Server User Guide

Version 1.0

User Guide to the Lima Project Server
Version 1.0

First Edition 25 Feb 2003
Authors: dc, ts

Contents

Introduction

Computers undoubtedly have become increasingly powerful and able to solve ever larger problems over the last few years. This has enabled previously impossible tasks such as the factorisation of large composite numbers to be performed in a timespan which facilitates practical uses of such algorithms. Nevertheless, as the capabilities of modern technology continues to grow, the demand continues to outstrip it as man further strives to break through ever greater barriers. Only by harnessing the collective power of a collection of computers can we ever achieve the goals of such pioneers, enabling mathematicians, scientists and engineers alike to benefit from this technological age.

Project Lima aims to provide a platform upon which large computations may be performed. These are then distributed over a network of computer systems in order that their combined computation power may be utilised in solving large problems. This platform is designed to be both robust and flexible that it may be used within the context of any algorithm which contains disjoint processing units across a network where the connections are uncertain. Automatic server detection and connection and an active-client/passive-server mode provide easy administration of the system and the flexibility of transient clients.

In this particular implementation, the system builds upon this platform to provide factorisation of large composite integers, optimised for integers up to approximately 100 digits. Although this is probably an NP-complete problem, various probabilistic algorithms exist to obtain prime factors in a reasonable amount of time, especially when they are distributed over a number of computer systems to share out the processing. Lima performs in succession trial division, Pollard Rho, Elliptic Curve and Quadratic Sieve algorithms, of which the last two use the Lima distribution server to spread out the work over a number of clients.

System Requirements and Installation

Running a server which communicates with a large number of client host machines will always require a machine which is able to give a significant amount of processing power to managing these requests. A good, reliable and fast network connection is essential if any reasonable level of performance is to be achieved. Furthermore, the factorisation of large integers requires considerable computer processing on the server side, even if portions of the work are sent out to clients. This is particularly true of Quadratic Sieve. Furthermore, to store the data structures generated, there needs to be plenty of memory available to act as storage, otherwise permanent storage may be used as virtual memory to hold this data and there will be significant performance penalties. Given these considerations, it is highly recommended that a dedicated, modern, well specified PC is used, connected by a 100Mbps Ethernet to the clients. As a guide, the following specification is recommended:

Hardware Required:

Intel Pentium® or AMD Athlon® XP™ Processor 1.5Ghz or better
512Mb DDR RAM or better
100Mbps Ethernet

Software Required:

Windows® 2000 or later OR Linux™ 2.2 or later
Java™ 1.4.1 or later
Any web server running on port 80 or 8080

Prerequisites

Since the system is Java based, Java is essential and due to various optimisations, version 1.4.1 is strongly recommended. The choice of operating system is limited to a platform supported by Java, although with the (non-Sun® related) Kaffe project, this is unlikely to be a problem. However, scripts to start the server have only been provided for Windows® and Linux™. If you do not have Java™ installed, then you need only to install the JRE, rather than the larger SDK. Windows®, Linux™ and Solaris® installations are available from http://java.sun.com. Sun also have ports to some other Unix™ based systems. All other operating systems can be used with Kaffe available from http://www.kaffe.org. Choose the 'Ports' link to check for availability. The applicable websites should be seen for installation instructions for Java.

The clients all run from applets which are served from web pages. This must be hosted on the same machine as the distribution server for security reasons. Therefore the server machine requires a web server. There are many high-quality web servers available, including Microsoft® IIS for Windows, Apache for Windows and Linux and Zeus™ Web Server for Windows. These however are all large and aside from Apache are very expensive. Therefore if a web server is not already in use, then a smaller, free web server is likely to suffice.

Installing Abyss Web Server on Windows

- Download Abyss Web Server from http://www.aprelium.com/
- Run the executable to install it
- Once it has installed, you should be asked to choose a username and password – type in something you will remember.
- You will then be asked to log in. If you are not, right-click on the Abyss Web Server icon by the clock and choose Show Console
- Choose Server Configuration and type in the documents path the location of the folder which you want to hold your website in. You will have to install Lima inside this folder.
- Set the port to 8080 if you are using a machine which you do not own
- Click OK and open a web browser on another computer. Type in the URL for the computer running the server (precede by http:// and followed by :8080 if you changed the port number) – you should get up a web page (possibly a file listing of the directory which you pointed to above). If you do not know the URL of the computer, then you can find out its IP address (which is one form of the URL) by running 'cmd' and then typing 'ipconfig'.

