Introduction

This is an informal manual on the gpu search engine 'gpuse'. There are some other documents available, this one tries to be a practical how-to-use manual.

Table of Contents

GPU Search Engine (GPUSE)

GPUSE comes with the GPU “Distributed Computing over a peer-to-peer network” package. GPU is the glue that holds several plugins, distributes requests, shows statistics and hosts the chat.

Contact

The chat is our main system for users and developers to exchange information. There is also a mailing list hosted at sourceforge “gpu-world” but you may notice direct contact may be very productive and interesting. The GPU chat is very interactive, the whiteboard can be used to exchange screenshots etc.

So, you wanted to crawl the web?

Consider a few things:

- Good hardware. Especially if your local database grows, it will take some system resources. Especially harddisk access, and system cache, are a factor. Another one is a good internet connection. A stable DNS server isn't a luxury either.
- Permanent connection. Preferably you have a permanent internet connection, so you can leave the crawler running overnight. Unlimited usage or a Fair Use Policy will help. Some ISP's will have another interpretation of fair use than you have, and with continuous crawling while time is ticking the amount of data easily cumulates. Be sure you know what you are doing.
- Patience. Continues crawling with 1 or 2 crawlers is much more efficient than short bursts with many crawlers.
- Controlled startup/shutdown of the gpu main application. With the search engine enabled, it may take (much) longer for gpu main application to completely shutdown. In case of a shutdown, first, all crawler threads need to be stopped. Then all recent collected data have to be flushed to disk. Normally this is done in 10-30 seconds or so, but if you have a busy crawler it may take longer.

Minimum System Specifications

The minimum recommended specs are:

- broadband. 50Kb/s or more.
- Reasonable cpu (133MHz or more).
- Modern harddrive (6Gb is really absolute minimum, 17 Gib a practical minimum, 120Gb or more will perform a lot better and is recommended.).
- Sufficient memory; 512Mb is no luxury. With 256Mb things will work, but you will notice the system is less usable as workstation in that case. 512 Mb with 2-3 crawlers running is reasonable stable on long term.
- Operating systems: XP, Windows 2000, 98 (not really recommended but should work), Linux: 2.4 or 2.6 kernel using wine. Most common releases should be working (Debian. Redhat) & wine (most recent releases) working. Wine may have issues.

Is GPUSE safe to run?

No and yes. With recent changes, we think yes, it is. There have been done a few modifications that significantly reduce disk access. Your harddisk should stay almost completely silent (no extraordinary head movements).

Also, there are some protection mechanisms to avoid network overload. In case a thread detects network 'malfunction' or slowness, it will sleep for half a minute before continuing with the next job, hereby avoiding abuse, overloading your DNS server, remote server or just your local system.

No because this is explicitly a beta test. Loads of situations might occur that somehow or another leads to undesired behavior. Although tested, we cannot guarantee an undesired behavior not to occur.

Is it safe for others (third party). Will it not 'DoS' sites?

We believe the crawler behaves very correctly. First of all, it examines the robots file that may be available, and strictly obeys instructions. Robots.txt may hold specific instructions for agent

'gpuse'. Then, a single crawler will never in short time access more than 5 documents of a single domain. Normally this will be less, but since there is some random factor involved, we limited number of pages from one site to 5 each time the crawler fetches a new set of urls. This number of 5 is a weighted efficiency factor that is introduced to reduce both database access and DNS Server requests. A local crawler gets urls on certain domains in batches of maximum 5 and cumulates until it reaches 100 or more urls or times out.

On startup with an empty database, the crawler has a few urls's hardcoded built in. Those urls typically point to pages that link to loads of other pages. Once the database grew to reasonable size, randomness is the factor that distributes the load.

Screw your isp's dns server.

Well probably you will not screw it but may get slow. “Crashing” a DNS server is definitively not impossible. Both scenario's may affect you and possibly other innocent users. Take care what you do. A single crawler is like 5 cyberpunks on speed surfing random sites non-stop 24h a day.

Can i view my crawled results immediately?

The first result set will be finished after about 1-2 hours of crawling with 1 or 2 crawlers on a typical broadband connection. You can view results of other crawlers on the network though.

Can i search in real-time over the p2p network?

Yes, you can, using the frontend. The search frontend also has a simple web server built in. For a collective reverse index, we are experimenting on a system based on mysql at http://search.dubaron.com

Can i search without crawling?

Sure you can. Just launch the frontend.

.

