

О подходах к построению надежной цифровой образовательной платформы

М.С. Дьяченко¹, М.А. Кузьменко², А.Г. Кушнirenко³, Г.О. Райко⁴, И.Г. Райко⁵

¹ФГУ ФНЦ НИИСИ РАН, Москва, Россия, mdyachenko@niisi.ru;

²ФГУ ФНЦ НИИСИ РАН, Москва, Россия, gmk@niisi.ru;

³ФГУ ФНЦ НИИСИ РАН, Москва, Россия, agk_@mail.ru;

⁴ФГУ ФНЦ НИИСИ РАН, Москва, Россия, raiko@niisi.msk.ru;

⁵ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ilya.rayko@niisi.ru

Аннотация. В нашей стране активно идет процесс цифровой трансформации образования, который выдвигает требования создания цифровых образовательных платформ, обеспечивающих надежный доступ к разнообразным цифровым образовательным ресурсам. Такие платформы по сути своей являются распределенными вычислительными системами. Важность и критичность задач, выполняемых цифровыми образовательными ресурсами, определяет высокий уровень технических требований к платформам, на которых они реализованы. В статье рассмотрены критерии, обеспечивающие надежность платформ, проанализированы существующие популярные решения в области обеспечения надежности цифровых образовательных платформ и проведен анализ соответствия этих решений указанным критериям. Показано, что существующие цифровые образовательные платформы, в основном далеки от соответствия поставленным требованиям. Авторами предложен вариант цифровой образовательной платформы, удовлетворяющей сформулированным критериям.

Ключевые слова: цифровая образовательная платформа, вычислительные системы высокой надежности, территориально-распределенные вычислительные системы, цифровая образовательная платформа Мирера

1. Введение

Россия является одной из лидирующих стран в области цифровизации финансов, экономики, образования, госуслуг, а также активно проводит цифровую трансформацию образования во всех его составных частях, от цифрового хозяйства университетов до цифровизации образовательного процесса на уровне территориальных и федеральных органов власти. Тот факт, что Россия является самым большим по территории государством в мире, накладывает определенные требования к подобным цифровым платформам, в том числе является необходимым обеспечение высокоскоростного и надежного доступа к ресурсам вне зависимости от места нахождения обучаемого. В век тотальной цифровизации потребности внедрения цифровых образовательных платформ растут, однако основы педагогических образовательных систем были заложены еще в прошлом веке [1].

Для соответствия растущим потребностям в области цифровизации образования были разработаны десятки систем, многие из которых не отвечают современным требованиям. Особенно выпукло это несоответствие проявилось во время пандемии Covid-19, которая привела к крупнейшему за всю историю сбою в функционировании систем образования [2].

Во многом напряженность ситуации с надежностью цифровых образовательных платформ связана с неправильной оценкой требований к отказоустойчивости и высокой готовности таких систем и смешением этих требований с требованиями по распределенности нагрузки. Особенно ярко эта проблема видна среди систем проведения олимпиад и соревнований по программированию разных уровней, которые долгое время считались передовыми среди различных образовательных платформ.

2. Основные принципы

Цифровая образовательная платформа является распределенной системой программного обеспечения с отслеживаемым состоянием. Такие системы должны удовлетворять требованиям *готовности* и *согласованности*.

Вычислительная система обладает *высокой готовностью*, если она может продолжать работать при отказе некоторых компонент. Системы разделяют по количеству отказов, которые они способны выдержать. Ввиду своей специфики образовательные платформы обязаны сохранять высокую готовность всегда, даже во время проведения обслуживающих работ. Поэтому такие системы должны проектироваться с возможностью работы при двух и более отказах. В таких

системах должны использоваться отказоустойчивые кластеры класса не хуже HA-2 (HA это сокращение от High-Availability cluster - Отказоустойчивый кластер).

Основой обеспечения высокой готовности любой системы является условие, что *среднее время восстановления системы* намного меньше *среднего времени между отказами* системы: при нарушении этого условия текущее количество отказов в системе будет расти.

