Subject: SCGI Class
Posted by jeremy_c on Tue, 17 Aug 2010 14:35:26 GMT
View Forum Message <> Reply to Message

I have created a SCGI class. SCGI is a more simple FastCGI interface. It allows the application to persist between connections, thus respond very quick unlike a CGI program that has to be loaded, initialized, executed and closed for each request.

This really comes in handy when there are expensive start up costs such as a database connection.

An example application:

```
#include <SCGI/SCGI.h>

class HelloWebApp : public ScgiServer {
public:
  HelloWebApp() : ScgiServer(8787)
  {
    WhenRequest = HandleRequest;
  }

  void HandleRequest()
  {
    clientSocket.Write("Content-Type: text/plain\r\n\r\n");
    clientSocket.Write(Format("Hello, %s!", query["NAME"]));
  }
};

CONSOLE_APP_MAIN
{
  HelloWebApp app;
  app.Run();
}
```

There are other callbacks such as WhenAccepted, WhenClosed. "query" is public and is an instance of HttpQuery that is automatically populated. "map" is a VectorMap<String,String> that is also automatically populated with the server variables that are passed such as REQUEST_URI, SERVER_NAME, etc...

On my NetBook (1.6ghz Atom) the above SCGI app runs ~1200 requests a second. A static hello.txt file is ~1900 requests a second.

I still have a few loose ends to wrap up (post form data, some more general testing) but wondering if anyone else would find this useful and how to share it?

Oh... most web servers (Apache included) have a "mod_scgi" to interface with this type of application. The applications need not reside on the same computer as the web server, thus they can be distributed and offer load balancing. Advanced servers like Apache can do load balancing themselves internally knowing which SCGI app is in use and which one is not.

For anyone who wants to know more about the SCGI protocol: http://python.ca/scgi/protocol.txt

Jeremy

Subject: Re: SCGI Class
Posted by zsolt on Wed, 18 Aug 2010 15:50:51 GMT
View Forum Message <> Reply to Message

This would be a useful stuff
I think, you can upload it here.

Subject: Re: SCGI Class
Posted by jeremy_c on Thu, 19 Aug 2010 13:32:01 GMT
View Forum Message <> Reply to Message

Ok, I am attaching the Scgi package as well as a ScgiHello example. It now handles parsing query string, post data and server variables internally. There are no docs yet but it's use is pretty simple to figure out. The ScgiHello example helps w/this as well.

Once it gets a review or two, I'll make any corrections and start documenting. The ScgiHello example includes a sample LigHTTPD configuration file that should run out of the box.

It does use some signal handling which I have not tested on Windows.

I would like to upload this and begin to maintain this in Bazaar. How does one go about doing that? (Oh, I have the MultipartForm class that I added to Web/util.h/cpp that I think would be helpful for others).

Jeremy

File Attachments

1) ScgiPreview1.zip, downloaded 666 times

Subject: Re: SCGI Class
Posted by mirek on Sun, 03 Oct 2010 18:19:09 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Thu, 19 August 2010 09:32Ok, I am attaching the Scgi package as well as a ScgiHello example. It now handles parsing query string, post data and server variables internally. There are no docs yet but it's use is pretty simple to figure out. The ScgiHello example helps w/this as well.

Once it gets a review or two, I'll make any corrections and start documenting. The ScgiHello example includes a sample LigHTTPD configuration file that should run out of the box.

It does use some signal handling which I have not tested on Windows.

I would like to upload this and begin to maintain this in Bazaar. How does one go about doing that? (Oh, I have the MultipartForm class that I added to Web/util.h/cpp that I think would be helpful for others).

Jeremy

Perfect! (sorry for begin late here...)

In fact, I think this can go directly to uppsrc, bypassing bazaar this time. Just please test if this works in Windows too...

ScgiServer now in uppsrc.

I am sending you access info to svn server now...

Mirek

Subject: Re: SCGI Class
Posted by jeremy_c on Mon, 04 Oct 2010 16:52:01 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 03 October 2010 14:19
Just please test if this works in Windows too...

Tested on Windows (all works as designed) using nginx (single executable web server that will work on Windows as well as *nix systems). I added a config file for nginx to the ScgiHello reference application so others can play with it on Windows as well.

