Xgrid

The simple solution for distributed computing.

Features

Distributed computing—anywhere

- Clustered computational power of underutilized computers in your organization or around the Internet
- Ability to create complex workflows that specify job and task dependencies
- Prioritization of agents based on available resources and capabilities required by the task

Flexible management

- Support for Kerberos, LDAP, XML, BEEP, and other open standards for easy administration and improved security
- Support for Bonjour and DNS lookups for discovery of Xgrid services
- Compatibility with wide range of unmodified command-line utilities and MPI programs
- Support for command-line and graphical applications, allowing local and remote users to submit and manage jobs

Enterprise-class capabilities

- Access controls for secure distributed computing across formal and ad hoc networks
- Support for hundreds of nodes, thousands of tasks, and gigabytes of data per job*
- Support for shared and clustered file systems, such as NFS and Xsan
- Fault-tolerance features that protect against failed nodes and system reboots
- Integration with Cocoa applications to speed up CPU-intensive calculations

Apple's Xgrid technology makes it easy to turn an ad hoc group of Mac systems into a low-cost supercomputer. Leveraging the power of Mac OS X Server, Xgrid is an ideal distributed computing platform for individual researchers, specialized collaborators, and application developers. For example:

- Scientists can search biological databases on a cluster of Xserve systems.
- Engineers can perform finite element analyses on their workgroup's desktops.
- Animators can render images using Mac systems across multiple dispersed corporate locations.
- Research teams can enlist colleagues and interested laypeople in Internet-scale volunteer grids to perform long-running scientific calculations.
- Anyone needing to perform CPU-intensive calculations can run a single job across multiple computers at once, dramatically improving throughput and responsiveness.

Xgrid functionality is integrated into every copy of Mac OS X and Mac OS X Server. That means system administrators can quickly enable Xgrid on Mac systems throughout their company, turning idle CPU cycles into a productive cluster at no incremental cost.

Xgrid was designed from the ground up for large-scale clusters. It has been tested successfully in configurations of up to 128 agents, 20,000 queued jobs (or 100,000 tasks per job), 2GB submitted data per job, 1GB results per task, and a total of 10GB aggregate results per job.

Unlike many supercomputing solutions, Xgrid was expressly developed for use with today's rich graphical applications. A built-in Cocoa API enables developers to write new Xgrid-aware applications or to update their existing software for use with Xgrid. Many command-line programs can use Xgrid right out of the box, without any changes whatsoever.

Scoreboard, a new feature in Xgrid 2, brings tremendous intelligence to the distribution of tasks, making the most of available processing power. Much more than a batch processing system, Scoreboard prioritizes jobs and sends them to Xgrid nodes with the required features or capabilities. Since each node reports its configuration to the controller, Scoreboard knows which available agents—for example—have a 64-bit Intel processor with more than 4GB of RAM and are connected to an Xsan network. In this way, Scoreboard makes sure that jobs are sent first to those agents that can handle the work most efficiently.

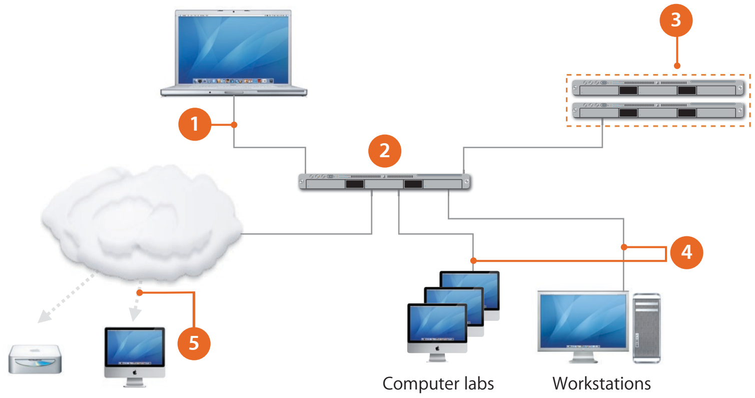
Xgrid demonstrates Apple's ongoing commitment to high-performance computing, which is also reflected in Xserve servers, the Apple Workgroup Cluster, and Shark performance tools. Combined with the ever-increasing array of third-party hardware interconnects, distributed resource managers, Fortran compilers, and other high-performance tools, Mac OS X with Xgrid is the ideal platform for researchers who desire maximum performance with minimum hassle.

The Xgrid protocol
The various components of Xgrid communicate using a lightweight protocol built on top of BEEP, the Blocks Extensible Exchange Protocol. BEEP is an IETF standard analogous to HTTP, but designed for two-way multiplexed communication, particularly for peer-to-peer networks. BEEP uses XML profiles to define multiple channels across a single socket, reducing negotiations and network resource usage. For maximum flexibility, Xgrid shares information using simple XML property lists on top of BEEP.

