YesWeb: A Concise Web Development Framework Based on Haskell

Xiao Liu[1*], Yeoneo Kim[2], Sugwoo Byun[3], and GyunWoo[4]

[1, 2, 4]Department of Electrical Computer Engineering, Pusan National University, Busan, Korea
[3]Department of Computer Engineering, Kyungsung University, Busan, Korea
[*Email*:{ liuxiao, yeoneo, woogyun}@pnu.edu[1,2,4],swbyun@ks.ac.kr[3]]

Abstract

In this paper, we present YesWeb, an abstraction of Yesod, both of which are web-development frameworks written in Haskell. It allows user to construct a web system in a more concise and convenient way and supports the high performance of parallel computing in multi-core architecture. YesWeb exploits nice features of Haskell such as type-safety, modularity, and parallelism and concurrency. We demonstrate an example and discuss its performance test and analysis.

Index Terms: Haskell, web development, parallel computing

I. INTRODUCTION

Constructing a web application could require considerable efforts and time [1], for practical purpose one often develops it by using web frameworks. In this paper, we present YesWeb, a compatible extension of Yesod, both of which are web-development frameworks written in pure functional language Haskell. Yesod includes most features of web components such as web site template design, web page routing, data storage, sessions, data storage and authentication. YesWeb provides more abstract programming environment than Yesod so that user could construct a web system in a more concise and convenient way. YesWeb also supports the high performance of parallel computing in multi-core architecture which can process large amount of data stream. It can be used in both front-end and back-end applications.

Much of nice features of YesWeb and Yesod ows Haskell properties such as static type systems, pure and monadic computation, parallelism and light-weight threads, data-stream processing based on lazy evaluation, and etc. Type safety guarantees that web applications run correctly once it is passed by the compile, which could save a lot of test cost.

This paper is organized as follows. Section 2 introduces related works such as Yesod framework and its web server named Warp. Section 3 presents the architecture of YesWeb framework. Section 4 demonstrates an implementation of web application witch developed based on YesWeb framework named YesWeb-Blog. Section 5 tests the parallelism and concurrency of YesWeb-Blog to verify the feasibility, practicability and functionality of YesWeb framework. Section 6 concludes

II. Related work

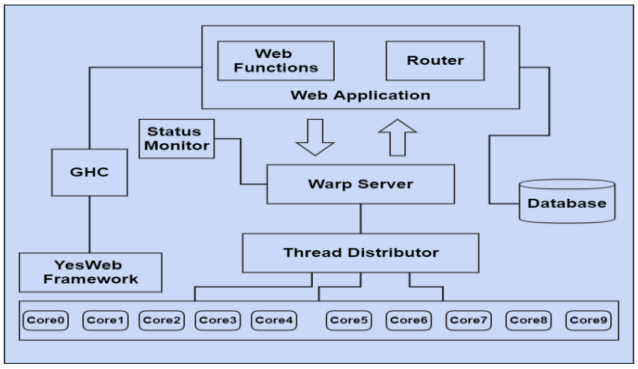

Fig. 1. The architecture of YesWeb framework.

Haskell is well-suited for parallel programming due to its purity of functional programing. As it uses no mutable variables, i.e. no assignment statements, every expression has a unique value independently in the order of evaluation. From this point of view, one can say Haskell has a property of natural parallelism.

Recent versions of GHC, a Haskell compiler, support the feature of parallel computation. Its virtual machine manages its own threads independently from OS threads, light-weight thread fork, thread scheduling, and profiling. Using the tool ThreadScope, one can analyze graphically the statistics produced from executing programs and debug the parallel performance [3]. One can code parallel programs conveniently using the parallel constructs such as `par`, `pseq`, and etc. The overhead of GHC threads is extremely low so that GHC is very powerful in the large amount of event processing. This feature is beneficial in the efficiency of web applications [4].

In web application frameworks, Yesod includes most of necessary web development functionalities of layout design, function building. and deploying. Using Yesod, one doesn't have to write all of the HTML, CSS and JavaScript code to print the layout of web. This framework includes several templates to generate various types of code: Shakespearean for Haskell code, Hamlet for HTML, Lucius and Cassius for CSS, and Julius for JavaScript [5]. All of these template languages are quite easy to use and the type safety of Haskell makes sure that all of generated code are correct.

Warp is an outstanding Haskell web server used not only for Yesod application but also for other Haskell web applications. Especially the seamless integration of Yesod with Warp makes the best of performance. Only little configuration is required for deploying a web application on Warp and running it, and this efficient approach could reduce lots of time and manpower.

III. Architecture of YesWeb

YesWeb, based on Yesod, is implemented to enable users to develop web system more conveniently and to improve the flexibility and scalability. YesWeb has modular architecture as shown in Fig. 1. It provides more concise syntax to users. An YesWeb application is translated into web functions and routers, which are compiled by GHC into executable file.

