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

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

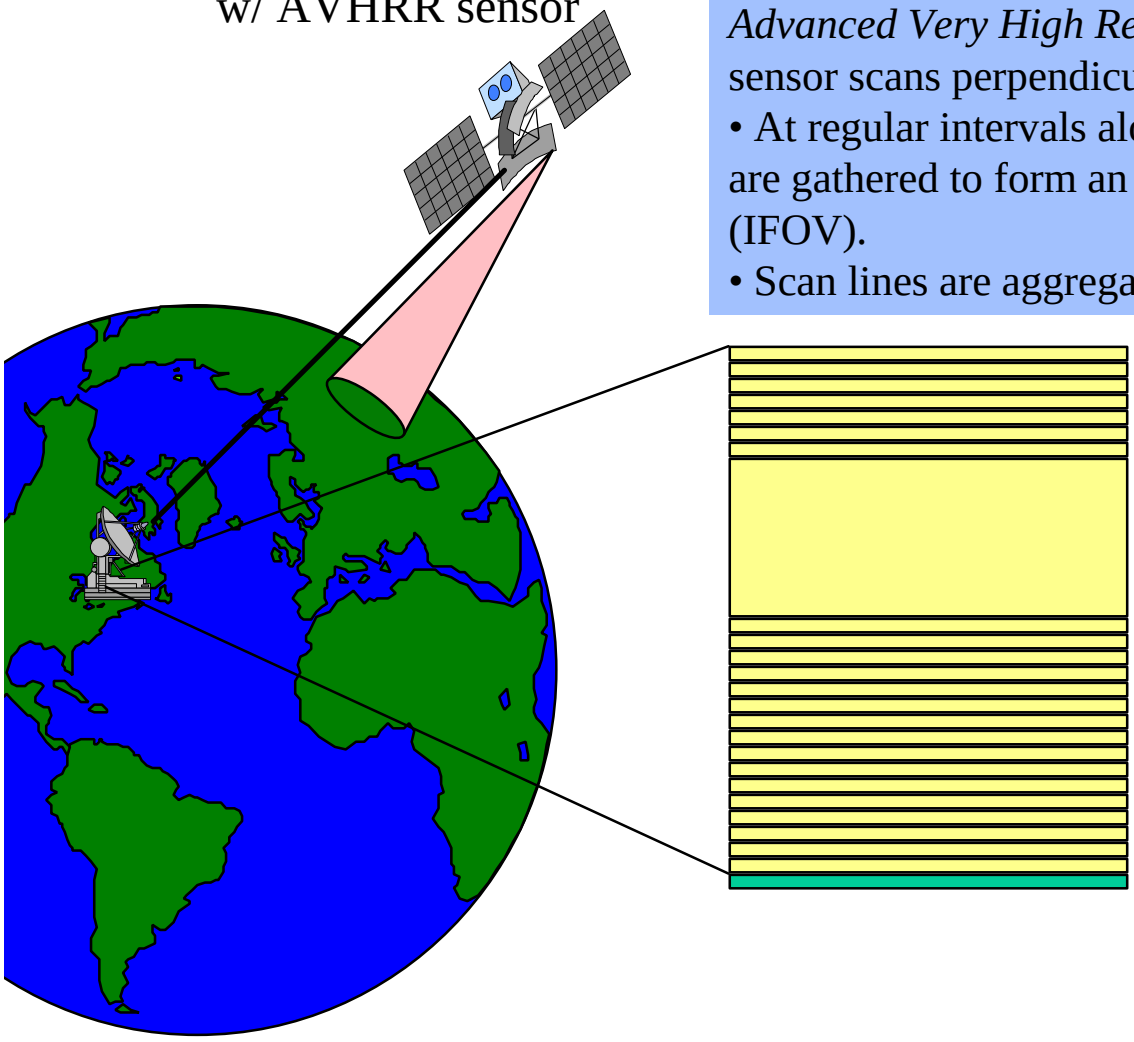
Construction of Application Emulators

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 so that resulting data product size is significantly smaller than results of range query

Processing Remotely Sensed Data

NOAA Tiros-N
w/ AVHRR sensor



AVHRR Level 1 Data

• As the TIROS-N satellite orbits, the *Advanced Very High Resolution Radiometer* (A sensor scans perpendicular to the satellite's trac

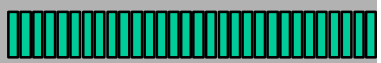
• At regular intervals along a scan line measure are gathered to form an *instantaneous field of v* (IFOV).