Three-Tier Architecture

Xgrid leverages the flexibility of a standards-based, three-tier architecture to simplify the distribution of complicated tasks, whether they're running on Mac systems across the room or around the world. Its user clients, grid controllers, and computational agents work together to streamline the process of assembling nodes, submitting jobs, and retrieving results.

Flexible Three-Tier Architecture



1 **Detachable clients.** Submit jobs—wait for or be notified of results.

2 **Controller.** Splits job into tasks, resubmits failures, retrieves results.

3 **Full-time agents.** Dedicated to running Xgrid tasks.

4 **Part-time agents.** Run Xgrid tasks when users are not active.

5 **Internet agents.** Volunteer to help with Internet-scale calculations.

Xcode
Xcode is a fast way for developers to create Mac OS X applications and the easiest way to take advantage of new Apple technologies. Xcode brings together the ease of use of Mac OS X, the power of UNIX, and a mix of high-performance development technologies. Its visual modeling and design features make it easier to both develop and maintain applications.

Client

An Xgrid client provides the front end of the Xgrid system. It is responsible for finding a controller, submitting a job, and retrieving the results. Clients can rely on the controller to mediate all job submissions; they do not need to be aware of the actual computational agents.

The simplest possible client is the Xgrid command-line tool, which can be invoked manually or as part of a shell script. This tool can submit jobs either synchronously (waiting for results) or asynchronously (such as receiving notification by email when a job is completed). Using the appropriate XML property list specifications, it even has the ability to submit jobs containing multiple tasks.

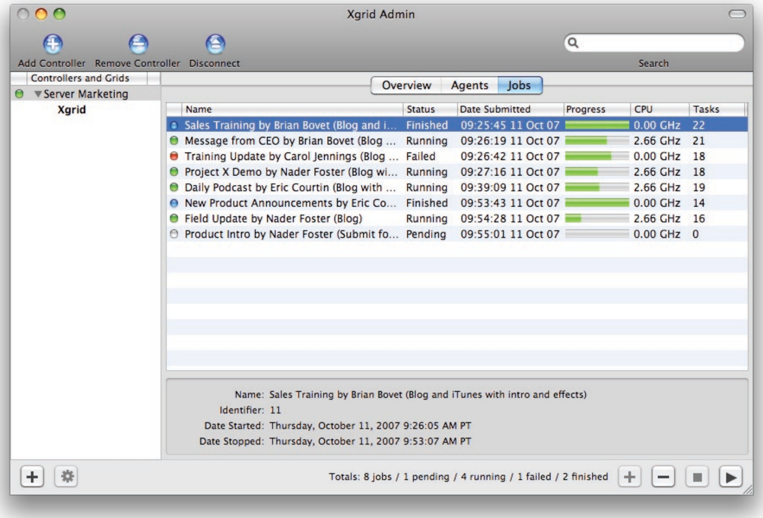
Developers can create rich Cocoa clients using the Xgrid Foundation API, as demonstrated by several examples in the Xcode Developer Tools (included with Mac OS X). The API uses a "lazy request" model, similar to that of the command-line tool, to identify resources and results, and to submit jobs and tasks. This allows developers to rapidly add Xgrid capabilities to existing applications, or to develop rich front ends to existing

UNIX tools. Cocoa developers can simply factor out the computationally intensive portion of their applications (even ones that use 64-bit pointers) and use the asynchronous, request-response Xgrid framework to distribute the resulting tool across as many machines as are available.

The Xgrid command-line tool and Cocoa application frameworks are available on Mac OS X as well as Mac OS X Server, making it easy to submit jobs from any system running Mac OS X v10.4 or later.

Controller

The controller, representing the middle tier, is the heart of Xgrid. Typically running on a dedicated system, the controller handles the hard work of receiving jobs from clients, splitting them into tasks to run on various agents, and collecting and returning the results.



Name	Status	Date Submitted	Progress	CPU	Tasks
Sales Training by Brian Bovet (Blog and i…	Finished	09:25:45 11 Oct 07		0.00 GHz	22
Message from CEO by Brian Bovet (Blog …	Running	09:26:19 11 Oct 07		2.66 GHz	21
Training Update by Carol Jennings (Blog …	Failed	09:26:42 11 Oct 07		0.00 GHz	18
Project X Demo by Nader Foster (Blog wi…	Running	09:27:16 11 Oct 07		2.66 GHz	18
Daily Podcast by Eric Courtin (Blog with …	Running	09:39:09 11 Oct 07		2.66 GHz	19
New Product Announcements by Eric Co…	Finished	09:53:43 11 Oct 07		0.00 GHz	14
Field Update by Nader Foster (Blog)	Running	09:54:28 11 Oct 07		2.66 GHz	16
Product Intro by Nader Foster (Submit fo…	Pending	09:55:01 11 Oct 07		0.00 GHz	0