Installing WebFS on Linux

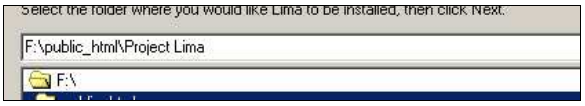
- Download WebFS from http://bytesex.org/
- Untar it by using tar xzvf <filename>
- Type make
- Run it by using ./webfs -r <docs-path> -p <port>
  where docs-path is the location to hold your web documents and port is 80 or 8080

Once you have a web server installed and running and Java is installed, you are ready to install Lima.

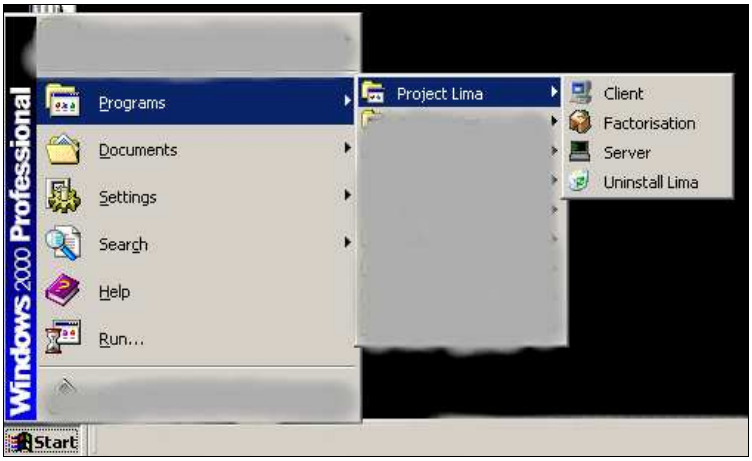
Installing Lima on Windows

- Run the setup program:

- Choose Next and type in the location where you want to install Lima. This should be inside the document root folder specified when you installed the web server. Click Next



- If you want, specify a different program group to install to. Otherwise accept the default option. Click Next. Choose Install. Wait for it to install and choose Finish. You should now have a new entry in the Start menu



Installing Lima on Linux

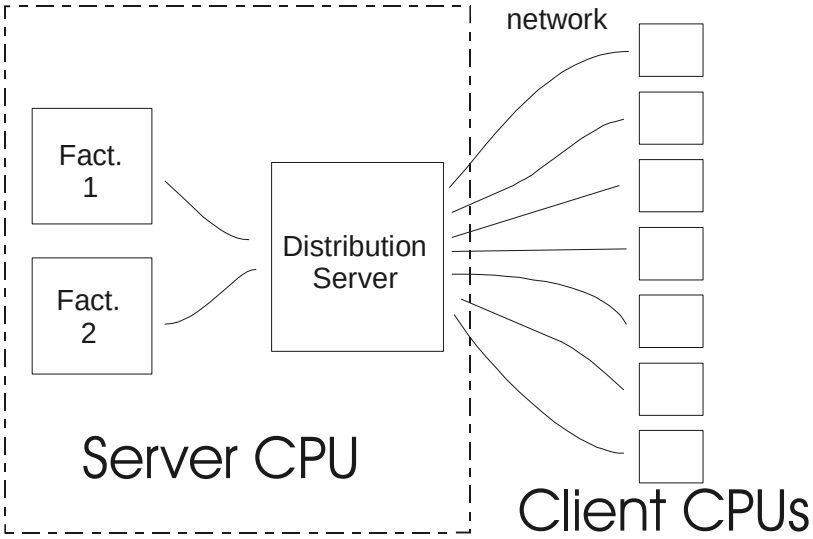
- Some knowledge of Linux and the command-line is assumed. Less technical users may prefer to use the Windows version due to its “point-and-click” interface
- Untar the Lima tarball (tar xzvf lima.tar) into a folder within your web server’s document path

The Server Components

The server computer runs two processes. The first is the distribution server and the second is the Lima factorisation server. A brief description of each of these is given below.

The distribution server sits on the server (listening on port 3000) waiting for clients to connect to it requesting work for them to do and for servers to connect to it to provide units of work which they need to be done. It is generally run in the background and although it continually displays messages, they are system logging messages which are only useful for troubleshooting. The distribution server should be run before everything else. A single distribution server can serve multiple processes providing work simultaneously. This allows for example several factorisations to be run at once and then for the work for all of them to be shared out amongst the different clients.

The Lima factorisation server provides the central control for the factoring of a large integer. A new instance of this is run every time you wish to factor a new composite. This runs via a web applet which provides a front end in which to provide a number to factor and a console which displays the current progress. The Lima factorisation server will look for a distribution server, when it requires one, and once found connect to it automatically. It will fail however if a distribution server is not already running at port 3000 on the same computer.