• Scan lines are aggregated into Level 1 data se

A single file of *Glo Coverage* (GAC) d represents:

• ~one full earth orbit.
• ~110 minutes.
• ~40 megabytes.
• ~15,000 scan lines.

One scan line is 409



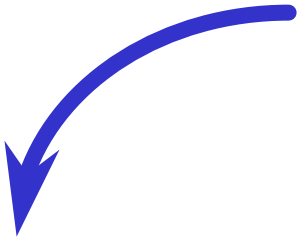
Preparing The Data

Level 0 Data (radiometry)

instrumental correctio

navigation

Level 1 Data (raw data + g



resample

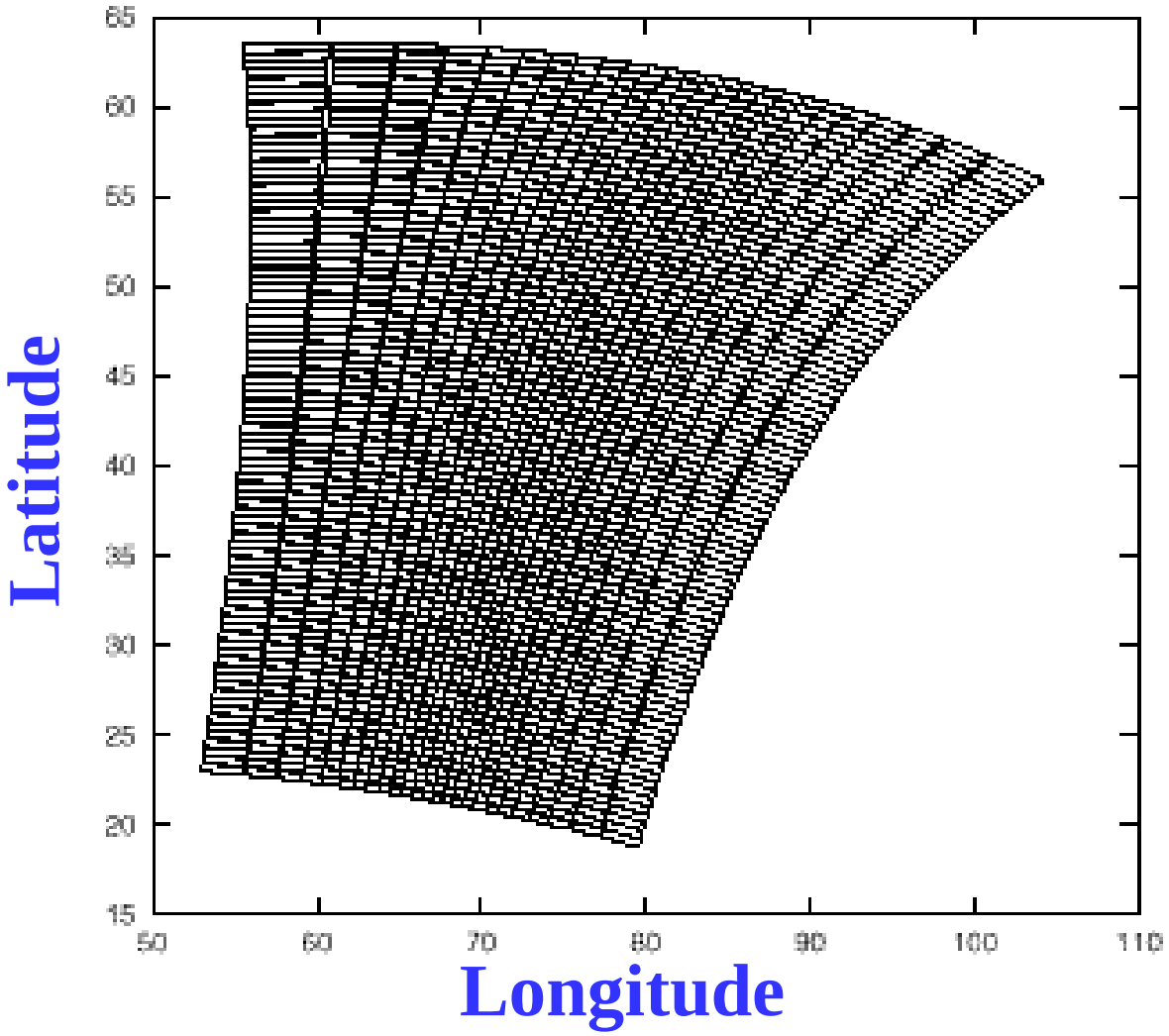
projection

Level 2 Data (2-D grid)

alysts

Spatial Irregularity

VHRR Level 1B NOAA-7 Satellite 16x16 IFOV blc



Application Emulators

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

- iterator specifies a number of independent computations
- dependencies can exist within each iteration and between phases