Name: Sales Training by Brian Bovet (Blog and iTunes with intro and effects)
Identifier: 11
Date Started: Thursday, October 11, 2007 9:26:05 AM PT
Date Stopped: Thursday, October 11, 2007 9:53:07 AM PT

Totals: 8 jobs / 1 pending / 4 running / 1 failed / 2 finished

Xgrid controller software is built into Mac OS X Server. The controller is managed via the Xgrid Admin application, which comes on the standard Server Admin CD bundled with Mac OS X Server. (Xgrid Admin can also be run from Mac OS X systems.) This program allows administrators to monitor and manage jobs and agents on multiple controllers, as well as to partition agents on a single controller into multiple distinct grids.

Agent

The final tier—the one that performs the actual computation—consists of Xgrid agents. Typically there is one agent per compute node; agents on dual-CPU nodes default to accepting one task per CPU. Each agent can be configured either to always accept tasks, or to accept or run tasks only when its system is idle. Note that the agent simply launches the provided task; it does not interfere with the actual execution of that task.

Xgrid agent software is built into both Mac OS X and Mac OS X Server. Mac computers running Mac OS X v10.3 or later can communicate with the Xgrid controller via agent software available as a free download from Apple.

Podcast Producer
Xgrid technology powers Podcast Producer, Apple's end-to-end solution for producing and distributing professional-quality audio and video podcasts. Podcast Producer leverages Xgrid to encode content into multiple formats, such as H.264 and MPEG-4, for playback on a Mac, PC, iPod, iPhone, or Apple TV. For large-scale podcast productions, Podcast Producer utilizes Xgrid technology to distribute encoding tasks automatically to other servers.

Setup

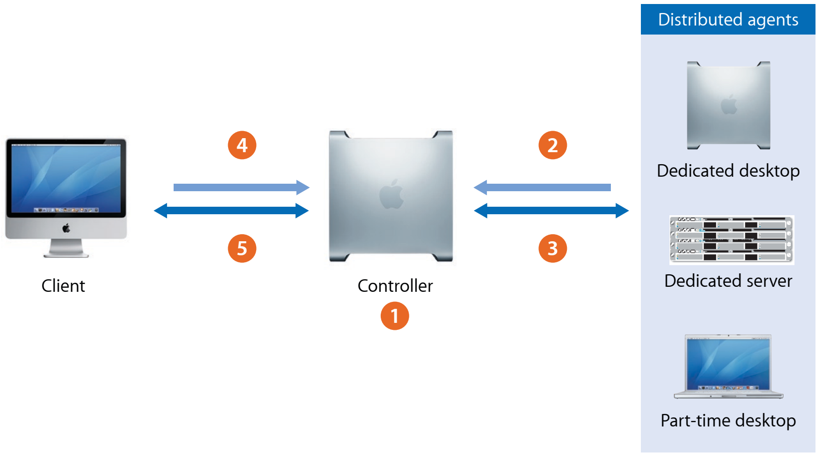
For most Xgrid projects, no installation is necessary. Appropriate client, agent, and controller command-line tools and GUI APIs are included with Mac OS X and Mac OS X Server.

Mac OS X desktop systems can be configured and enabled as Xgrid agents locally through the Sharing pane in the System Preferences application—or remotely via SSH or one of Apple's network-based desktop management tools (Apple Remote Desktop, NetBoot, or NetInstall). Mac OS X Server systems are configured and enabled as Xgrid agents or controllers through Server Admin.

You will also need to use Server Admin to specify a means of authentication for your agents and controllers: Password, Single Sign-On (using Kerberos), or None. When everything is up and running, the Xgrid Admin application makes it easy to manage agents, controllers, and Xgrid jobs.

Note that the Xgrid control interface tool allows you to enable and manage Xgrid services (although not authentication) from the command line.

Client and Agent Discovery



1 Controller advertises via Bonjour.

2 Agents locate controller using Bonjour, DNS, or name/address.

3 Agents and controller mutually authenticate using passwords or single sign-on.

4 Clients locate controller using Bonjour, DNS, or name/address.

5 Clients and controller mutually authenticate using passwords or single sign-on.

For optimum security and flexibility, both clients and agents are responsible for finding the controller. The controller is the only Xgrid component that requires an open port. If you are using a firewall, you will need to register the controller with the firewall.

Xgrid can automatically discover available resources on a local network via Bonjour. If you specify a DNS domain, Xgrid will find the computer acting as the Xgrid controller for that domain. You can also configure Xgrid resources by manually entering IP addresses or host names.