The distribution server sits between the clients and the processes running the algorithms which produce work to be distributed

Running the Distribution Server

Microsoft Windows

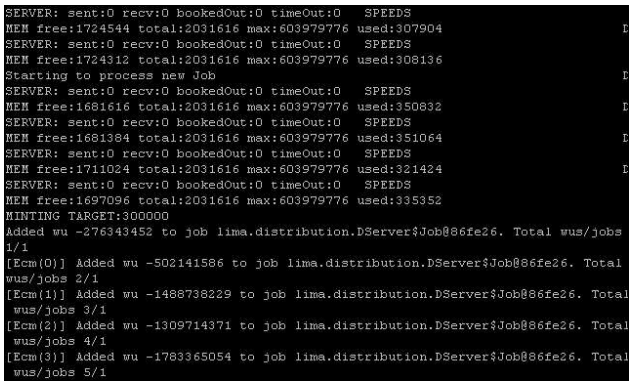
The distribution server can be started in Microsoft Windows by choosing the 'Server' item in the Start Menu. This should open up a new console which will immediately show something like:

The console opened for the distribution server

Java should then be started and messages will begin to appear from the distribution server. If it reports that it "is now bound in the registry" and proceeds to give status messages, it has started up correctly. If it gives an error message and terminates, it has failed and you should go to the troubleshooting section to resolve the issue. You will need to note the IP address of the computer you are running on. If you do not know what this is, then run 'cmd' from the Run entry on the Start menu and type 'ipconfig'.

Linux

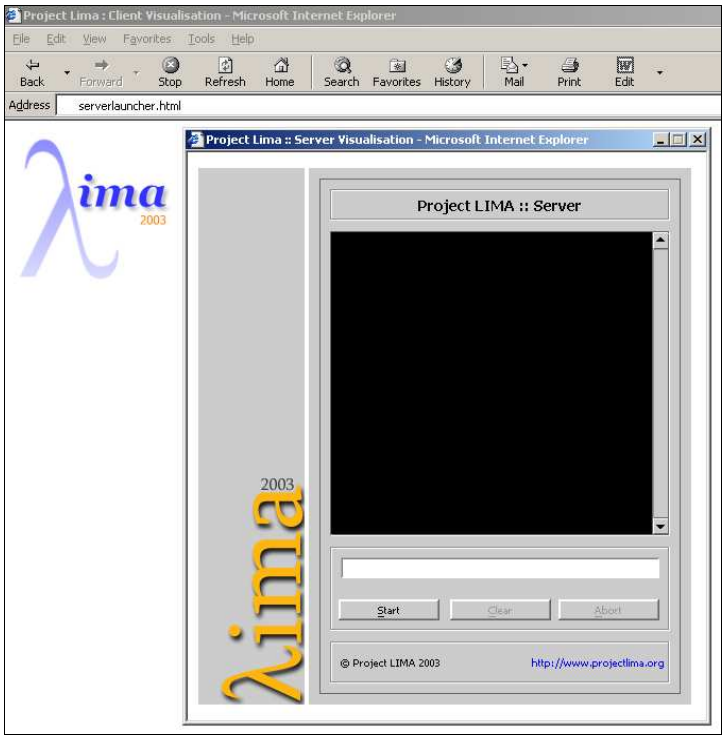
There is a bash script which will run the distribution server provided in the package. 'cd' to the path containing your Lima installation and run '`./server`'. You should be told that it has bound to the registry and status messages will be continually displayed. If you wish to sent these to a log file and have it run in the background, use '`./server > logFile &`'. Be sure to read this logfile immediately to ensure that the server has started correctly. You will need to note the IP address of the computer you are running on. If you do not know what this is, then run 'host $HOST'.



The distribution server in action – the messages displayed are generally only of troubleshooting use, but are interesting to gain an insight into the system nonetheless

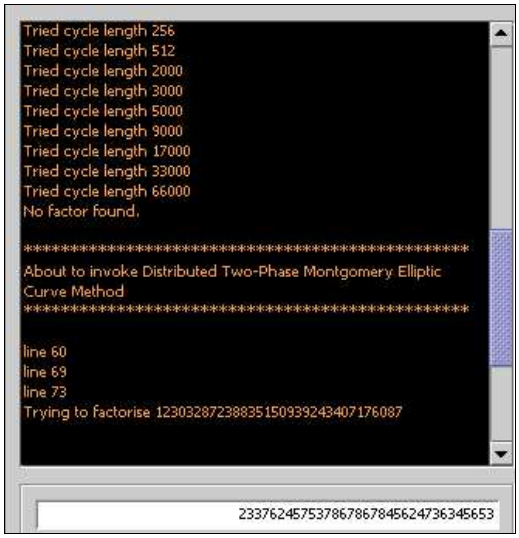
Factorising integers

To start a factorisation, the Lima factorisation server needs to be started. In Microsoft Windows choose the factorisation command from the Start menu and a web browser should be opened with the factorisation applet. In Linux open a browser and point it to `serverlauncher.html` in the installation folder. In either case, the following should be displayed:



