IceGrid

IceGrid is the location and activation service for Ice applications. For the purposes of this discussion, we can loosely define *grid computing* as the use of a network of relatively inexpensive computers to perform the computational tasks that once required costly "big iron." Developers familiar with distributed computing technologies may not consider the notion of grid computing to be particularly revolutionary; after all, distributed applications have been running on networks for years, and the definition of grid computing is sufficiently vague that practically any server environment could be considered a "grid."

One possible grid configuration is a homogeneous collection of computers running identical programs. Each computer in the grid is essentially a clone of the others, and all are equally capable of handling a task. As a developer, you need to write the code that runs on the grid computers, and Ice is ideally suited as the infrastructure that enables the components of a grid application to communicate with one another. However, writing the application code is just the first piece of the puzzle. Many other challenges remain:

- How do I install and update this application on all of the computers in the grid?
- How do I keep track of the servers running on the grid?
- How do I distribute the load across all the computers?
- How do I migrate a server from one computer to another one?
- How can I quickly add a new computer to the grid?

Of course, these are issues faced by most distributed applications. As you learn more about IceGrid's capabilities, you will discover that it offers solutions to these challenges. To get you started, we have summarized IceGrid's feature set below:

- Location service
  As an implementation of an Ice location service, IceGrid enables clients to bind indirectly to their servers, making applications more flexible and resilient to changing requirements.

- On-demand server activation
  Starting an Ice server process is called *server activation*. IceGrid can be given responsibility for activating a server on demand, that is, when a client attempts to access an object hosted by the server. Activation usually occurs as a side effect of indirect binding, and is completely transparent to the client.

- Application distribution
  IceGrid provides a convenient way to distribute your application to a set of computers, without the need for a shared file system or complicated scripts. Simply configure an IcePatch2 server and let IceGrid download the necessary files and keep them synchronized.

- Replication and load balancing
  IceGrid supports replication by grouping the object adapters of several servers into a single virtual object adapter. During indirect binding, a client can be bound to an endpoint of any of these adapters. Furthermore, IceGrid monitors the load on each computer and can use that information to decide which of the endpoints to return to a client.

- Sessions and resource allocation
  An IceGrid client establishes a session in order to allocate a resource such as an object or a server. IceGrid prevents other clients from using the resource until the client releases it or the session expires. Sessions enhance security through the use of an authentication mechanism that can be integrated with a Glacier2 router.

- Automatic failover
  Ice supports automatic retry and failover in any proxy that contains multiple endpoints. When combined with IceGrid's support for replication and load balancing, automatic failover means that a failed request results in a client transparently retrying the request on the next endpoint with the lowest load.

- Dynamic queries
  In addition to transparent binding, applications can interact directly with IceGrid to locate objects in a variety of ways.

- Status monitoring
  IceGrid supports Slice interfaces that allow applications to monitor its activities and receive notifications about significant events, enabling the development of custom tools or the integration of IceGrid status events into an existing management framework.

- Administration
  IceGrid includes command-line and graphical administration tools. They are available on all supported platforms and allow you to start, stop, monitor, and reconfigure any server managed by IceGrid.

- Deployment
  Using XML files, you can describe the servers to be deployed on each computer. Templates simplify the description of identical servers.

- Database Independence
  By default, IceGrid uses a Freeze database to store its state. However, you can configure IceGrid to use a different database, such as MySQL (among others).

As grid computing enters the mainstream and compute servers become commodities, users expect more value from their applications. IceGrid, in cooperation with the Ice run time, relieves you of these low-level tasks to accelerate the construction and simplify the administration of your distributed applications.

Topics

- IceGrid Architecture
- Getting Started with IceGrid
- Using IceGrid Deployment
- Well-Known Objects
- IceGrid Templates
- IceBox Integration with IceGrid
- Object Adapter Replication
- Load Balancing
- Resource Allocation using IceGrid Sessions
- Registry Replication
- Application Distribution
- IceGrid Administrative Sessions
- Glacier2 Integration with IceGrid
- IceGrid XML Reference
- Using Descriptor Variables and Parameters
- IceGrid Property Set Semantics
- IceGrid XML Features
- IceGrid Server Reference
- IceGrid and the Administrative Facility
- Securing IceGrid
- IceGrid Administrative Utilities
- IceGrid Server Activation
- IceGrid Troubleshooting