POEMS

POEMS - Performance Of End-to-End Models of HPC Systems

Research Team

Adve - compilers ---> performance models
Bagrodia - parallel simulators, I/O and network models
Browne - parallel computation, systems modeling
Houstis - performance data base
Lubeck - applications and models
Teller - memory hierarchy models
Vernon - analytic models

6/16/97

POEMS

What is to be modeled?

Workload			
comp	I/O	comm.	application
			operating system
			processor architecture
			network architecture

POEMS

Application Requirements

1. end-to-end models

2. Model Development and Evaluation - 10% rule

3. Confidence in results

4. Accuracy - resolve system architecture alternatives

5. Reuse of Components

POEMS

Technology Requirements

1. Multi-domain models

2. Multi-level models

3. Multi-method evaluation

4. Parallel evaluation

5. Associative Interactions

6. Preservation of behavior patterns across levels of abstraction

7. Automatic synthesis of performance models

8. Associative Interactions

9. Multiple time/space scales

POEMS

POEMS Constituents

1. Specification Language - capture and composition

2. Model Libraries - applications, operating systems, architectures

3. Performance Data Base - Execution Data ----> performance models

4. Model Evaluation - simulators, analytic solvers, actual hardware

5. Model Generation - Programs ----> performance models

7. Interfaces to performance monitors

8. Systems Management Language

POEMS

Validations

Component model validation
existing applications, existing architectures
LANL ASCI applications
GloMo, Quorum, etc

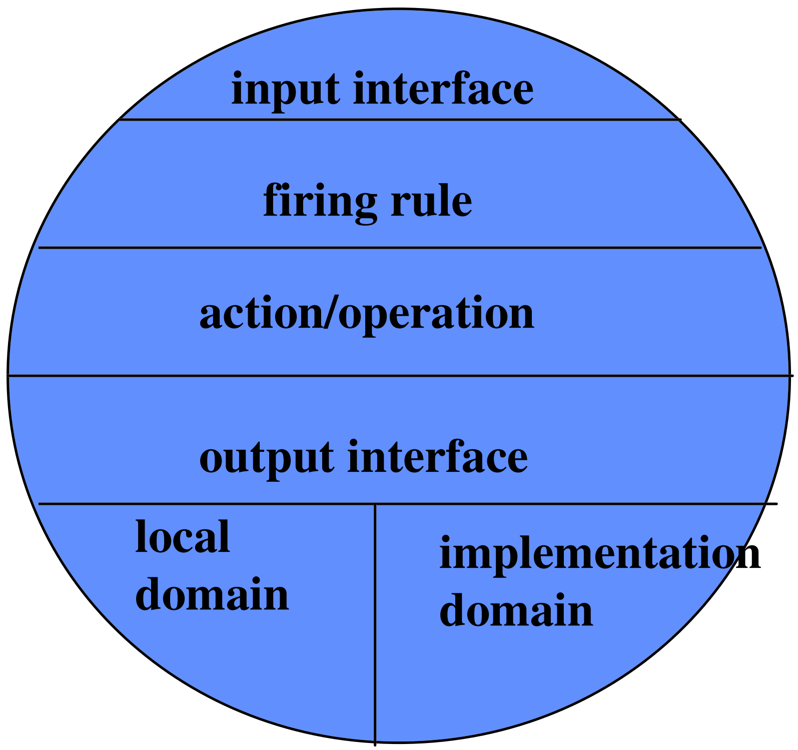
Collaborations
SGI
Cray
IBM

POEMS

Representation Issues - Model of Parallel Computation

Generalized Dependence Graph

Nodes, relationships, transactions



node specification

POEMS

Associative Model of Interaction

1. set of parameters

2. set of attributes

3. profiles - subset of attributes

4. selectors- a template for a conditional expression with slots for attributes or attribute values

5. hyperactive messages - messages which contain an executable action.

6. Parameter mappings

POEMS

Input Interface

(transaction specifications - units of work processed, parameters)
(profile - specifications for what transaction the model can process)

Output Interface

(transaction specifications, parameters)
(selector - specifications for the target model for the transaction)

Hyperactive Transactions - transactions contain executables.

POEMS

Application of Associative Interface

1. Construct input and output interfaces for each model component.

2. Construct transaction and parameter maps.

3. Resolve at each of: compile time, link time and runtime.

4. Invoke mappings as required.

Example Model Specification

cache (transactions, parameters profile)

transactions - instructions = {single, block}
parameters - none
profile - state resolution = {full, stochastic}
- cache type = {direct, set associative)
- cycle time = {value}

POEMS

How is all of this possible?

- **Existing resources**
  - **HPF compilers**
  - **parallel simulators**
  - **analytic solvers**
  - **algorithm analyses**

Collaboration from vendors

- **access to data**
- **access to measurements**

Experienced Investigators