The factorisation server applet

To factorise a composite, simply enter it into the white input box and click “Start”. You should then get status information about the current progress in the black console above.



In action – factoring a 33 digit number

At any stage the operation can be aborted by choosing the ‘Abort’ button which will stop all further processing on the server. Note that clients will continue with their current piece of work until they next attempt to submit it, since they only talk through the distribution server and not to the factorisation algorithm, and so do not know that it has terminated.

Abort will terminate the factorisation

The console can be cleared with the ‘Clear’ button.

To run clients, you will need to open a web browser on the client computers and point them to:

http://your-ip-address:port/lima-path/client-launcher.html

where your-ip-address is the IP address of the server (see above if you do not know what this is), port is the port number of your web server (usually 80 by default, 8080 if you specified it as such) and lima-path is the location of lima in your web documents path. For example:

http://142.33.6.34:8080/lima/client-launcher.html

Further Considerations

Factoring large numbers will take a long time, even with the probabilistic and finely tuned algorithms implemented in Lima. The time taken depends on the size of the factors rather than the composite and for numbers with factors of length less than 14, factorisation should be very fast. However, for those numbers with factors larger than 25, the process can be considerably longer and time should be allowed for such a job to complete.

Also it should be noted that due to the overhead of adding extra clients especially when the server has to talk to more than one client simultaneously, adding extra clients does not increase the performance of the job linearly. Therefore if you have a large number of clients available and several jobs to run, it may prove more prudent to run these in parallel rather than in serial. However, due to the amounts of memory consumed, it is recommended not to run more than one factorisation of large integers per 512Mb of RAM.

Troubleshooting

I get a ConnectException shortly after it says it is invoking Distributed Two-Phase Montgomery Elliptic Curve Method

> The factorisation process is unable to find the distribution server to send items of work ("WorkUnits") over the network. This is required for operation and so it fails. Ensure you have started the distribution server on the same computer as the factorisation process is running.

The distribution server fails saying that port 3000 is already in use

> Either you have a distribution server already running, in which case you do not need to run another one, or otherwise you have another program which is providing a service at the same location as the Lima distribution server runs from. Close down other network servers (other than the web server) and try again.

My clients stay in the 'resolving' state forever, even though I have a distribution server running

> Ensure that you are running the distribution server on the same computer as the web server providing the web page for the client. You may otherwise have a network problem. This can be checked by attempting to run a client on the server computer using the IP address 127.0.0.1. If this succeeds then the problem lies with your network.

The client cannot display all the font information in Linux clients

> This is a known problem. Install the Tahoma font by installing the Microsoft core fonts available from http://corefonts.sourceforge.net/.

The Lima factorisation module does not completely factorise my number

> The algorithms used are probabilistic in order to give a reasonable performance, since deterministic methods would take an inordinate period of time to complete. Occasionally however this will lead to failures.

Frequently Asked Questions

- **How long does it take to factor a number?**
  - That depends upon the size of the prime factors, the speed of the server, the network latency and the number and speed of the clients. Factors less than 8 digits are found very quickly (one or two seconds) and factors less than 24 digits are found in a fairly short amount of time. Factors larger than this may taken anything from a few minutes to a number of hours. However, it is very dependant on your hardware.
- **What happens if a client is turned off part way through doing some work?**
  - The distribution server detects that a client has taken too long without passing a completed work unit and so gives that work to another client to do. This is repeated until the work is finally returned by a client. Therefore the system is tolerant to client failure.
- **Can I start new clients after the process has begun?**
  - Yes. Clients register their interest with the distribution server when they start and the distribution server will use whoever calls it at any time it has work available.
- **Will this put a heavy load on my network?**
  - Although there is going to be an increase in your network traffic due to the inherent nature of the system, this has been minimised and the load is not particularly heavy, so should not have any noticeable effect on the average LAN.
- **Where can I find out more about the factorisation algorithms?**
  - See the website http://www.projectlima.org for recommended reading