

Исследование моделей параллельных вычислений веб-серверов

Кочетов Павел Сергеевич

Нижегородский государственный университет им. Н.И. Лобачевского

Магистрант

Штанюк Антон Александрович

Нижегородский государственный университет им. Н.И. Лобачевского

доцент

Аннотация

В статье рассматривается исследование моделей параллельных вычислений веб-серверов. Приводятся результаты тестирования разработанного ранее веб-сервера на языке программирования Erlang, анализируются результаты тестирования его многопоточности с помощью системы Tsung и сравниваются с результатами тестирования веб-сервера Apache с MPM Worker.

Ключевые слова: сервер, тестирование веб-сервера, модель параллельных вычислений, модель акторов, язык программирования Erlang.

The study of models of concurrent computing of web-servers

Kochetov Pavel

Nizhny Novgorod State University

Student

Shtanyuk Anton

Nizhny Novgorod State University

associate professor

Abstract

The article discusses the study of models of concurrent computing of web servers. The results of testing the previously developed web server in the Erlang programming language are presented, the results of testing its multi-threading with the Tsung system are analyzed and compared with the results of testing the Apache web server with MPM Worker.

Keywords: server, testing the web server, model of concurrent computing, Erlang programming language.

Введение

Как известно, веб-технологии в настоящее время стремительно развиваются. Веб-серверы перестают справляться с нагрузкой из-за

огромного числа одновременных пользователей. Для решения подобных проблем используются различные модели параллельных вычислений, каждая из которых подходит под определенный тип задач. Некоторые модели параллельных вычислений используются в веб-серверах как модули, например, Apache MPM (Multi-Processing Modules): PreFork, Worker, Event. Также существуют модели, которые являются основой языка программирования, на котором реализован веб-сервер (Erlang).

Цель данной работы - исследование моделей параллельных вычислений веб-серверов.

Для достижения указанной цели были выполнены следующие задачи:

1. Обзор существующих моделей параллельных вычислений веб-серверов.
2. Выбор модели с наилучшей производительностью.
3. Тестирование выбранной модели (Apache) и модели акторов (Erlang), анализ полученных результатов.

Обзор моделей параллельных вычислений веб-серверов

В настоящее время существуют различные модели параллельных вычислений, используемые в веб-серверах. Рассмотрим основные модели параллельных вычислений веб-сервера Apache (в Apache они реализованы посредством модулей MPM).

PreFork

При использовании для сервера Apache MPM PreFork запускается отдельный процесс на каждый запрос. При этом каждый процесс одновременно обрабатывает только 1 поток на одно соединение. Т.к. **PreFork** определяет фиксированное количество процессов сразу при инициализации сервера, то такой вариант является наиболее производительным среди всех **MPM**. Однако при большом количестве процессов будут происходить задержки их выполнения [1].

Преимущества:

- Безопасность (в связи с изолированностью каждого процесса)

Недостатки:

- Большой объем используемой оперативной памяти при большом числе одновременных процессов.

Worker

Apache MPM Worker использует многопоточную обработку запросов, что позволяет увеличить количество обрабатываемых соединений. При использовании **MPM Worker** запускается несколько процессов, каждый из которых запускают несколько потоков. Эти потоки обслуживают запросы

клиентов. При данном подходе используется меньший объем RAM, чем при применении **PreFork**. Так же, при использовании данной модели для получения запроса нужен поток, а не процесс, то есть происходит существенная экономия вычислительных ресурсов [1].

Преимущества:

- Меньший расход оперативной памяти, чем у PreFork.

Недостатки:

- Относительная нестабильность, так как проблемы в одном процессе могут повлиять и на другие процессы.

Event

Apache MPM Event по принципу работы очень похож на **MPM Worker**, однако, есть и отличия. Главное отличие **Event** от **Worker** – Event поддерживает поток для каждого установившегося подключения. При этом он передает потокам запрос только после непосредственного выполнения запроса. Сразу после обработки этого запроса поток освобождается для выполнения следующего запроса. Данная модель хорошо подходит для клиентов, которые поддерживают долгие соединения, не делая при этом частых запросов к серверу. В случае обработки **SSL**-соединений Event ведет себя как **Worker** [1].

Модель акторов в Erlang

Модель акторов - это математическая модель параллельных вычислений, основанная на процессе обмена сообщениями между акторами (примитивами параллельного расчета).

Актор в ответ на полученное сообщение может одновременно:

- Отправлять сообщения другим акторам;
- Создавать новых акторов;
- Выбирать тип используемого поведения для следующего сообщения в свой адрес.