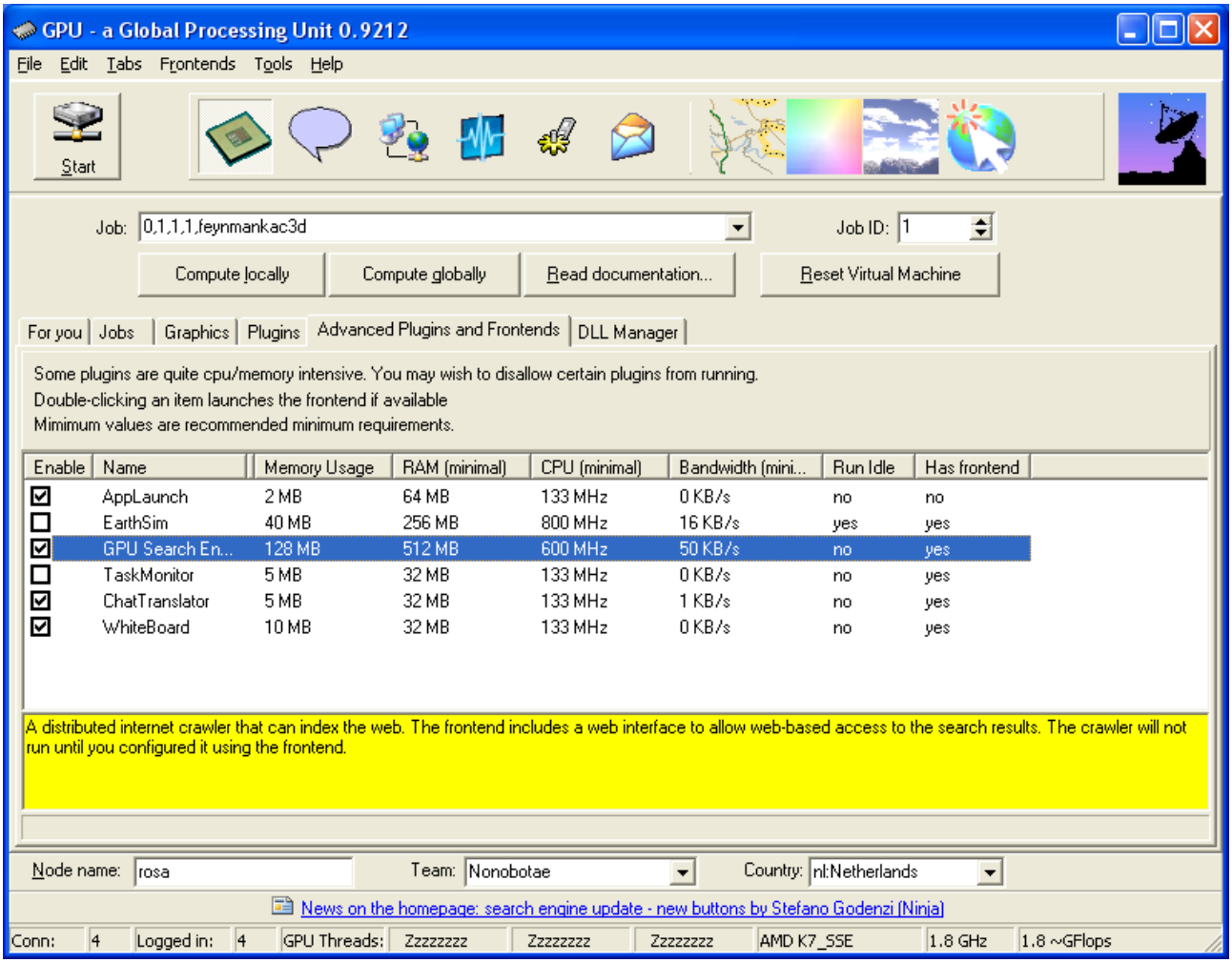
So you think you are ready? Ok. Let's go...

Some screen shots that may help guide you trough the process of enabling the crawler

Enabling the GPU Search engine plugin

GPUSE needs to be enabled on two places. First, within the GPU main application, you have to enable the search engine plugin. Then ***restart GPU*** to activate it.

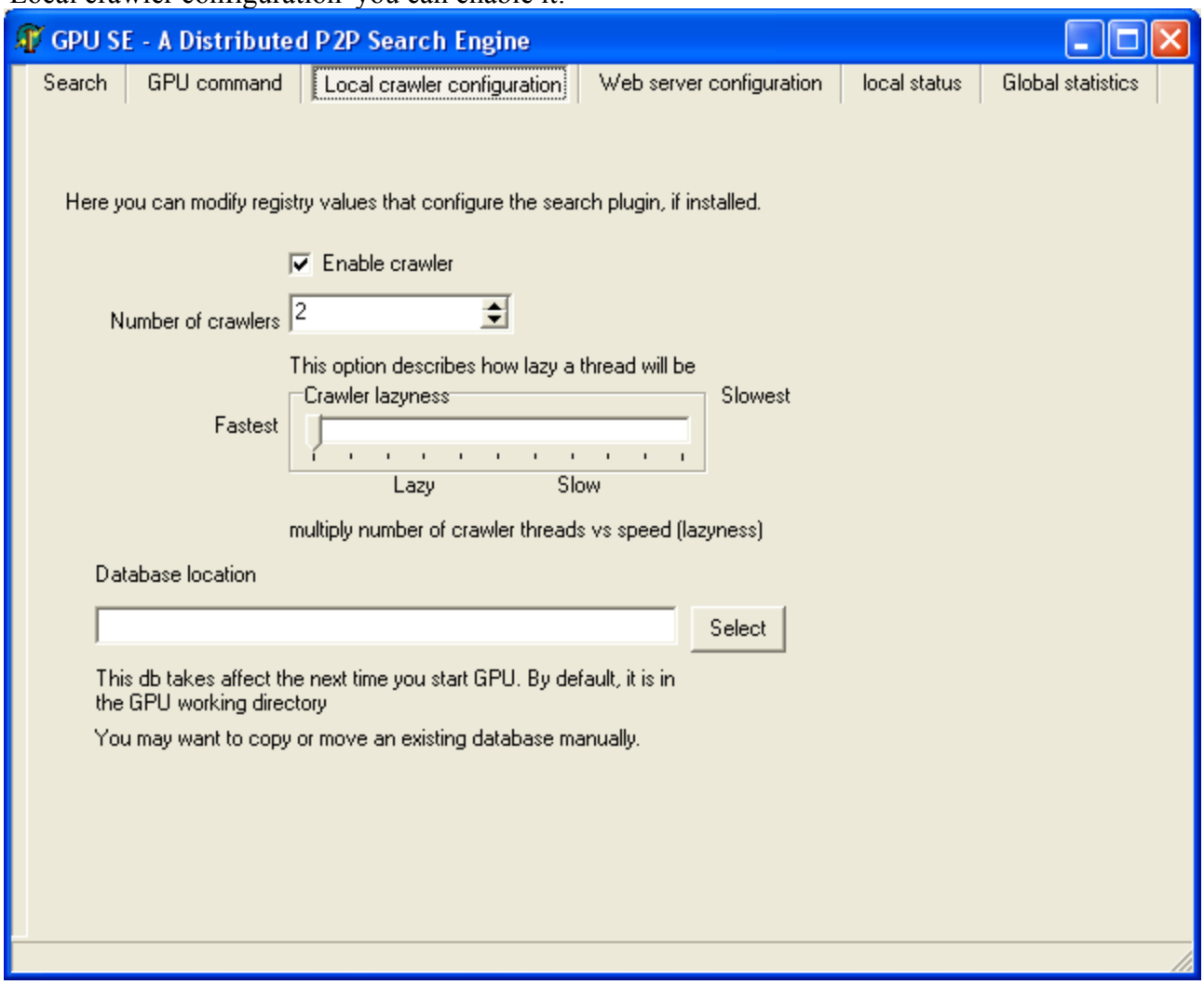
In the screenshot below, you can see the activated tabsheets. The line with the search engine is selected. Make sure the check box is checked.