Data product may itself be used or may be used as par 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

- Titan
  - Satellite data processing
  - peer-to-peer
- Pathfinder
  - Satellite data processing
  - client-server (separate IO and Compute nodes)
- Virtual Microscope
  - Microscope image database server
  - data server (multiple simultaneous queries), peer-to-peer

Current Applications Emulators

Scientific irregular

- Sparse Gaussian
- Fast multipole method (Vortex dynamics)

Database

- Data Cube
- Data Mining
- External Sort

Simulators

Suite of simulators that simulate to varying degrees of fidelit

All simulators abstract pipelining

Howsim carries out detailed architectural simulation using empirical and published device characteristics

Petasim carries out rough analysis that accounts for costs associated with moving data between levels of memory hierarchy

Block level data driven simulators -

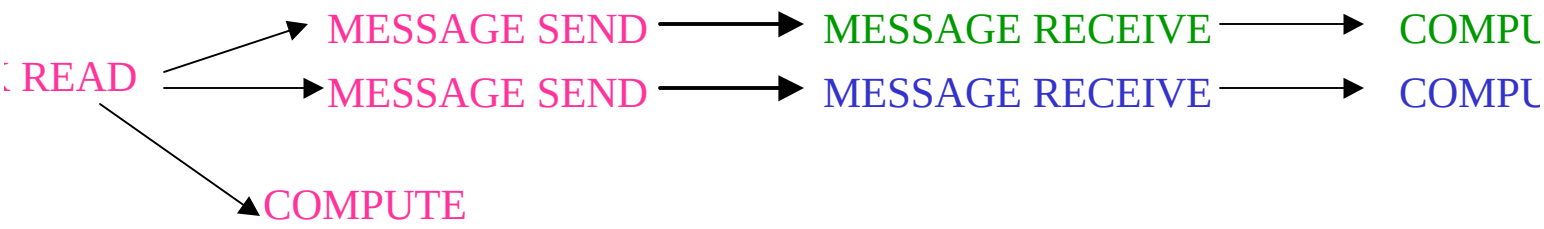
- Data driven simulators -- Fastsim, Gigasim, Dumbsim -- trace chunks of data through retrieval, data movements, processing and storage (or output to network)

Emulators coupled to simulators through incremental generation and consumption of work flow graphs

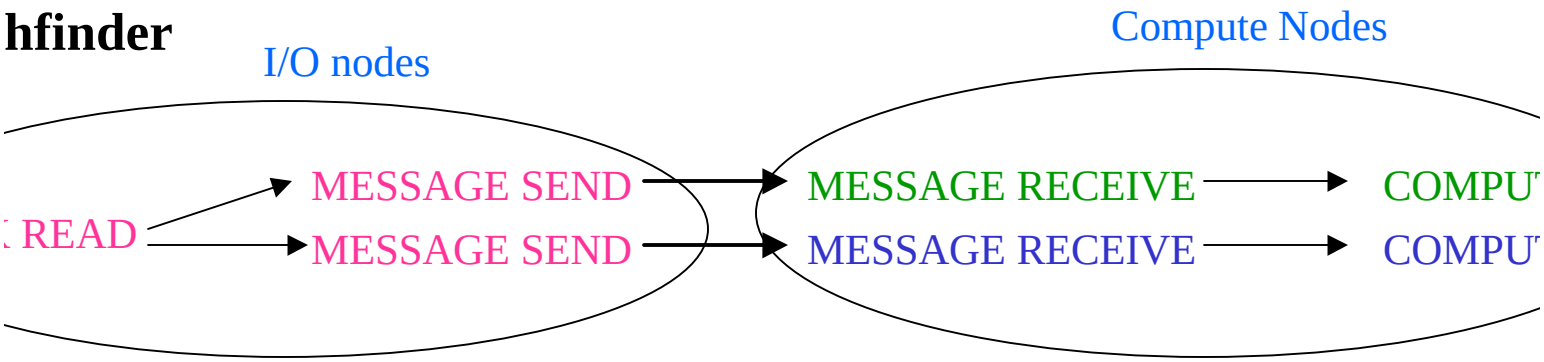
Coupling to Simulators

(Work flow Graphs)