Все указанные выше действия могут быть в случайной последовательности и выполняться параллельно.

В модели акторов отправка и получение сообщений выполняются асинхронно, следовательно, задержек из-за ожидания не возникает и нет необходимости в механизме синхронизации.

Получатели сообщений идентифицируются по «почтовому» адресу. Актор способен взаимодействовать только с акторами, адреса которых он знает заранее, либо из полученных сообщений.

В отличие от других моделей, модель акторов была разработана именно как параллельная модель, последовательность в которой представляет собой особый случай, вытекающий из одновременных вычислений.

Согласно модели акторов, поведение - математическая функция, которая выражает действия актора при обработке получаемого сообщения и определение поведения на обработку следующего сообщения. В модели акторов поведение обеспечивает функционирование математической модели параллелизма [2].

В языке программирования Erlang акторами являются процессы внутри виртуальной машины Erlang VM.

Уникальная легковесная модель Erlang не зависит от операционной системы и способна справляться с огромным числом одновременно запущенных процессов. В Erlang виртуальная машина не создает отдельного потока вычислений операционной системы для каждого процесса, процессы создаются и выполняются внутри самой виртуальной машины и не зависят от операционной системы, в результате чего создание процесса занимает микросекунды и время создания не зависит от числа уже запущенных процессов, поэтому в Erlang процессы называют легковесными [3].

За счет использования модели акторов в Erlang также достигается отказоустойчивость Erlang (при ошибке в одном из легковесных процессов внутри виртуальной машины Erlang VM просто завершит этот процесс и на работоспособности других процессов это не скажется). Поэтому использование веб-серверов, разработанных на Erlang может оказаться весьма эффективным.

Тестирование веб-серверов и анализ полученных результатов

Для проведения сравнительных тестов производительности были выбраны Erlang-сервер [4] и веб-сервер Apache с MPM Worker.

В качестве настроек окружения для тестирования была использована виртуальная машина под управлением CentOS 7 x64, для работы которой выделялось 5 Гб оперативной памяти, 4 ядра процессора AMD A8-6410 и 20Гб на жестком диске.

В ходе проведения тестов многопоточности были выполнены эксперименты для сервера Apache с использованием MPM Worker и веб-сервера, разработанного на Erlang [4].

Для тестирования Erlang-сервера [4] в тестирующей системе была настроена следующая конфигурация:

- Первая фаза – 2 минуты, 500 новых посетителей в секунду
- Вторая фаза – 2 минуты, 1000 новых посетителей в секунду
- Итоговое время выполнения – 4 минуты (2 фазы)

После завершения работы теста были получены следующие результаты.

