# Galaxy Design Document

## 0 Terminology

* Job

A job should at least contain an executable binary with its command line arguments which can run in the console. A job should also contain its resource files if it has.

* Job Graph

A job graph is composed of one or more jobs and the dependencies between the jobs.

## 1 Design Goals

Galaxy is a distributed execution environment to help users run their jobs on a machine cluster. The design goals of Galaxy are as follows:

* User can designate the affinity of the job
* The environment can show the status of the job to the user
* Failover
* The environment should have a language to describe the job graph
* The environment should have less constraint on the type of the executable binary (unmanaged, managed, perl script, etc.)

## 2 Architecture



**Picture 1. The architecture of Galaxy**

We use .NET Remoting to implement the main logic of Galaxy. The main reason that it is more natural and easier to use remote objects than SOCKET to achieve our design goals(**Section 1**).

### 2.1 Job Manager

Job manager receives the job graph from the client. Then it will be responsible for scheduling and monitoring the jobs and make sure that the jobs will be finished successfully except that:

* There are not enough resource (machines, memories, etc.) after we have tried MAXIMUM\_TRY times
* The job is failed after we have run it MAXIMUM\_TRY times

Note that it will not be strange that the some outputs of a job graph are left in the Galaxy when the job graph is finished. If we simply leave all the local outputs here, they will make the disks of the process node full of garbage. Currently, the users are responsible for deleting the useless data and copying the useful data back. Otherwise, Galaxy will delete all the data after some time.

### 2.2 Process Node

The jobs are run on the process node. A machine can host one or more process nodes. The process node can:

* Run the job which the job manager has assigned to it
* Tell the status of the job to the job manager

### 2.3 The Client Libraries

The client libraries are very helpful for users to use Galaxy. The client libraries should at least include:

* The libraries to construct the job graph

Although we have a language to construct the job graph, sometimes it is difficult for us to build a complex one. The advantage of these libraries is that users can use codes (e.g. for statement) to get a very complex job graph.

* The interfaces exposed by job manager
* The interfaces exposed by process node

Because most of the codes are managed, unmanaged c++ or other language which isn’t not under .NET Framework cannot use it. So for these languages, we expose some COM objects from Galaxy and some libraries are built on these COM objects.

### 2.4 Job Status Viewer

When users submit the job graph to Galaxy, they can view the status of each job from job status viewer. It can supply the following information to the users:

* The status of the job
* The process node which the job is running on
* The location of the output of the job

## 3 Design Details

## Document History

|  |  |  |
| --- | --- | --- |
| **Alias** | **Description** | **Date** |
| guomxin | Create the document | 2006-10-23 |