Говорят, что распределенная система обладает *согласованным состоянием*, если в каждый момент данные о состоянии некоторого хранящегося в системе логического объекта, полученные от разных компонент системы, совпадают. В некоторых случаях это требование ослабляют, допуская временное расхождение в данных, полученных от разных компонент. К сожалению, такое ослабление не является допустимым для цифровых образовательных платформ.

Связи между готовностью и согласованностью системы описываются эвристическим

утверждением, которое специалисты в теории распределенных систем несколько вольно называют CAP-теоремой [3]. Сочетание латинских букв CAP произведено от трех английских терминов, относящихся к распределенным системам: согласованность данных (Consistency), доступность данных (Availability) и устойчивость к разделению сети (Partitioning). Согласно CAP-теореме, любая распределенная система в случае ее разбиения на части в результате какого-либо отказа может либо продолжать работать (быть доступной пользователям), либо поддерживать согласованность данных.

Для достижения надежности системы должны существовать стратегии по восстановлению после катастроф. Такие стратегии связывают со следующими двумя величинами: временем на восстановление работоспособности (RTO) и интервалом времени, данные за который могут быть утеряны (RPO) (Рис. 1). У надежной цифровой платформы эти показатели должны быть близки к нулю.

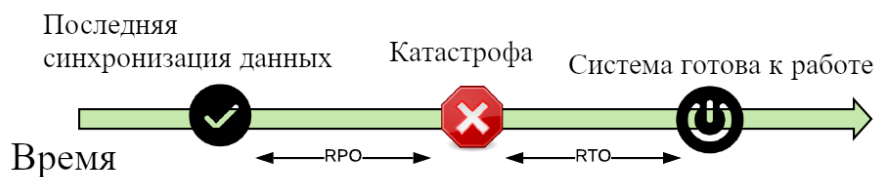


Рис. 1. RTO и RPO

Все эти требования выполняются для активно-активных территориально-распределенных систем. Такие системы являются набором удаленных комплексов (кластеров), каждый из которых является распределенной вычислительной системой. В отличие от активно-пассивных систем, в которых все кластеры, кроме одного, во время нормальной работы находятся в резерве, в активно-активной схеме все кластеры обслуживают запросы клиентов. Из-за этого удастся достичь околонулевого значения RTO: клиент быстро находит другой активный кластер и продолжает нормальную работу. Для предотвращения разбиения системы в случае катастроф (и потери готовности и согласованности, согласно CAP-теореме) *такие системы должны включать в себя хотя бы три кластера*. В таком случае низкий RPO достигается посредством строгой согласованности данных между кластерами в системе (что требует наличия высокоскоростного канала связи). Для поддержания условия HA-2 *каждый кластер должен состоять из хотя бы пяти вычислительных узлов*.

3. Имеющиеся решения

Ниже представлен обзор имеющихся решений.

3.1 Система *Programming Contest Control, PC²*

Система PC² [4] — разработка Университета штата Калифорния, США (California State University, CSUS) в Сакраменто. Система использовалась при проведении Международной студенческой олимпиады по программированию (International Collegiate Programming Contest), проводимой под эгидой Ассоциации вычислительной техники (Association for Computing Machinery, ACM inc.) и ряда других мероприятий. Система имеет модульную структуру. За хранение всех конфигурационных данных соревнований, прием решений от участников соревнований и хранение таких решений и ведение коммуникационных соединений между всеми остальными модулями системы отвечает модуль Server.

Система PC² допускает собственное географическое распределение, в каждом месте прове-

дения соревнований должен быть запущен экземпляр модуля типа Server. Однако, такая конфигурация решает лишь задачу минимизации задержек сети при обмене данными между географически распределенными точками. В системе должен существовать один экземпляр модуля типа Server, который должен быть объявлен основным, остальные модули типа Server считаются вспомогательными. Другие модули системы также допускают функционирование на выделенных аппаратных ресурсах, но лишь в целях распределения нагрузки на аппаратуру, минимизации нагрузки на единый сервер. Таким образом, средства распределения вычислений, предоставляемые системой PC², решают задачу проведения соревнований в реальном масштабе времени.