The Search engine front end

After restarting the gpu application, enable the search engine frontend. Here you can control the crawler in real-time. There may be some latency before the crawler responds, varying from seconds to minutes, depending on the setting changed.

With tab sheet 'local status' you see logbooks of what the crawler is doing. With tab sheet 'Local crawler configuration' you can enable it:



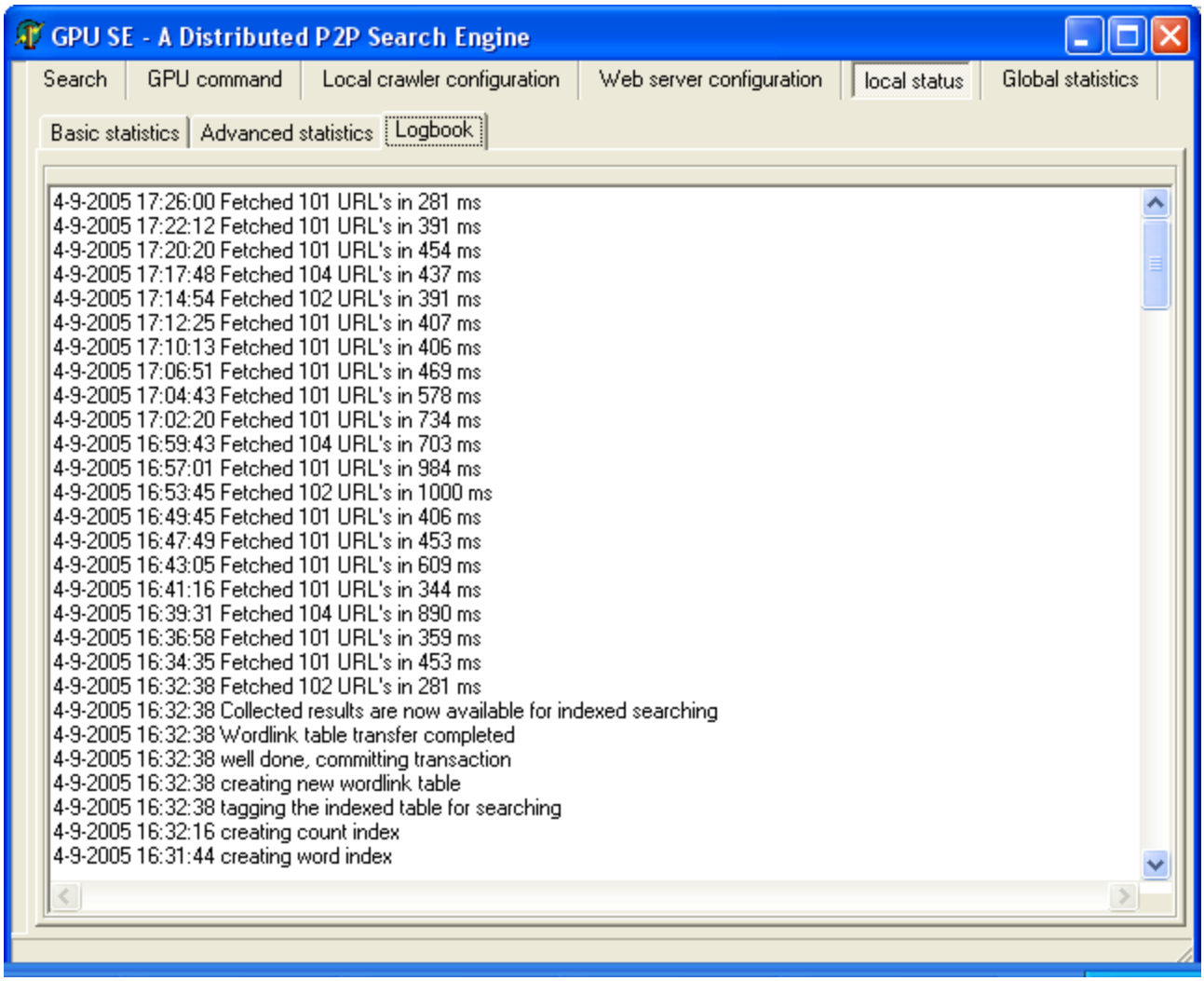
The default number of crawlers is 2. Use of 1 crawler is also very suitable. Remember that continues crawling with 1 or 2 crawlers is much more efficient than short bursts with many crawlers.