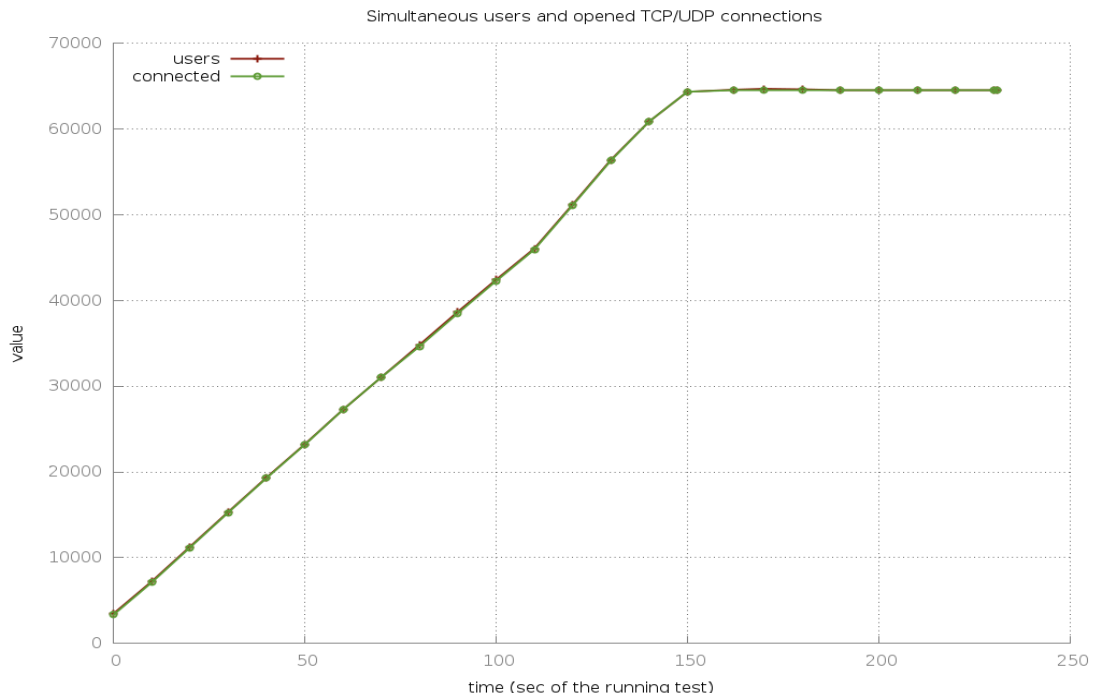


Рисунок 1 - Количество одновременных посетителей и открытых соединений

На данном рисунке видно, как в течение всего времени выполнения теста количество одновременных посетителей растет до 150-й секунды проведения теста, после чего рост прекращается.

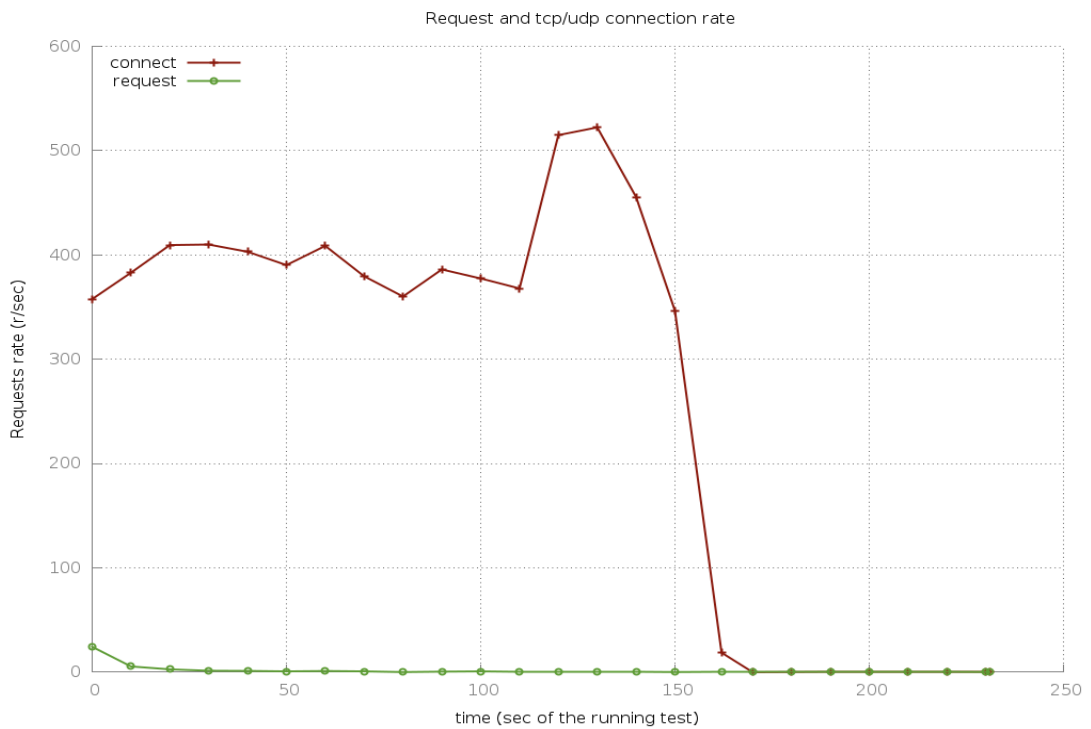


Рисунок 2 - Количество обработанных сервером запросов и подключений к серверу в секунду

На данном рисунке можно увидеть, что количество обработанных запросов в секунду уменьшается, количество подключений к серверу в секунду снижается после 150-й секунды выполнения теста.

