs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Shared queue within SMP</td>
<td></td>
</tr>
</tbody>
</table>

### Adoptable Systems Infrastructure

- **Distributed Computing Resources**
  - Globus, Legion, Condor, Harness, …

### Distributed Platform

- **MPP**
- **NOW**
- **SP**
The NGS Program develops Technology for integrated feedback & control
Runtime Compiling System (RCS) and Dynamic Application Composition

Dynamic Analysis Situation

Launch Application(s)

Application Model

Application Program

Application Intermediate Representation

Dynamically Link & Execute

Dynamically Link & Execute

Distributed Programming Model

Compiler Front-End

Compiler Back-End

Performance Measurements & Models

Application Components & Frameworks

Distributed Computing Resources

Distributed Platform

Adaptable Computing Systems Infrastructure

MPP
NOW
SP
Models are needed, that

- Will enable the RCS (runtime compiling system) to map applications, without:
  - Requiring detailed resource management specifications by the user
  - Requiring specification of data location and execution
- Advanced concepts of dynamic adaptive resource management
  - Decoupled execution and data location/placement
  - Memory consistency models
  - Multithreaded hierarchical concurrency
- Combination of language and library models / hybrid
- “Active Programming”, and “Active data distribution”
Application Programming Models

• **Applications will increasingly be multi-component**
  - some parts fine-grain, some parts coarse-grain concurrency
  - different algorithms - different sets of processor resources;
    leaving this burden to the user exclusively- not desirable
  - for specific algorithms high-level systems (e.g. Netsolve) provide
    portability and optimized mapping

• **Will continue to pursue models**
  - Shared Memory
  - Message Passing
  - Remote Procedure Call (RPCs)
  - Hybrid and Higher Level Models

• **Embodied in**
  - Library based implementations
  - Language extension based implementations
  - New/Advanced programming models
Programming Environments

- Procedural $\rightarrow$ Model Based
- Programming $\rightarrow$ Composition
- Custom Structures $\rightarrow$ Customizable Structures (patterns, templates)
- Libraries $\rightarrow$ Frameworks $\rightarrow$
  
  **Compositional Systems**
  *(Knowledge Based Systems)*

- Application Composition Frameworks and Interoperability extended to include measurements
- Data Models and Data Management
  - Extend the notion of Data Exchange Standards *(Applications and Measurements)*
The Role of the Applications

- Need to include applications to validate the effectiveness of the new models and environments
- Experience with using the new models in applications and platforms will guide the model evolution
- The effort should be coupled with advances in compiler technology and performance analysis tools
Summary

• Need programming models that
  - Will shorten distributed application development time
  - Make parallel/concurrent/distributed algorithms easier to express
  - Improve portability, application efficiency across platforms

• SPMD still remains useful, ... but there are many challenges and opportunities

• Further advances needed on
  - programming models
  - compilers
  - libraries
  - integrating these software components into an application development and runtime support system,
  - ... and don’t forget performance and efficiency!!!

• NGS fosters research in these directions; and specifically calls for proposals on programming models

• NGS Program [www.cise.nsf.gov](http://www.cise.nsf.gov) (Program Announcements)