Transparent Grid Enablement
Using Transparent Shaping and GRID superscalar

S. Masoud Sadjadi, Juan Carlos Martinez, Luis Atencio, and Tatiana Soldo
Autonomic and Grid Computing Research Lab
School of Computing and Information Sciences, FIU
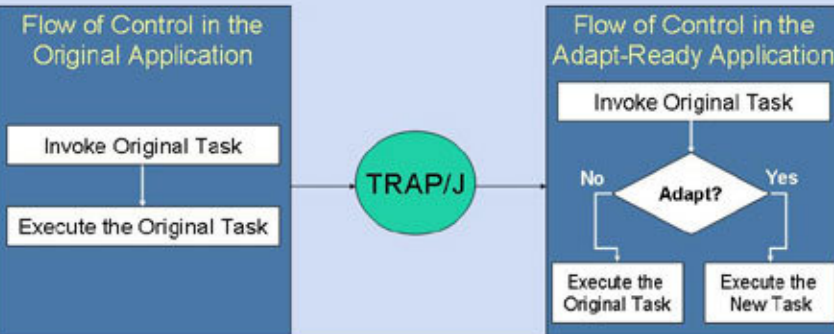Contact: sadjadi@cs.fiu.edu

Rosa M. Badia and **Jorge Ejarque**
Grid Computing and Clusters
Barcelona Supercomputing Center and UPC, Spain
Contact: rosa.m.badia@bsc.es

I. Description and Motivation

- **What is the project trying to do?**
  – Explore techniques and tools to simplify the process of grid enablement

- **Why is it important?**
  – Building, debugging, deploying, and maintaining Grid enabled application is difficult (even when a sequential version of the application exists)

- **What is the expected output?**
  – Methodologies and tools that enable easy implementation of Grid applications.

- **Case Study**
  – Matrix Multiplication
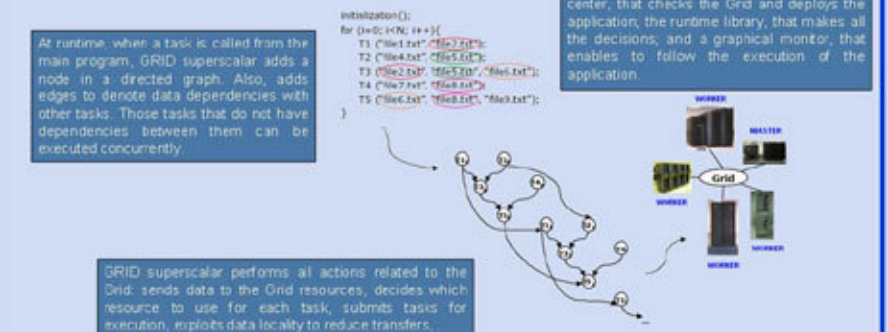 – Hurricane Mitigation Applications

II. Background Information: Transparent Shaping

- Transparent Shaping is a programming model that allows software systems to change their behavior *transparently* at runtime, i.e. without any manual modifications to the original code.

- TRAP/J (Transparent Reflective Aspect Programming in Java) is the realization of this model in Java, which provides dynamic adaptation to existing Java application.

Flow of Control in the Original Application — Invoke Original Task — Execute the Original Task — TRAP/J — Flow of Control in the Adapt-Ready Application — Invoke Original Task — Adapt? No / Yes — Execute the Original Task — Execute the New Task

III. Background Information: GRID Superscalar

- GRID superscalar is a new programming paradigm for GRID enabling applications. With GRID superscalar a sequential application, composed of tasks of certain granularity, is automatically converted into a parallel application where the tasks are executed in different servers of a computational GRID.

IV. Activity Diagram: Integration Approach with Matrix Multiplication Case Study

① Original Application

Scientist

A scientist will develop the business logic of the application using a sequential code (e.g., a matrix multiplication algorithm).

②

A computer expert will identify sections of the code that can be parallelized.

Coarser-Grain Parallelism: Adaptive code for maximum parallelism of 4.

Finer-Grain Parallelism: Adaptive code for maximum parallelism of 9.

Computer Expert

A computer expert will develop adaptive code with different parallelism granularity.

TRAP/J → Adapt-Ready Application

GRID superscalar → Grid-Enabled Application

Merge → Grid-Enabled Adapt-Ready Application → Configuration

System Administrator

A system administrator will configure the application to use the appropriate parallel algorithm based on the availability of resources (e.g., number of available nodes).

Maximum parallelism is 4

OR

Maximum parallelism is 9

Development Time | Compile Time | startup Time | Runtime

Time