

## PROJECT INTRODUCTION

### Objectives

To support execution of different types of applications in a Grid environment.

### Project Investigator / Manager

Dr. Terence Hung  
Institute of High Performance Computing  
terence@ihpc.a-star.edu.sg

### Period of Project

1 Dec 2004 – 30 Nov 2006

### Website

<http://www.ihpc.a-star.edu.sg>

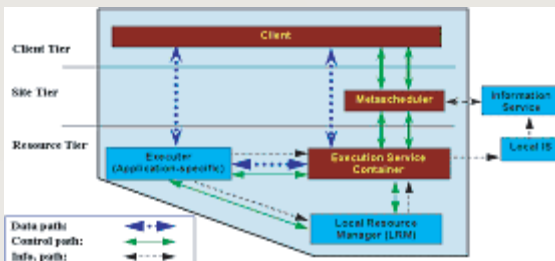
### Abstract

Execution of many applications on a pool of dynamic and heterogeneous resources requires effective and efficient management. Execution Management System (EMS) is a set of Web services developed to fulfil the above requirements. Its functions are scheduling and executing jobs across a large number of Grid resources and at the same time achieve high performance for different applications.

## PROJECT DETAILS

### Description

In order to support execution of many applications running at the same time on a large number of Grid resources, the EMS is structured into three tiers, i.e., client tier, site tier, and resource tier (see figure). Cross-tier communication can happen across the Internet.



Architecture of EMS

On the resource tier, every resource is equipped with a Grid service called Execution Service Container (ESC). It prepares environment for arriving jobs, retrieves input data of jobs, manages execution of concurrent jobs, and stages out output data. The ESC adopts standard specifications for job submission and control, i.e., Distributed Resource Management Application API (DRMAA) and Resource Specification Language (RSL-2).

Another Grid service called Metascheduler (MS) locates on the site tier and works across many Grid resources. It receives jobs from clients and schedules these jobs to a set of ESCs running on the Grid resources. MS also adopts DRMAA and RSL specification as ESC. In order to hide complexity of the Grid, MS contains a set of components such as resource monitor (retrieves information from Information Service), matchmaker (matches jobs with suitable resources), job dispatcher and job monitor (coordinate with ESCs).

EMS has been developed using GT3 (OGSI) and GT4 (WSRF) and supported execution of many scientific applications such as mesh processing, remote visualization, modeling and simulations. Further, it has been integrated with Science & Engineering Grid portal to form a Problem Solving Environment. Currently, EMS is being upgraded with new features such as resource reservation, QoS negotiation and guarantee, and Local Resource Manager integration.