Lisp in Summer Projects Submission

Submission Date	2013-10-03 00:47:13
Full Name	Mu Lei known as NalaGinrut
Zip/postal code	5108048
Country	China
Project Name	Artanis
Type of software	library
General category	framework
LISP dialect	Scheme (MIT, Gambit, Chicken, Guile ...)
GitHub URL	https://github.com/NalaGinrut
Did you start this project?	Yes, all the code is written by me
Project Description	I want to describe my project in this form.
Purpose	Artanis is aim to the lightweight and fastest web-framework of Scheme, which is written with Guile-Scheme.
Function	Users may use this web-framework to build site quickly, and it can work alone or cooperate with Nginx. Nowadays, users may want to build small web-app easily, Artanis provides convenient interface for that.
Motivation	The original idea is to build a full-stack web-framework of Guile, since Guile is the official extension language of GNU. It's reasonable to use Guile to build GNU projects site. But it's a big project, Artanis would be the core of it. Anyway, this project is to prove that Guile is a very nice language for web programming.
Audience	Anyone who wants to use Scheme to build site or web-app. Students who has finished SICP may try Artanis for some cool ideas. The guys who interested in high performance web developing in the future.

Methodology

The reason why I implemented Artansi with many Guile specific features is because of the efficiency. One of the main aim of Artanis is 'the fastest'. But users may try any RnRS compatible things in the framework.

At present, this web-framework doesn't pursue very neat/academic algorithm or other pinciple. It just implements HTTP and other RFCs rules with mature technics.
Now the features included URL-remapping, template, ssql(s-expr like SQL), DB support(mysql/postgresql/sqlite3), cookie, MIME, sendmail, multipart/form-data(file upload), websocket(not done yet).

* The http-server in Artanis is the inner server of Guile, which can hold 10k request/s in my benchmark.

* URL-remapping uses a hashtable to hold all rules users registered, and every url of requests will be checked if it hits one of the rules, then try to run the handler of that rule.

* Artanis has two types of template: 1. SXML based template, users may write HTML as s-expr, then convert to HTML; 2. Embedded template, users may embed Scheme code within HTML. Artanis takes advantage of regexp to parse the template text, and treat HTML part as common print statment, for Scheme code part, it just appended to the text of result. Finally, it call eval-string to run it to generate HTML dynamically. Moreover, it uses local-eval to eval the expr which was defined out of the template file. Then it's unecessary to define all variables in template.
** now embedded template engine is very fast, a 130K length template file could be redered in 0.5s, the old engine takes 6.5s!

* DB support takes advantage of guile-dbi which support mysql/postgresql/sqlite3. The next step is to provide ORM for that.

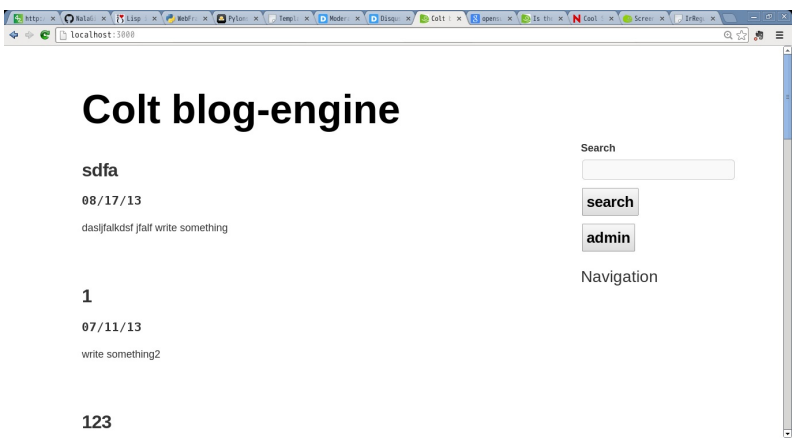
* sendmail is just a wrapper of commandline 'sendmail', it depends on users configuration of sendmail/postfix.

The social-challenges was faced that few people get interested in writing web site with Scheme/Lisp. But many Schemer/Lisper shows great interesting of it. One of the way to describe it I think, is to provide a easy-to-try thing to let them try Scheme/Lisp for some cool ideas.

Conclusion

Scheme is a very suitable language for web programming. Moreover, Guile is becoming efficient for implementing web-app or servers. I listed these reasons for that:

* Guile has unified server interface, so we don't need the extra things like WSGI(Python) or Rack(Ruby).
* Guile will have AOT compiler in the next big release. It may means 85% execution efficient could be increased in principle.
* Guile may change all the ports to support non-block, which is easier to implement async I/O.
* Guile has well-intergrated delimited-continuation, which could be used to implement co-routine. Another project of mine is to write actor-model based servers to provide high performance concurrency web-server. Fortunately, Artanis can change server as you wish.

Build Instructions	Artanis doesn't need to be installed, you may try examples/ directly. But if you want to try blog.scm or other databased related thing. Please install guile-dbi first: http://home.gna.org/guile-dbi/ And make sure your database was well-configured.
Test Instructions	* hello world cd examples/ ./test.scm * a simple blog test mysql -u root < blogdb.mysql ./blog2.scm others are likewise.
Execution Instructions	Please read Teset instructions.
Describe any bugs or caveats	* Now I haven't finished all the test-cases * websocket.scm is not done yet, so don't try it. * upload.scm is based on irregex, it can't handle too big files. The fix needs to modify the server reader, it's on my TODO list. * sendmail wasn't tested since it's submited juse before the deadline. And it depends on user's sendmail/postfix configuration. * base64/md5/sha-1 was implemented with Scheme, I borrowed from other projects. Anyway, it's not fast. But we may expect Guile's AOT to give them power. * Auth and session is suck, I'm struggling against this. * config.scm is a temporary design, which hard coded config items. I will change it to read /etc/artanis/serv.conf * HEAD method was handed like "GET then drop return body", which sounds suck. Althoug it's OK for dynamic pages, it's sack for static pages except you co-work with Nginx, one may handle all static pages Nginx. The proper way maybe checkout if a request URL is a dynamic rules rather than static page. Since rules table is hashtable, it sounds fair. * Unfortunately, I just found a bug in embedded template when testing blog2.scm, it's too late to fix it before submit, But I fixed it in master branch. :-(
Screen shots	

	artanis.png
Official	I have read rules and have abided by them. I am 18 years of age or older. I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.