Xgrid in action

Xgrid is the enabling technology for numerous cutting-edge scientific projects, including the following:

- Center for Advanced Computation at Reed College—nonlinear-system computations for an epidemiological model
- University of Utah Student Computing Labs—rendering POV-Ray animations of LDraw models
- Simon Fraser University—finding low-autocorrelation binary sequences
- Ontario Cancer Institute at the University of Toronto—Monte Carlo simulations of biophotonic tissue analyses

Xgrid@Stanford

The world's largest Xgrid cluster is run by Charles Parnot, a post-doctoral fellow in the Molecular and Cellular Physiology Department at Stanford University. He uses Xgrid@Stanford for 3D modeling of G protein–coupled receptors to study heart disease and heart-rate control. For faster calculations and faster results, his work depends on a cluster of more than 200 computers—in which each computer belongs to a different person, connected to the Internet somewhere in the world. This cluster allows his team to run in one week a calculation that would take an entire year without Xgrid technology.

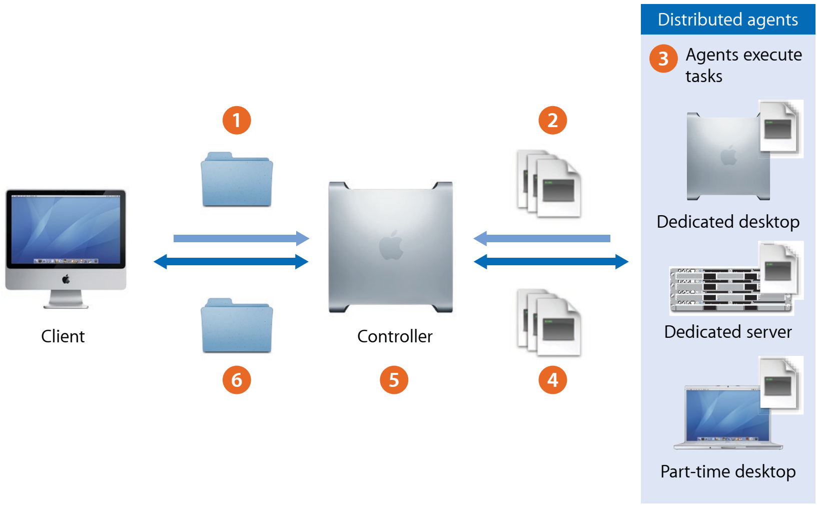
Message Passing Interface (MPI)

The MPI protocol is an efficient solution for communications among the nodes of a distributed computing job. It consists of a standard set of API calls that manage all aspects of internode communication and data transfer. Popular MPI implementations for Mac OS X Server include MPICH, LAM/MPI, and MacMPI.

Using Xgrid

Jobs can be submitted to the Xgrid controller using either the Xgrid command-line tool or a Cocoa GUI application. An accompanying specification file tells the controller whether to break down the job into multiple tasks, which code and data files are needed to execute the job, and whether the submission is synchronous or asynchronous.

Xgrid Job Process



1. Client submits job to controller with binaries/data.
2. Controller splits job into tasks, then submits tasks to agents.
3. Agents execute tasks.
4. Agents return results to controller.
5. Controller monitors and resubmits tasks as needed.
6. Controller collects tasks results, then returns job to client.

Xgrid schedules jobs in the order they are received, assigning each task to the fastest agent available at that time. Alternatively, you can schedule dependencies among jobs and tasks to ensure that they are scheduled in the proper order. In most cases, if any task fails (such as if a user turns off his or her Mac), the scheduler will automatically resubmit it to the next available agent. The exception is an MPI-style cooperative job, where all tasks are scheduled simultaneously and either succeed or fail together.

Xgrid tasks using password-based authentication will minimize possible interactions with the rest of the system by executing as unprivileged Xgrid users. Tasks using single sign-on for both clients and agents will run as the submitting user, allowing appropriate access to local files or network services.

Because of the three-tier architecture of Xgrid, clients can submit jobs to the controller asynchronously, then disconnect from the network. The controller will cache all the required data, manage scheduling and task-level failover, then hold and return all the results, including standard output and standard error from each task. If requested, the controller will notify the user via email that the job has been completed. The user can then retrieve the results from any authenticated Xgrid client system using the relevant job ID.

Apple Server Solutions

Xgrid is just one of the services built into Mac OS X Server—Apple's UNIX® server operating system. Combining the latest open source technologies with Mac ease of use, Mac OS X Server unleashes the power of Xserve, Apple's rack-optimized server hardware. With phenomenal performance, massive storage capacity, high-bandwidth I/O, and integrated remote management tools, Xserve running Mac OS X Server is the ideal solution for education, small businesses, and enterprises alike.

For More Information

For more information about Mac OS X Server, Xserve, and other Apple server solutions, visit www.apple.com/server.

*Actual scalability depends on problem type and hardware configuration.