We set no maximum to the number of crawlers, but a real-life maximum is about 6. In general, it seems having 3 or 4 crawlers running will already take most of your available bandwidth.

Any number above 12 would be absurd, even on high end hardware.

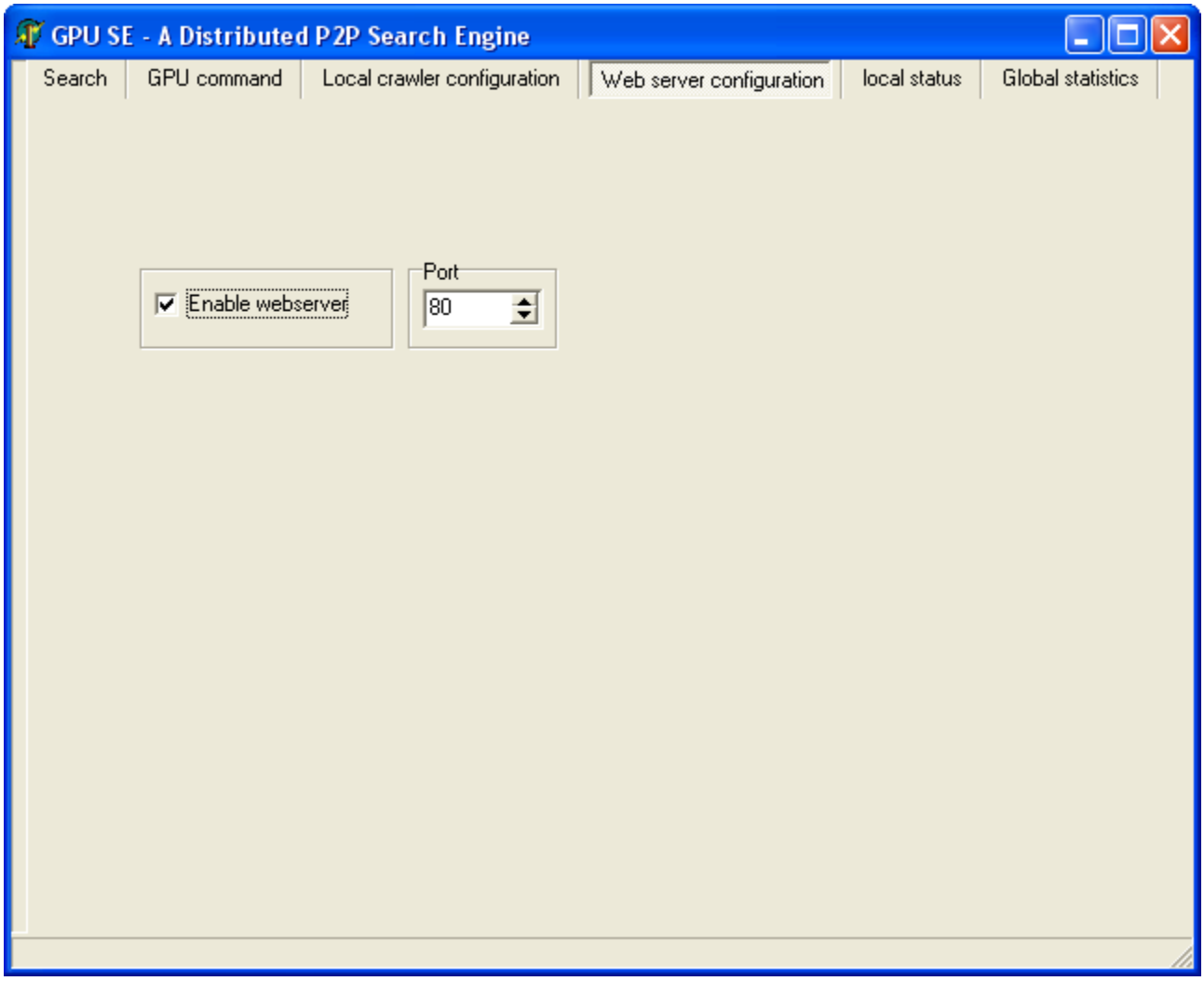
Check what your crawler is doing.

Check the tabsheet logbook:



If it says “Fetched 5 url's” or so, this is normal during initial startup phase. However, quite soon you should see numbers above 100. If it stays below, you have too few robots threads crawling. Reduce the number of crawlers in that case, and give the robots threads a chance to build a nice database of crawlable url's.