Jeremy

Subject: Re: SCGI Class
Posted by mirek on Mon, 04 Oct 2010 17:12:16 GMT

Looking at it, I have strong impression that in this case it would be more logical to use virtual methods instead of Callbacks. I guess even HelloScgi would be quite simpler that way..

Also, as this seems to be quite short piece of code, I consider moving it to Web package (you would retain access rights there too).

What do you think about this?

Subject: Re: SCGI Class
Posted by jeremy_c on Mon, 04 Oct 2010 17:19:33 GMT

luzr wrote on Mon, 04 October 2010 13:12Looking at it, I have strong impression that in this case it would be more logical to use virtual methods instead of Callbacks. I guess even HelloScgi would be quite simpler that way..

Also, as this seems to be quite short piece of code, I consider moving it to Web package (you would retain access rights there too).

What do you think about this?

I was thinking the same about virtual methods vs. callbacks.

In regards to moving to the web package, that depends on how you would like the overall structure. Say someone else comes along and makes FastCGI and then WSCGI... How would you envision those being added? I'm not saying someone is, but just something to think about.

It does seem odd out there in uppsrc/ all by it's lonesome.

Jeremy

Subject: Re: SCGI Class
Posted by mirek on Mon, 04 Oct 2010 18:28:55 GMT

jeremy_c wrote on Mon, 04 October 2010 13:19luzr wrote on Mon, 04 October 2010 13:12Looking at it, I have strong impression that in this case it would be more logical to use virtual methods instead of Callbacks. I guess even HelloScgi would be quite simpler that way..

Also, as this seems to be quite short piece of code, I consider moving it to Web package (you would retain access rights there too).

What do you think about this?

I was thinking the same about virtual methods vs. callbacks.

Well, perhaps you could test the svn access rights and provide this change (and to reference example too)?

Quote:
In regards to moving to the web package, that depends on how you would like the overall structure. Say someone else comes along and makes FastCGI and then WSCGI... How would you envision those being added? I'm not saying someone is, but just something to think about.

I do not think this is really a problem - we can live with complicated stuff in separate package and simple stuff as part of something...

Subject: Re: SCGI Class
Posted by jeremy_c on Mon, 04 Oct 2010 21:54:23 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 04 October 2010 14:28
Well, perhaps you could test the svn access rights and provide this change (and to reference example too)?

No problem, it's working now (didn't commit yet). Is there a naming convention I should follow in regards to the virtual events? For example, WhenAccepted... OnAccept ?

Jeremy

Subject: Re: SCGI Class
Posted by mirek on Mon, 04 Oct 2010 22:14:18 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Mon, 04 October 2010 17:54luzr wrote on Mon, 04 October 2010 14:28
Well, perhaps you could test the svn access rights and provide this change (and to reference example too)?

No problem, it's working now (didn't commit yet). Is there a naming convention I should follow in regards to the virtual events? For example, WhenAccepted... OnAccept ?

Jeremy

Just "Accept" would do fine IMO, but if you feel like "On" makes it any better, no problem with me

---

Subject: Re: SCGI Class
Posted by jeremy_c on Mon, 04 Oct 2010 22:30:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 04 October 2010 14:28
Well, perhaps you could test the svn access rights and provide this change (and to reference example too)?

Made the change. moved the code and when I attempted to commit I got in the SVN output:

```
Deleting    Projects\UppSrc\uppsrc\ScgiServer
Adding      Projects\UppSrc\uppsrc\Web\ScgiServer.cpp
Adding      Projects\UppSrc\uppsrc\Web\ScgiServer.h
Sending      Projects\UppSrc\uppsrc\Web\Web.upp
Sending      Projects\UppSrc\uppsrc\Web\init
Transmitting file data ..svn: Commit failed (details follow):
svn: Access denied
```

TheIDE then promptly quit on me w/a Windows "stopped working" dialog.

Jeremy

Subject: Re: SCGI Class
Posted by mdelfede on Thu, 07 Oct 2010 15:40:42 GMT
View Forum Message <> Reply to Message

jeremy_c wrote on Tue, 05 October 2010 00:30luzr wrote on Mon, 04 October 2010 14:28
Well, perhaps you could test the svn access rights and provide this change (and to reference example too)?