tan



hfinder



tual Microscope

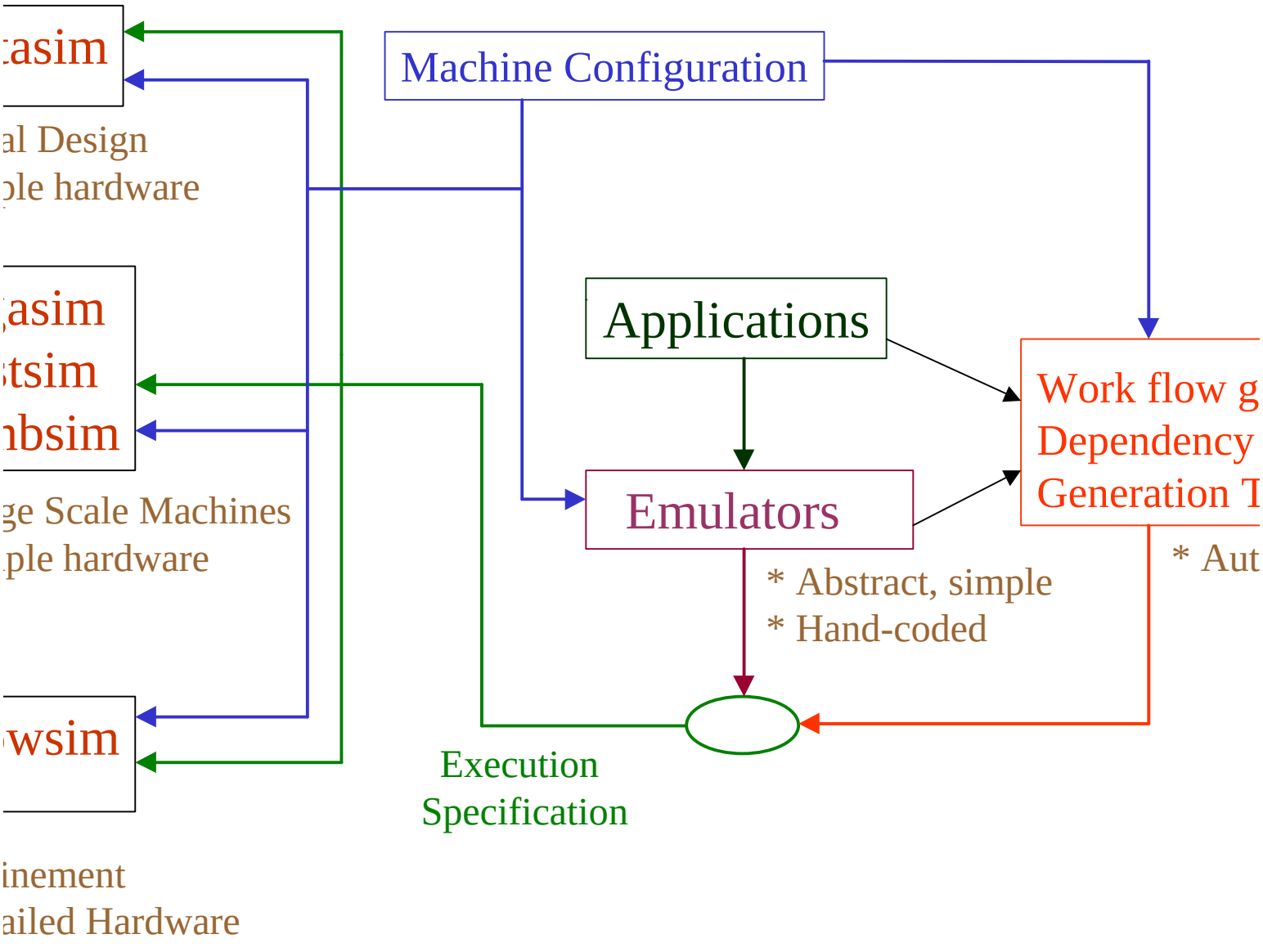
Compiler Support for Generating Application Emulators