Enabling the webserver



Make sure the checkbox is checked
Use your browser to navigate to http://localhost/
If you use another port number than 80, for example 81, tell your browser:
http://localhost:81/

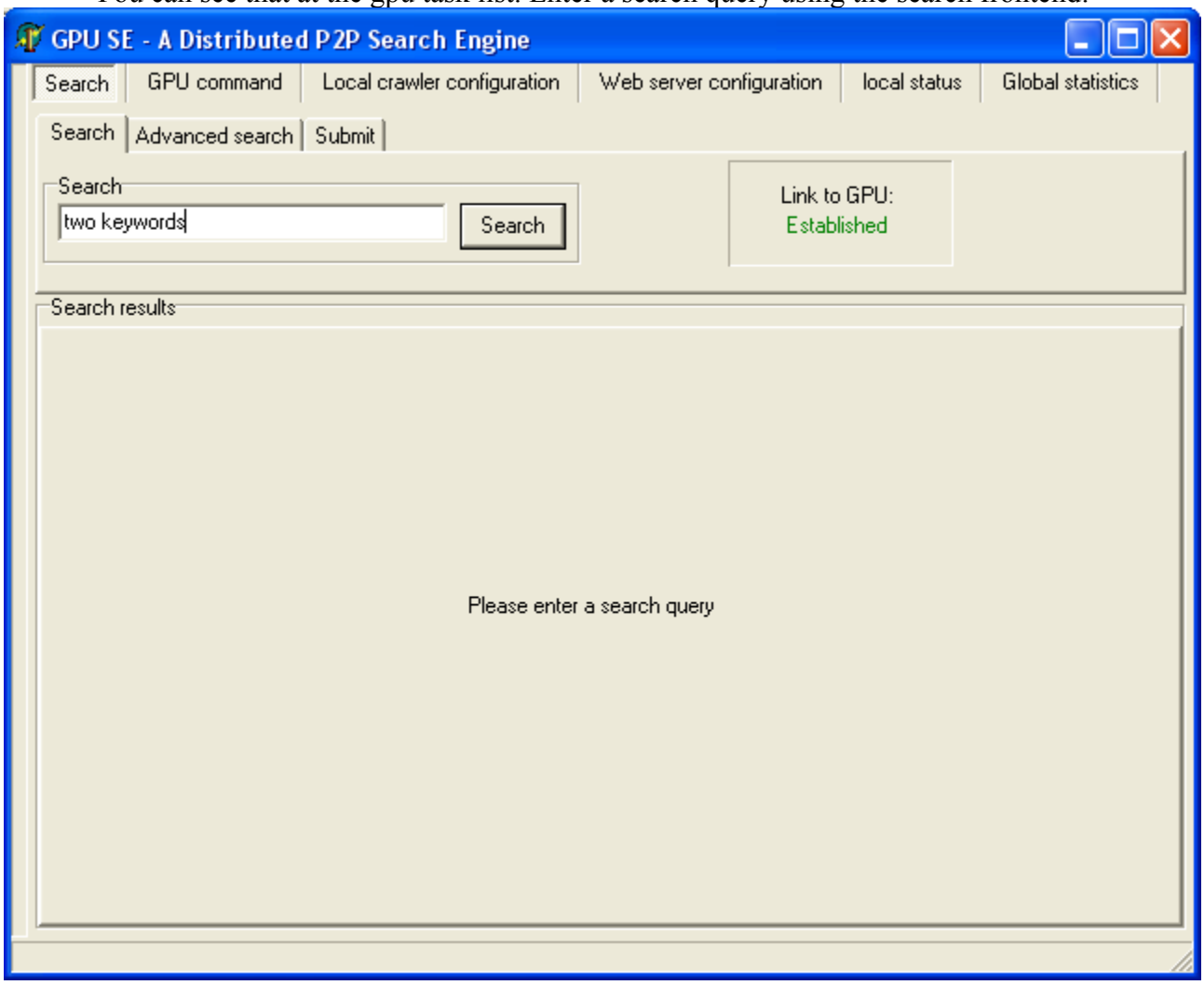
Security considerations

Please keep in mind that if you enable the web interface others can use it as well. Although it is multi-threaded, there is a chance of overloading the gpu network, so take care.

What is more

And, how to see if my gpu returns results?