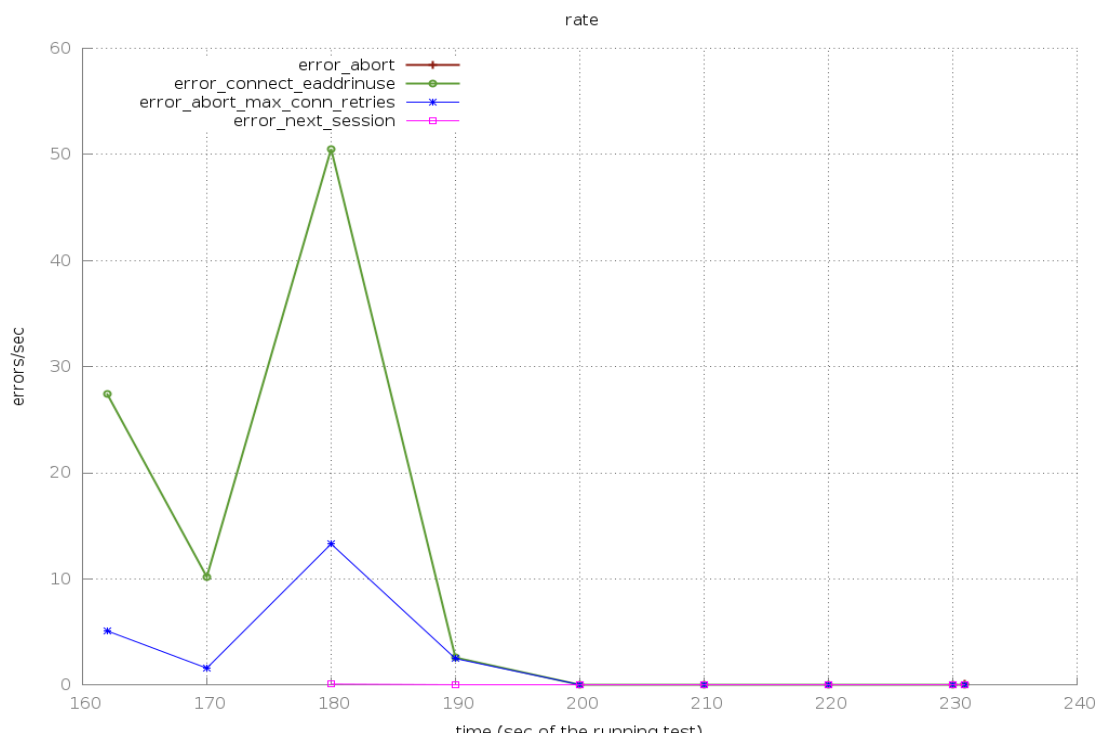


Рисунок 3 - Количество ошибок сервера в секунду

На рисунке 3 заметно, как со 160-й секунды появляются ошибки сервера, то есть веб-сервер не справляется с нагрузкой.

По результатам выполненного теста Erlang-сервера можно сделать следующий вывод: данный сервер способен выдерживать нагрузку около 65000 одновременных подключений пользователей к нему без ошибок в работе.

Далее было проведено тестирование веб-сервера Apache с использованием MPM Worker. Для этого была использована следующая конфигурация тестирующей системы:

- Первая фаза – 2 минуты, 20000 новых посетителей в секунду
- Вторая фаза – 2 минуты, 40000 новых посетителей в секунду
- Итоговое время выполнения – 4 минуты (2 фазы)

После завершения работы теста были получены следующие результаты:

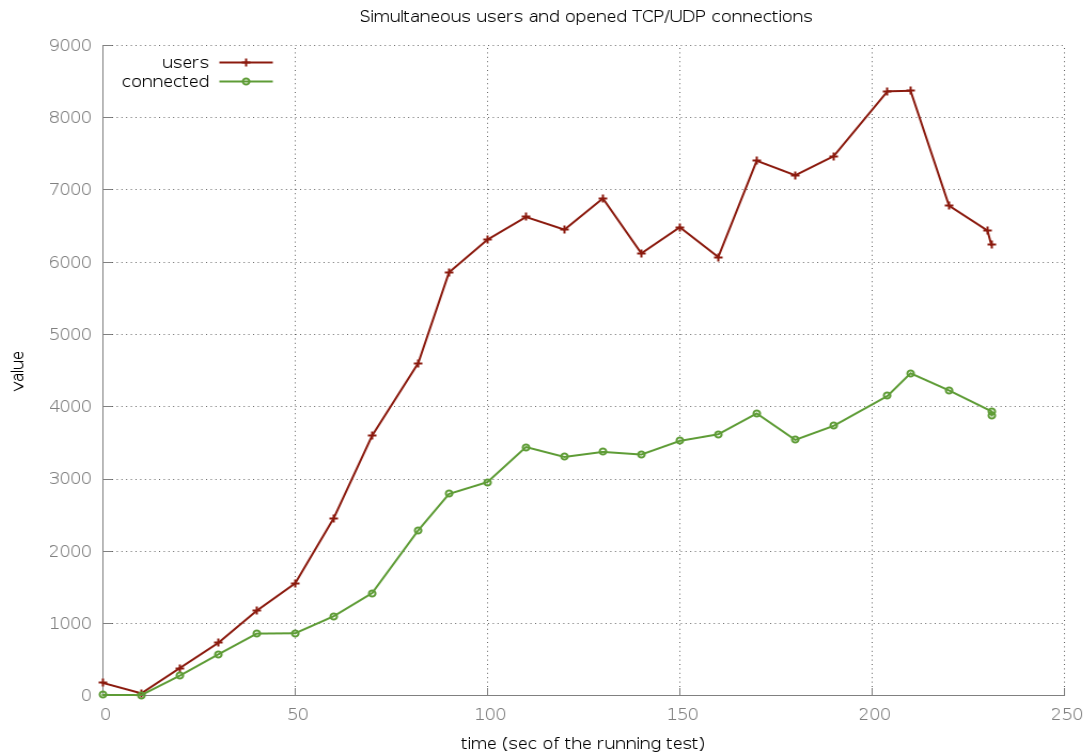


Рисунок 4 - Количество одновременных посетителей и открытых соединений (Apache MPM Worker)

На данном можно наблюдать, как в течение всего времени выполнения теста число подключенных одновременно клиентов увеличивается, но не превышает 4500.

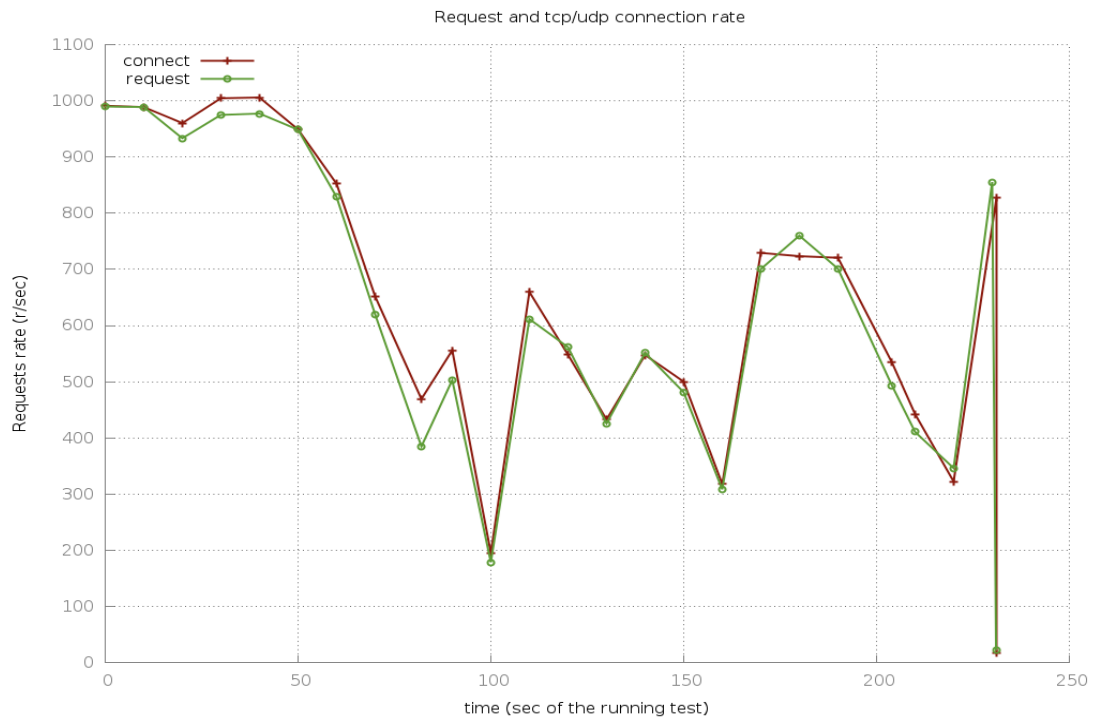


Рисунок 5 - Количество обработанных запросов и подключений к серверу в секунду (Apache MPM Worker)

На данном рисунке можно увидеть, как количество обработанных запросов в секунду после первых 50 секунд выполнения теста уменьшается, затем колеблется в диапазоне 200-800 запросов к серверу в секунду. Аналогичное поведение наблюдается на графике количества подключений к серверу в секунду.