Made the change. moved the code and when I attempted to commit I got in the SVN output:

```
Deleting    Projects\UppSrc\uppsrc\ScgiServer
Adding      Projects\UppSrc\uppsrc\Web\ScgiServer.cpp
Adding      Projects\UppSrc\uppsrc\Web\ScgiServer.h
Sending      Projects\UppSrc\uppsrc\Web\Web.upp
```

Sending Projects\UppSrc\uppsrc\Web\init
Transmitting file data ..svn: Commit failed (details follow):
svn: Access denied

TheIDE then promptly quit on me w/a Windows "stopped working" dialog.

Jeremy

Hi, Jeremy... did you solve the issue ? I wish to try your reference example, but it can't find ScgiServer in web package.

Ciao

Max

Subject: Re: SCGI Class
Posted by jeremy_c on Fri, 08 Oct 2010 12:14:49 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Thu, 07 October 2010 11:40
Hi, Jeremy... did you solve the issue ? I wish to try your reference example, but it can't find ScgiServer in web package.

No, I did not. I sent Luzr a PM w/no response yet. Sorry about this!

Jeremy

Subject: Re: SCGI Class
Posted by mdelfede on Fri, 08 Oct 2010 13:57:27 GMT
View Forum Message <> Reply to Message

NP, Jeremy, I adapted reference app to include SgiServer package instead of web and it works perfectly

Ciao

Max

Subject: Re: SCGI Class
Posted by Mindtraveller on Wed, 16 Mar 2011 20:07:31 GMT
View Forum Message <> Reply to Message

Recently I've tested ScgiServer class inside FastCGI application under FreeBSD 8 with nginx 0.7.67.
Nginx refused to accept data with following error:
upstream sent unsupported FastCGI protocol version: 72 while reading response header from upstream
I googled and found that FastCGI uses some transport protocol with binary header:
http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S3.1

Then I looked inside ScgiServer source and found no support for this binary header, which is strange.

Also I had exceptions when no QUERY_STRING or REQUEST_METHOD was defined in user's request. This is a potential problem, so I propose change serverVars.Get("...") calls with serverVars.GetAdd("...", "")

Subject: Re: SCGI Class
Posted by zsolt on Wed, 16 Mar 2011 21:06:21 GMT
View Forum Message <> Reply to Message

SCGI and FastCGI are different things. They are not compatible.

Subject: Re: SCGI Class
Posted by Mindtraveller on Wed, 16 Mar 2011 22:41:35 GMT
View Forum Message <> Reply to Message

shame on me

Subject: Re: SCGI Class
Posted by Mindtraveller on Thu, 24 Mar 2011 21:44:19 GMT
View Forum Message <> Reply to Message

I plan using SCGI server class for potentially high-load service and so I make some tests first.
Simple example is created:#include <Core/Core.h>
#include <Web/Web.h>
using namespace Upp;

```
class MyScgiServer : public ScgiServer
{
public:
 MyScgiServer() :ScgiServer(7000) {}
```

```
 virtual void OnRequest()
 {
  Write("HTTP/1.0 200 OK\r\nContent-Type: text/plain\r\n\r\nHello, world!");
 }

private:
};

CONSOLE_APP_MAIN
{
 MyScgiServer server;
 server.Run();
 Cout() << "cleanup exit";
}
```

From the first glance, it works well, at least in junction with nginx/mod_scgi. You may even try no-so-high-load tests which will give you good results:

Quote:$ ab -t 10 -c 8 localhost:7000/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 5000 requests
Completed 10000 requests
Completed 15000 requests
Completed 20000 requests
Completed 25000 requests
Completed 30000 requests
Completed 35000 requests
Completed 40000 requests
Completed 45000 requests
Completed 50000 requests
Finished 50000 requests

Server Software:
Server Hostname: localhost
Server Port: 7000

Document Path: /
Document Length: 13 bytes

Concurrency Level: 8
Time taken for tests: 6.806 seconds
Complete requests: 50000
Failed requests: 0
Write errors: 0