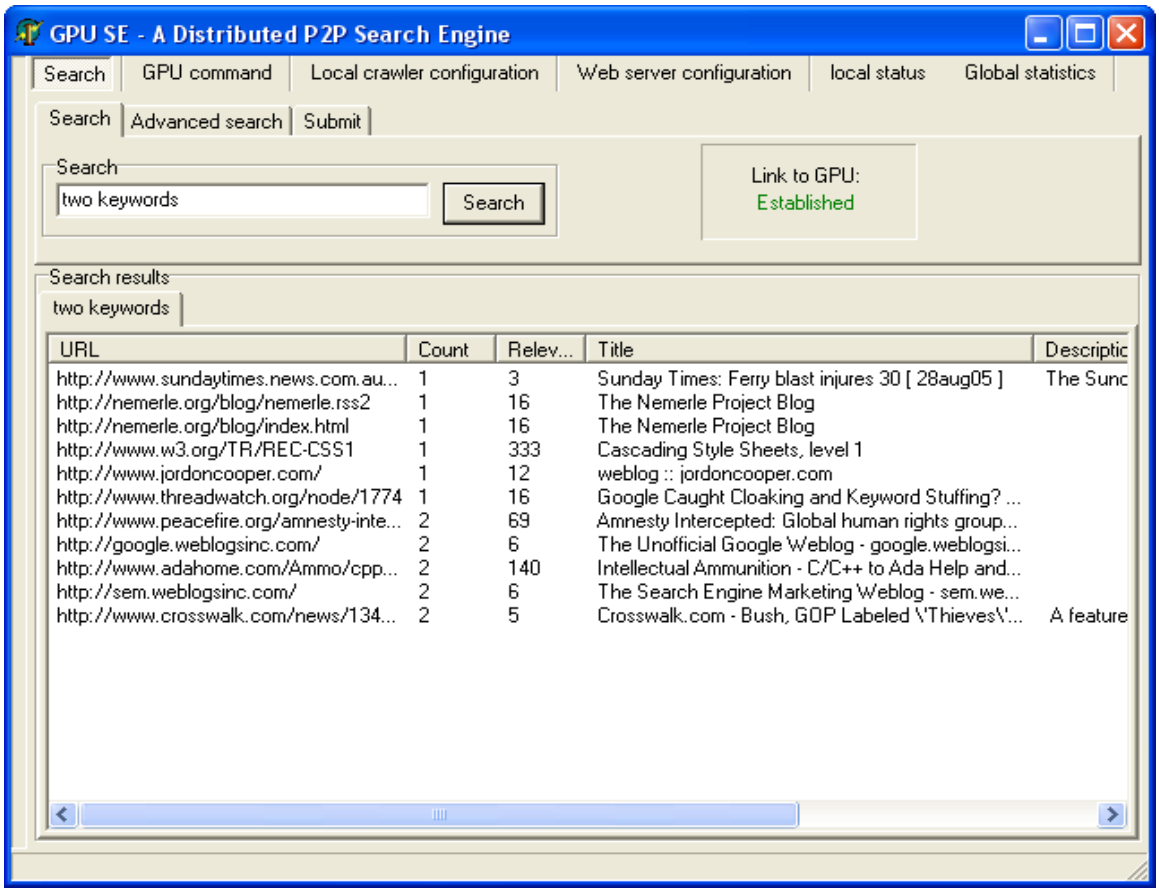
You can see that at the gpu task list. Enter a search query using the search frontend:



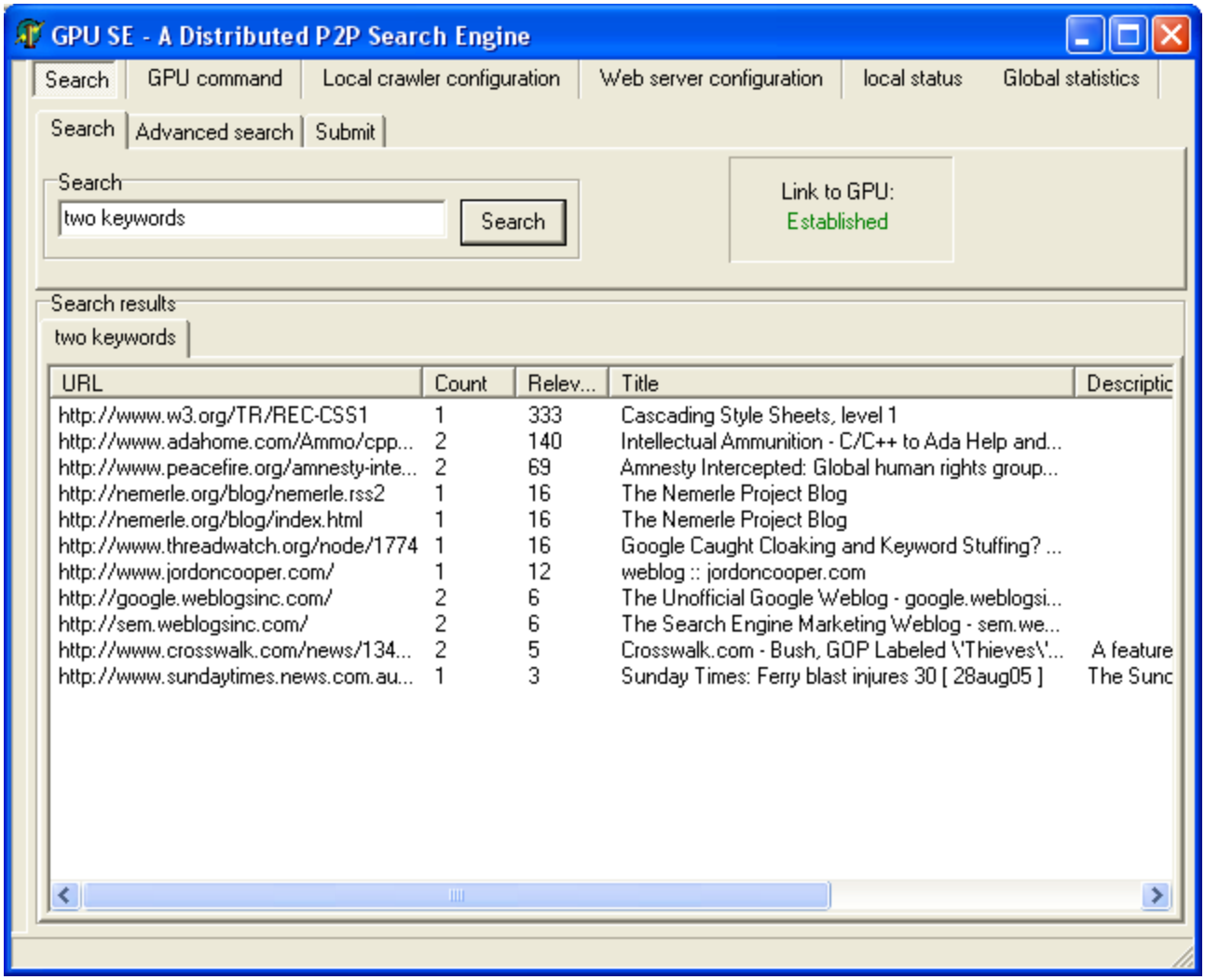
With gpu task list you can see the search is executed:

You see it twice due to an (unfiltered) roundtrip. You got the same request back from another client. This is a gnutella issue we are working on.

