Software Architectures for Morphing in Polymorphous Computing Architectures

Dan Campbell, Mark Richards
Georgia Institute of Technology

Dennis Cottel, Randall Judd
USN SPAWAR Systems Center, San Diego

**Development Process**
- Two-stage compile process enables portable performance across PCA architectures

**Morphware Stable Interface Architecture**
- Two intermediate representations
  - Stable API: application code in C/C++ and a stream language such as Brook or Streamit
  - Stable Architecture Abstraction Layer: PCA virtual machine code

**SAPI and SAAL**
- Stable APIs (SAPI)
  - Streamit
  - C/C++
  - Brook
  - Others...
- Virtual Machine API
  - UVM
  - SVM
  - TVM-HAL

**Machine Models**
- Used to optimize VM output for different target platforms
  - Coarse grain mapping of application to target resources
- Describes target platform using common dictionary of virtual resources and attributes
  - Processors
  - Memories
  - Net links

**The Morphware Stable Interface**
- Standard PCA Application Environment
  - Defined by a set of open standards documents
- Based on a virtual machine (VM) abstraction layer with standardized metadata and programming languages

**Morphing in the MSI**
- MSI assumes component-based architecture
  - natural and intuitive boundaries for compilation and run time reconfiguration
  - natural support for multiple SWEPT-variant implementations of units of functionality
- Morphing implies changing ...
  - component implementations in use;
  - resources assigned to components;
  - or both
- Implies a taxonomy of morph types
- Morphing will be implemented at various levels of MSI
  - compiler
  - run time system
  - resource manager

**Machine Models**
- User accesses User level VM for thread code, Stream VM for stream code
- TVM HAL abstracts low level hardware to UVM

**VM Layers**
- User accesses User level VM for thread code, Stream VM for stream code
- TVM HAL abstracts low level hardware to UVM

**Morph Taxonomy**

For more information: www.morphware.org