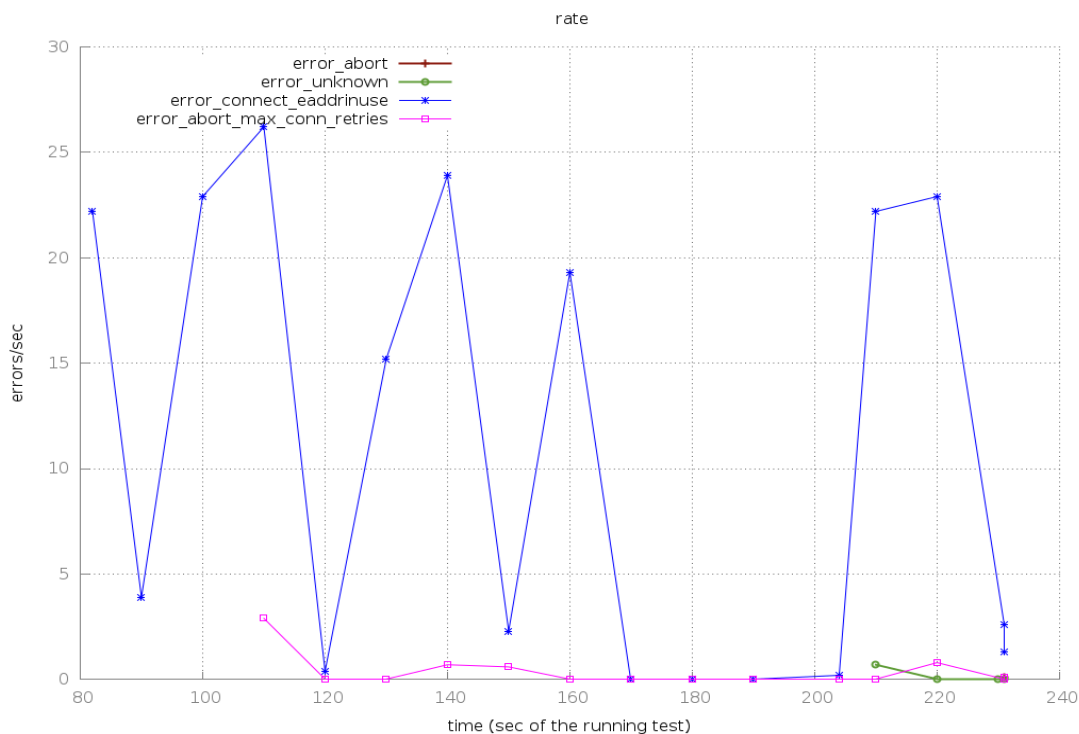


Рисунок 6 - Количество ошибок сервера в секунду (Apache MPM Worker)

На данном рисунке можно заметить, как после 80-й секунды выполнения теста появляются ошибки в работе сервера, следовательно, сервер перестает справляться с нагрузкой.

По итогам выполненного теста веб-сервера Apache с MPM Worker можно сделать следующий вывод: сервер способен выдерживать до 3000-4000 одновременных подключений посетителей.

Таким образом, разработанный на Erlang веб-сервер может поддерживать одновременно подключенных клиентов почти в 15 раз больше, чем веб-сервер Apache с MPM Worker.

Выводы

В ходе данной работы было проведено исследование моделей параллельных вычислений веб-серверов. Для этого был проведен анализ существующих моделей, установлено и настроено приложение для нагрузочного тестирования Tsung и проведены соответствующие эксперименты с разработанным веб-сервером [4] и сервером Apache. Выяснилось, что разработанный сервер способен выдерживать примерно в 10 раз больше одновременных подключений клиентов, чем сервер Apache с MPM Worker, что говорит о высокой масштабируемости Erlang-сервера за

счет использования модели акторов в основе языка программирования Erlang.

Библиографический список

1. Apache: MPM – worker, prefork или event? – Режим доступа: <https://rtfm.co.ua/apache-mpm-worker-prefork-ili-event/>
2. Математические модели параллельных вычислений – Режим доступа: <http://online.infosecurityrussia.ru/2012/articles/cloud/matematicheskie-modeli-parallelnyh-vychisleniy>
3. Кочетов П.С., Штанюк А.А. Использование языка программирования Erlang при разработке web-серверов // Постулат. 2016. № 12 (14). С. 23.
4. Кочетов П.С., Штанюк А.А. Исследование многопоточности веб-сервера, разработанного на языке программирования Erlang // Постулат. 2017. № 5-1 (19). С. 132.
5. Tsung Documentation. – Режим доступа: http://tsung.erlang-projects.org/user_manual.pdf