Total transferred: 2900000 bytes
HTML transferred: 650000 bytes
Requests per second: 7346.41 [#/sec] (mean)
Time per request: 1.089 [ms] (mean)
Time per request: 0.136 [ms] (mean, across all concurrent requests)
Transfer rate: 416.11 [Kbytes/sec] received

Connection Times (ms)
              min mean[+/-sd] median   max
Connect: 0   0   0.3   0   18
Processing: 0   1   1.1   1   48
Waiting: 0   1   0.8   1   46
Total: 0   1   1.2   1   56

Percentage of the requests served within a certain time (ms)
  50%   1
  66%   1
  75%   1
  80%   1
  90%   1
  95%   1
  98%   2
  99%   2
 100%   56 (longest request)

The bad thing is that if you try to test with concurrency > 10, your test will fail:
Quote:$ ab -t 10 -c 9 localhost:7000/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
apr_socket_recv: Connection reset by peer (54)
Total of 13 requests completed

I started investigation on the source of the problem. If you have any clues, you are welcome.

Subject: Re: SCGI Class
Posted by Mindtraveller on Fri, 25 Mar 2011 15:32:49 GMT
View Forum Message <> Reply to Message

OK, the problem is solved for now.
To support multiple connections on high-load projects we just need to increase listen_count parameter for server socket. So I propose a little update to ScgiServer class:
void ScgiServer::Run(int listenCount /*= 10*/)

```
{
 ServerSocket(serverSock, port, false, listenCount);
 ...
```

That's it. With this little patch scgi server withListenCount = 10000 handles more than 1500 connection requests per second and processing more than 8000 sessions simultaniously.

Subject: Re: SCGI Class
Posted by mirek on Fri, 25 Mar 2011 17:25:34 GMT
View Forum Message <> Reply to Message

Makes sense, patch applied, thank you!

Subject: Re: SCGI Class
Posted by zsolt on Fri, 25 Mar 2011 21:55:59 GMT
View Forum Message <> Reply to Message

BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.

Subject: Re: SCGI Class
Posted by mirek on Fri, 25 Mar 2011 22:56:54 GMT
View Forum Message <> Reply to Message

zsolt wrote on Fri, 25 March 2011 17:55BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.

Actually, I would guess that on that much traffic, you simply run out of TCP/IP per port limit... (which is in thousands per second max).

Subject: Re: SCGI Class

Posted by Mindtraveller on Sat, 26 Mar 2011 22:00:20 GMT
View Forum Message <> Reply to Message

zsolt wrote on Sat, 26 March 2011 00:55BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.
What if in the main cycle the the newly created client socket is processed in another thread while server socket is free to accept more connections?

```
while (run)
{ if  (serverSocket.Accept(&clientSocket)
  {
    GetThreadFromPoolAndProcess(clientSocket);
  }
}
```

Subject: Re: SCGI Class
Posted by nlneilson on Sun, 27 Mar 2011 04:54:28 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 26 March 2011 23:00What if in the main cycle the the newly created client socket is processed in another thread while server socket is free to accept more connections?

```
while (run)
{ if  (serverSocket.Accept(&clientSocket)
  {
    GetThreadFromPoolAndProcess(clientSocket);
  }
}
```

That seems like the logical way.
My Upp apps as clients interact with Java apps/server.
New thread/socket for each client.

```
 // The body of the server thread. Loop forever, listening for and
 // accepting connections from clients. For each connection,
 // create a Connection object to handle communication through the
 // new Socket.
 public void run() {
  try {
   while (true) {
    Socket client_socket = listen_socket.accept();
    Connection c = new Connection(client_socket);
```

```
   }
  } catch (IOException e) {
   fail(e, "Exception while listening for connections");
  }
 }
}
```

I did not write the server code but just followed an example.

Neil

Subject: Re: SCGI Class
Posted by mirek on Sun, 27 Mar 2011 07:59:24 GMT
View Forum Message <> Reply to Message

Well, I guess that in that case it would likely made sense to break Run into smaller parts, just like it is done with XmlRpc, so that client code could arrange any complicated usage scenario it wishes...