After waiting a few seconds, you can see the results:



You can sort the results by clicking the column name:



Of course, you can also use the web interface on localhost:

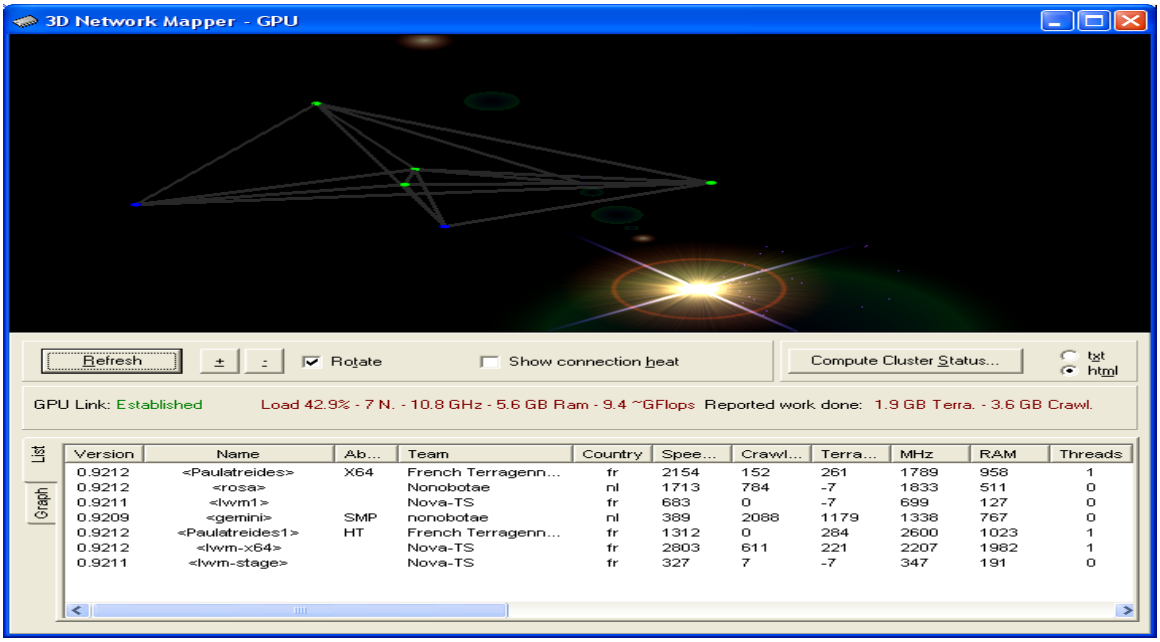
The reverse index

Or, even more sophisticated, using one of the web interfaces that maintain a permanent reverse index (discussion of how to set up this mysql webinterface in another document):

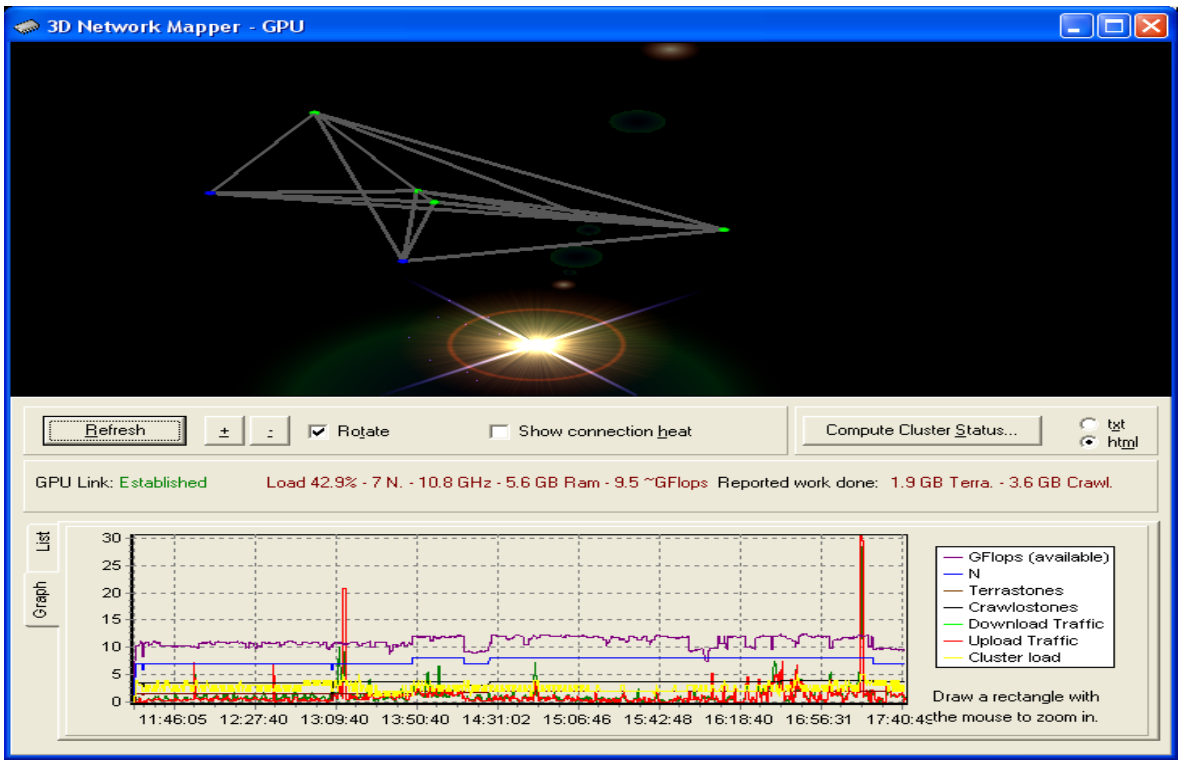
Investigating the gpu network with netmapper

