Subject: Web framework....
Posted by mirek on Fri, 25 Nov 2011 20:37:40 GMT
View Forum Message <> Reply to Message

So, the next big thing is web framework.

Here are some thoughts:

- primary inspiration Django
  - clean URLs, view=function bind to view, templates have special language
  - we will have more rich language for templates (Django templates sometimes too primitive for the task - first hand info from web designers:)

- other interesting frameworks to check: RoR
  - for C++ CppCMS, TnTNet (but we will "JIT" compile templates, more practical in team development).

- we will attempt to enrich the thing by designing forms using regular Layout designer - should be quite possible...

- primary external interface - http (and either use 'naked' or with apache with mod_proxy)

- preferred multithreaded apps (single process, many threads to serve connections) - advantage is to store some common cached data; alternative is preforked processes

Please feel free to comment, request, suggest...

...to be continued

Subject: Re: Web framework....
Posted by chickenk on Sat, 26 Nov 2011 10:55:13 GMT
View Forum Message <> Reply to Message

comments about things to check :

- for multithread, comm, etc: look at ZeroMQ, and the bindings already done in bazaar
- look at mongrel2 too, it's a webserver not a web framework, but it makes good use of zeromq
- another nice framework: CodeIgniter in PHP

that's all for now...

Subject: Re: Web framework....
Posted by dolik.rce on Sat, 26 Nov 2011 11:22:08 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 25 November 2011 21:37- preferred multithreaded apps (single process, many threads to serve connections) - advantage is to store some common cached data; alternative is preforked processes
May I ask you why you prefer MT over preforked processes? At work we use custom preforked server with shared memory. It proved very stable, since a problem (fatal error, lock-up etc.) in single child doesn't take down entire server. Also it is possible to limit each child to certain number of requests or set it some time of life, after which it is killed. This can be helpful in case of leaking apps (we don't use U++, so leaks happen from time to time).

Of course I see that MT makes other things simple. So maybe having both options available is the right way to go...

Best regards,
Honza

Subject: Re: Web framework....
Posted by mirek on Sat, 26 Nov 2011 13:24:10 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 26 November 2011 06:22mirek wrote on Fri, 25 November 2011 21:37- preferred multithreaded apps (single process, many threads to serve connections) - advantage is to store some common cached data; alternative is preforked processes
May I ask you why you prefer MT over preforked processes? At work we use custom preforked server with shared memory.

I am well aware about prefork advantages and in fact, we do the same.

However, right now I have some doubts about using shared memory for common data.

E.g. it looks like we are going to have 'jit' compiled html templates into 'functional nodes' (as shown in another thread). I am afraid that it might be a bit hard to use shared memory for caching this. Basically, I believe that single heap makes these things much simpler.

Quote:
Of course I see that MT makes other things simple. So maybe having both options available is the right way to go...

Sure, for web app developer, this will not matter too much. It will be possible to switch to MT or to preforked with ease, but with preforked, things will be duplicated in memory.

Mirek

Subject: Re: Web framework....

Posted by lectus on Sun, 27 Nov 2011 12:21:43 GMT
View Forum Message <> Reply to Message

Mirek,

If it's possible would be great if the same U++ code (including layouts) would run on the web with a web interface.

One application I want you to take a look: qBittorrent. It has a normal GUI and in the options you can activate a web GUI you access by pointing your browser to your IP address, this way you can operate the application from distance.

This feature above doesn't imply a full web framework in the same sense as Django, but a way to deliver web interfaces (better if using layout editor).

I hope you like my suggestion.

Thanks

Subject: Re: Web framework....
Posted by mr_ped on Mon, 28 Nov 2011 08:38:59 GMT
View Forum Message <> Reply to Message

I use Nette framework for PHP, and I'm quite happy about it's template language and overall library interface.
I will surely compare it with the results of U++ development.

Subject: Re: Web framework....
Posted by cbpporter on Wed, 30 Nov 2011 08:16:08 GMT
View Forum Message <> Reply to Message

If you seek inspiration from the Java world, I think the best candidate would be JSF (Java Server Faces). It is the only painless Java frameworks that I have encountered. If we could "compile" the views, it could turn to be quite pleasant to use.

RoR is also very pleasant. I have no experience with Django, but it probably shares at least a few principles with RoR.

Subject: Re: Web framework....
Posted by lectus on Fri, 09 Dec 2011 15:49:37 GMT
View Forum Message <> Reply to Message

Here's a good idea:

* HTML 5 GUI designer in the layout editor.
* Use U++ classes to handle the back end.
* Connect both through HTML 5 WebSocket and U++ Sockets.

Basically the framework should make it easy to connect these sockets and this should not be visible to the programmer (wrapped in classes).

So, we can implement MVC for the web:
Model: U++ handling Sqlite, MSSQL, Oracle, MySql,etc
View: HTML5 UI designer on the IDE.
Controller: C++ code using U++ NTL and all classes.

Example of HTML 5 WebSocket:

```
socket= new WebSocket('ws://www.example.com:8000/somesocket');
socket.onopen= function() {
   socket.send('hello');
};
socket.onmessage= function(s) {
   alert('got reply '+s);
};
```

This way U++ can run on servers while HTML UI front ends will be available for Desktops, Notebooks, Netbooks, Tablets and Smartphones.

Subject: Re: Web framework....
Posted by mirek on Fri, 09 Dec 2011 16:04:07 GMT
View Forum Message <> Reply to Message

lectus wrote on Fri, 09 December 2011 10:49Here's a good idea:

* HTML 5 GUI designer in the layout editor.
* Use U++ classes to handle the back end.
* Connect both through HTML 5 WebSocket and U++ Sockets.

Basically the framework should make it easy to connect these sockets and this should not be visible to the programmer (wrapped in classes).

So, we can implement MVC for the web:
Model: U++ handling Sqlite, MSSQL, Oracle, MySql,etc
View: HTML5 UI designer on the IDE.
Controller: C++ code using U++ NTL and all classes.

Example of HTML 5 WebSocket:

```
socket= new WebSocket('ws://www.example.com:8000/somesocket');
socket.onopen= function() {
   socket.send('hello');
};
```

```
socket.onmessage= function(s) {
   alert('got reply '+s);
};
```

This way U++ can run on servers while HTML UI front ends will be available for Desktops, Notebooks, Netbooks, Tablets and Smartphones.

Like Wt?

Well, that is planned like one possible high-level paradigm. However, the core functionality will be much more down to the earth.

Subject: Re: Web framework....
Posted by lectus on Mon, 12 Dec 2011 14:05:24 GMT
View Forum Message <> Reply to Message

That's nice.

So what are the plans of this web framework? Are there already a roadmap?

Subject: Re: Web framework....
Posted by mirek on Mon, 12 Dec 2011 19:00:05 GMT
View Forum Message <> Reply to Message

lectus wrote on Mon, 12 December 2011 09:05That's nice.

So what are the plans of this web framework? Are there already a roadmap?

It is already in development. Current phase is 'experiment', should be ended by the end of year.

You can watch the progress in svn sandbox.