Subject: Re: SCGI Class
Posted by mirek on Sun, 27 Mar 2011 08:02:18 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 26 March 2011 18:00zsolt wrote on Sat, 26 March 2011 00:55BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.
What if in the main cycle the the newly created client socket is processed in another thread while server socket is free to accept more connections?

```
while (run)
{ if  (serverSocket.Accept(&clientSocket)
  {
    GetThreadFromPoolAndProcess(clientSocket);
  }
}
```

Actually, this can be quite nicely with something like

```
while(run)
{ if  (serverSocket.Accept(&clientSocket) {
```

```
    DoWork();
  }
}
```

and then simply starting several threads to run this loop (with single clientSocket). As accept is reentrant and MT safe, it would return only for single thread running, thus managing the thread pool.

Subject: Re: SCGI Class
Posted by zsolt on Sun, 27 Mar 2011 20:51:58 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 27 March 2011 10:02Mindtraveller wrote on Sat, 26 March 2011 18:00zsolt wrote on Sat, 26 March 2011 00:55BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.
What if in the main cycle the the newly created client socket is processed in another thread while server socket is free to accept more connections?

```
while (run)
{ if  (serverSocket.Accept(&clientSocket)
  {
    GetThreadFromPoolAndProcess(clientSocket);
  }
}
```

Actually, this can be quite nicely with something like

```
while(run)
{ if  (serverSocket.Accept(&clientSocket) {
    DoWork();
  }
}
```

and then simply starting several threads to run this loop (with single clientSocket). As accept is reentrant and MT safe, it would return only for single thread running, thus managing the thread pool.

This is a very elegant way to solve the problem using threads.

My only problem with MT in situations like this is, that a few things in Upp are global, such as lenguage-dependant setups (e.g. date or number formatting).

It is a common requirement now from a web based app to show content in the user's language, so I think multi process arrangement would be better.

Or is it somehow possible to make these global Upp settings thread local?

Subject: Re: SCGI Class
Posted by mirek on Mon, 28 Mar 2011 06:54:41 GMT
View Forum Message <> Reply to Message

zsolt wrote on Sun, 27 March 2011 16:51mirek wrote on Sun, 27 March 2011 10:02Mindtraveller wrote on Sat, 26 March 2011 18:00zsolt wrote on Sat, 26 March 2011 00:55BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.
What if in the main cycle the the newly created client socket is processed in another thread while server socket is free to accept more connections?

```
while (run)
{ if  (serverSocket.Accept(&clientSocket)
  {
    GetThreadFromPoolAndProcess(clientSocket);
  }
}
```

Actually, this can be quite nicely with something like

```
while(run)
{ if  (serverSocket.Accept(&clientSocket) {
     DoWork();
  }
}
```

and then simply starting several threads to run this loop (with single clientSocket). As accept is reentrant and MT safe, it would return only for single thread running, thus managing the thread pool.

This is a very elegant way to solve the problem using threads.

My only problem with MT in situations like this is, that a few things in Upp are global, such as lenguage-dependant setups (e.g. date or number formatting).

It is a common requirement now from a web based app to show content in the user's language, so I think multi process arrangement would be better.

Or is it somehow possible to make these global Upp settings thread local?

I guess that while there is no direct support, it is in fact quite easy to do in app, as there is:

```
String     GetLngString(int lang, const char *id);
```

-> #undef t_, replace it with your own version which is thread local... Or perhaps just use 'lang' parameter.

Subject: Re: SCGI Class
Posted by zsolt on Tue, 29 Mar 2011 19:27:11 GMT
View Forum Message <> Reply to Message

Thanks, but I think, there are problems whith some code in Core like this:

```
static char s_date_format[64] = "%2:02d/%3:02d/%1:4d";

void   SetDateFormat(const char *fmt)
{
 strncpy(s_date_format, fmt, 63);
}

String   Format(Date date) {
 String  s;
 if(IsNull(date))
  return String();
 return Format(s_date_format, date.year, date.month, date.day, DayOfWeek(date));
}
```

Subject: Re: SCGI Class
Posted by mdelfede on Sat, 11 Feb 2012 18:34:14 GMT
View Forum Message <> Reply to Message

I'd really like to have a working MT example on this.....

Max