With netmapper, we can see what the nodes are crawling. The newest software has just been installed and is being tested:

A numeric representation:

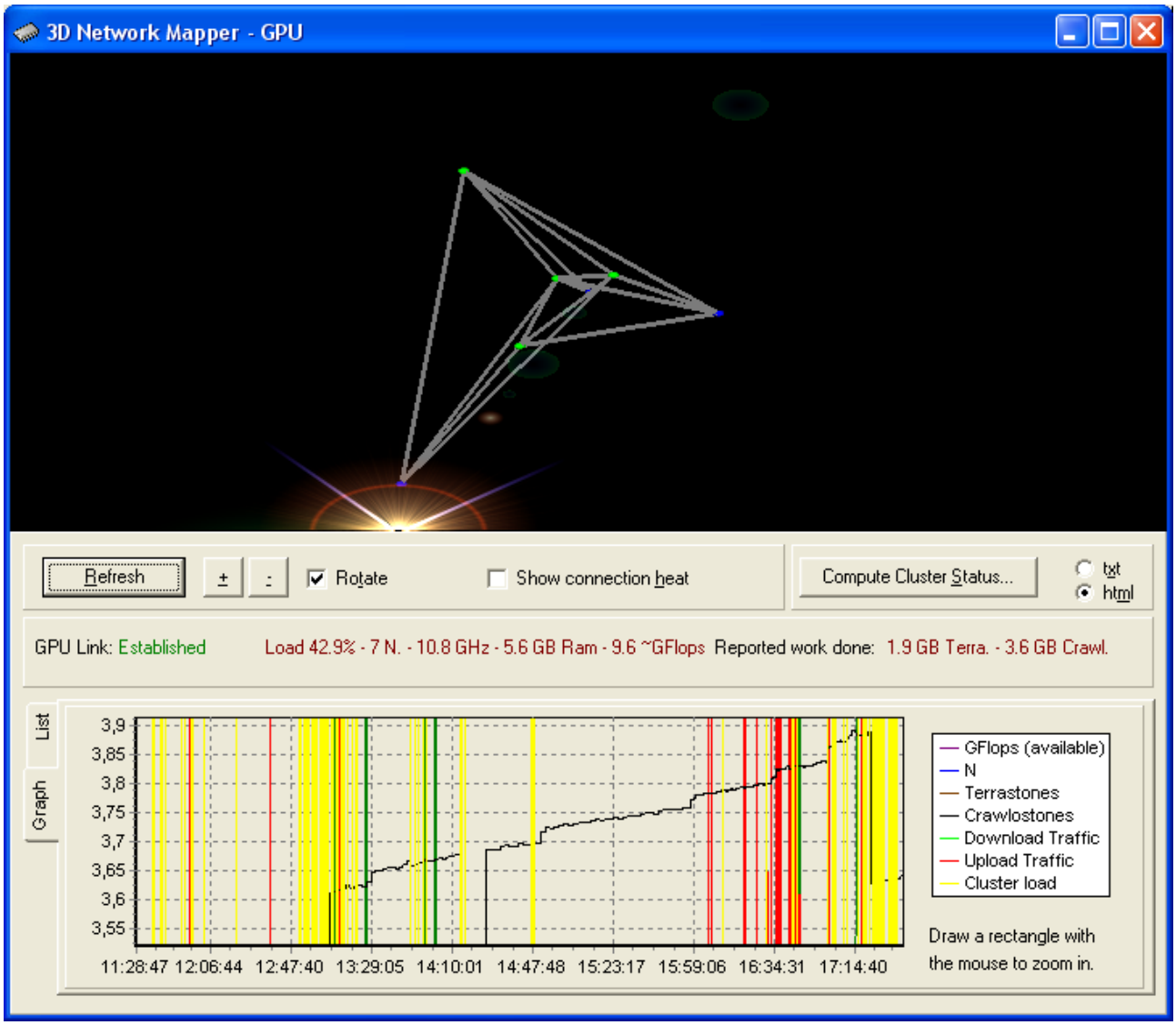


A Graphical representation:



The crawler's progress

If we zoom in on the crawler line (the black one) we can see the database slowly growing:



Concluding

Although not a scientific project, we believe the GPU Search Engine is a working proof of concept. If this network would scale to tens or even hundreds of crawlers, we can expect excellent search results and a large portion of the web crawled. Improvements may be made to sorting algorithms, but this KISS concept also has advantages. For many queries, page relevance sorting shows much similarities to other search engines, like google.

Many thanks to all that keep the GPU network alive!