Requests occurred in web application are processed by Warp server running on multicore architectures. The current processing status of Warp is displayed on the administrator's monitor. Requests sent from web application to Warp server are assigned by the thread distributor.

IV. Implementation of YesWeb: YesWeb-Blog

Fig. 2 shows an example of YesWeb blog and its translated version of Yesod. Generating HTML code for rendering web page on browser, users can choose one of two alternatives – Haskell programming style or traditional style. As YesWeb translates the code of traditional style is translated into Haskell code, non-Haskell programmers also can use YesWeb. The description of traditional style is more concise than Haskell code. Fig. 2-1 shows that 4 lines of codes is used to describe URL, available port for accessing, its manager ID which can be passed by the authentication function, and its posting function to publish articles. This description is translated into Yesod-based Haskell code in Fig 2-2, which is more verbose.

```
$doctype 5
yesweb-blog.url=pl.pusan.ac.kr:5111
yesweb-blog.pages.Auth.Manager = YesWeb.Auth("pllab@pusan.ac.kr");
yesweb-blog.pages.Post = YesWeb.SendInfo(title, article);
```

(1): HTML code generation in YesWeb

```
instance Yesod YesBlog where
    approot = ApprootStatic "http://pl.pusan.ac.kr:5111"
isManager visitor = visitorMailAddr visitor == "pllab@pnu.ac.kr"
postListR :: Handler Html
postListR = do
    ((res, articleWidget), enctype) <- runFormPost articleF
    case res of
        FormSuccess article -> do
            articleId <- runDB $ insert article
            setMessageI $ MsgEntryCreated $ articleHeading article
            redirect $ ListEntryR articleId
        _ -> defaultLayout $ do
            setTitleI MsgPleaseCorrectEntry
```

(2): HTML code generation in Yesod

Fig. 2. Comparison of HTML code generation between YesWeb and Yesod.

V. Performance of YesWeb-Blog

The Fig. 3 presents the benchmark variation of YesWeb-Blog with increasing processor cores. The X-axis of the histogram stands for the number of cores and the Y-axis means the number of hitting times on YesWeb-Blog in 60 seconds. The graph in Fig. 3 shows two cases: 500 and 1000 concurrent users. YesWeb-Blog processed similar number of accesses from 500 users no matter how many cores are processing. But when there were 1000 users, YesWeb-Blog only was able to process requests on more than 6 cores. According to our other research, we suspect the performance degradation is caused by the garbage collection mechanism of GHC compiler, which spends much time in 10 or more cores.

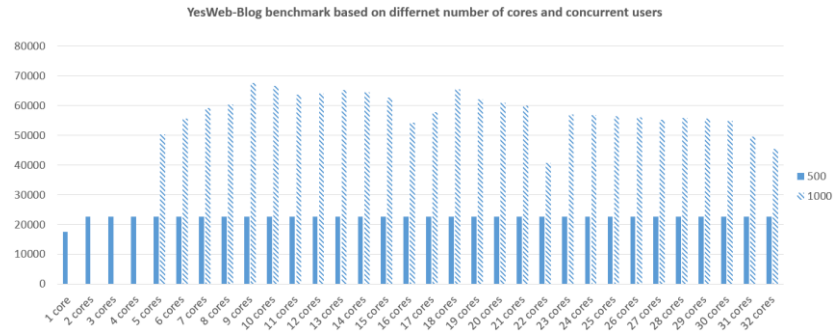

Fig. 3. YesWeb-Blog benchmark.

VI. Conclusion

Most of web application frameworks and web servers have issues to improve their efficiency and compatibility. YesWeb, discussed in this paper, shows satisfactory results in this issues. One can use YesWeb in the whole process of web application covering front-end design, data handling, requests processing and responding. YesWeb is a compatible extension of Yesod framework to provide more concise programming syntax and archive parallelism and concurrency. The concise programming syntax of YesWeb is productive. YesWeb-Blog, an online web blog application, can be written only in 220 lines of Haskell code. The performance of YesWeb is also satisfactory in up to 21 cores of parallel architecture even though it needs improvement for more cores. The garbage collection issues to improve the scalability remains as the future work.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0101-16-0644, Research on High Performance and Scalable Manycore Operating System).

REFERENCES

[1] J. Chao, K. Parker, B. Davey, Navigating the framework jungle for teaching web application development[J]. *Issues in Informing Science and Information Technology*, 10: 95-109, 2013.

[2] G. Hutton, *Programming in Haskell*. Cambridge University Press, 2007.

[3] S. Marlow, *Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming*, O'Reilly Media, Inc., 2013.

[4] S. Marlow, P. Maier, H. W. Loidl, et al., Seq no more: better strategies for parallel Haskell, *ACM Sigplan Notices*. ACM, 45(11): 91-102, 2010.

[5] M. Snoyman, *Developing web applications with Haskell and Yesod*, O'Reilly Media, Inc., 2012.