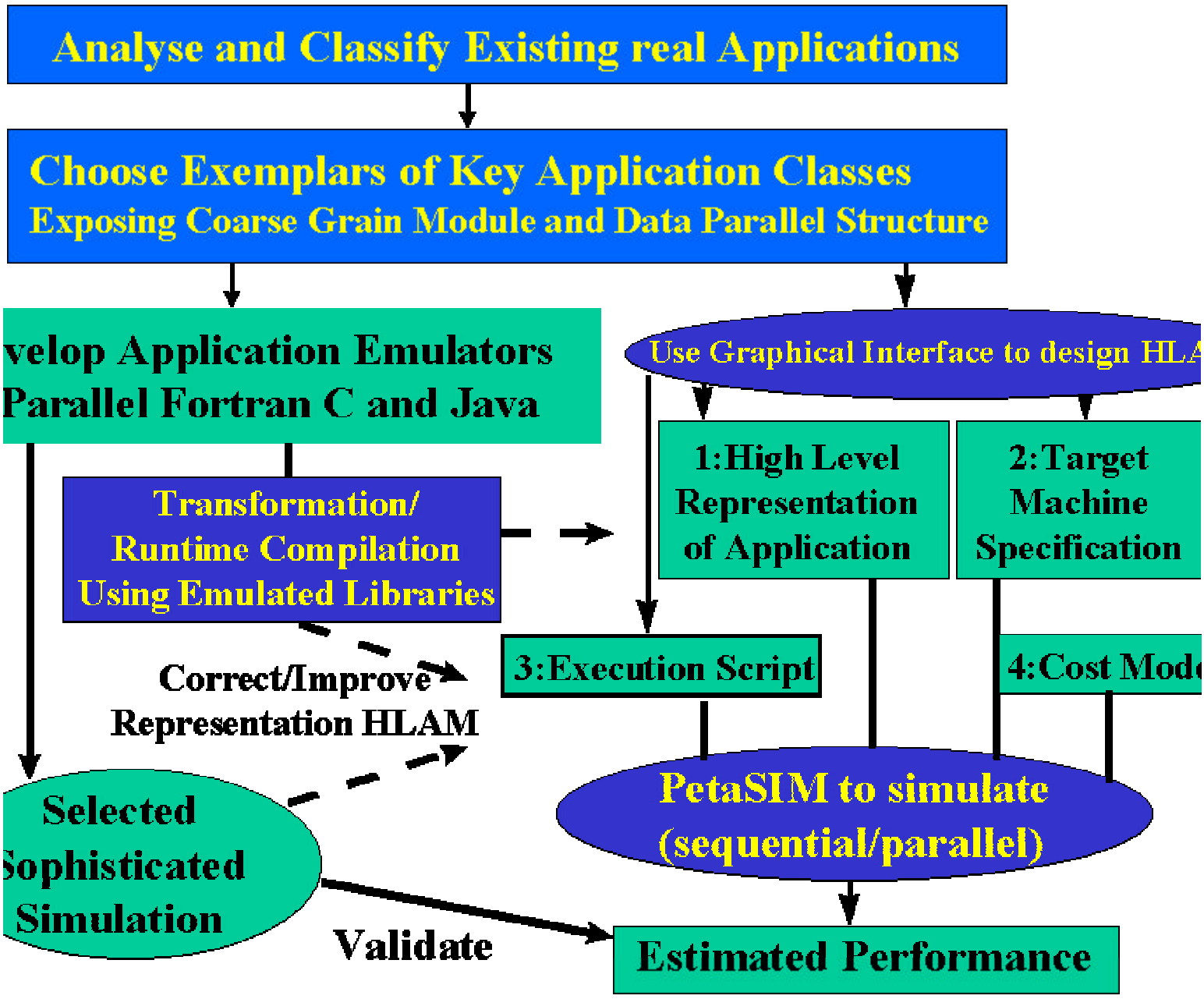
Used to generate work flow description used in petasim and in block level simulators

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




Analyse and Classify Existing real Applications
Choose Exemplars of Key Application Classes
Exposing Coarse Grain Module and Data Parallel Structure
velop Application Emulators
Parallel Fortran C and Java
Use Graphical Interface to design HL
1:High Level Representation of Application
2:Target Machine Specification
Transformation/ Runtime Compilation Using Emulated Libraries
Correct/Improve Representation HLAM
3:Execution Script
4:Cost Mod
Selected Sophisticated Simulation
PetaSIM to simulate (sequential/parallel)
Validate
Estimated Performance

Driving Applications

Sensor data and land cover characterization

Visualization and analysis of very large microscopy datasets

Circuit simulation

Stealth aircraft design

MSTAR

Combustion simulation

Bay and estuary simulation

Data mining

Data cube