В целом, система не решает задач обеспечения отказоустойчивости или высокой готовности.

3.2 Система Contest Management System (CMS)

Система Contest Management System [5], CMS создана группой независимых разработчиков. Данная система использовалась при проведении Международной олимпиады по информатике (International Olympiad in Informatics, IOI) и ряда других мероприятий.

Система имеет модульную структуру, сервисы системы могут быть запущены на разных серверах, допускается репликация сервисов на различных серверах. Система использует СУБД PostgreSQL для хранения различных состояний соревнования, использование иных СУБД невозможно, поскольку CMS существенно опирается на специальный тип PostgreSQL «большой двоичный объект» (Binary Large Object, BLOB). Функции обеспечения отказоустойчивости, обеспечиваемые CMS также существенно опираются на соответствующие возможности СУБД PostgreSQL. К преимуществам такого подхода следует отнести то, что до тех пор, пока база данных находится в корректном состоянии, сервисы CMS могут функционировать независимо, в частности, в любой момент времени любой сервис может быть остановлен и перезапущен на другом сервере. При этом отсутствует необходимость в явном переносе информации между серверами. Сама СУБД PostgreSQL должна быть сконфигурирована так, чтобы предоставлять функции резервирования и отказоустойчивости. Следует отметить, что для PostgreSQL существуют разнообразные широкораспространенные решения для обеспечения резервирования и отказоустойчивости.

Все основные сервисы системы CMS спроектированы так, что любой из сервисов может

быть остановлен и перезапущен без потери целостности данных и без блокирования других сервисов системы. Система CMS обладает всеми требуемыми средствами обеспечения отказоустойчивости. Потенциальные проблемы, возникающие из-за существования в системе единой точки отказа, СУБД PostgreSQL, могут быть эффективно решены на уровне самой СУБД.

3.3 Система ejudge

Система ejudge [6] разрабатывается на факультете вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова. Система используется при проведении различных олимпиад по программированию в России, в частности, при проведении регионального этапа Всероссийской олимпиады школьников по информатике и программированию в некоторых регионах.

В качестве системы управления базой данных ejudge использует СУБД MySQL или MariaDB. Интерфейсные компоненты функционируют в рамках HTTP-сервера Apache и запускаются при поступлении запроса HTTP протокола, инициируемого клиентами.

Несмотря на то, что общесистемное программное обеспечение промежуточного уровня, включая СУБД и HTTP-сервер имеет развитые средства обеспечения отказоустойчивости, архитектура системы ejudge не позволяет полноценно задействовать данные функции. Управляющие компоненты работают в одном экземпляре, для обмена данными между некоторыми управляющими компонентами используются средства локальных коммуникаций — сокеты типа UNIX. Следует отметить, что второй тип взаимодействия между управляющими компонентами — структура общих (разделяемых) каталогов, хотя и допускает использование распределенных файловых систем, функционирующих по протоколам NFS или SMB/CIFS, настройка функций отказоустойчивости для подобных файловых систем требует штата системных администраторов очень высокой квалификации.

В целом, следует констатировать, что система ejudge не обладает функциями отказоустойчивости или высокой готовности, несмотря на наличие таких средств у отдельных частей системы.

3.4 ЦОП Мирера

Цифровая образовательная платформа Мирера [7], разрабатываемая специалистами ФГУ ФНЦ НИИСИ РАН, состоит из нескольких компонент:

- Система аутентификации пользователей;
- Система проверки решений учеников;
- API сервер для выдачи и приема решений задач;

– HTTP-сервер для создания, редактирования индивидуальных заданий, отслеживания и хранения результатов выполнения заданий.

Платформа является автономной и легко масштабируемой, как вертикально, так и горизонтально. Это обусловлено выбором технологий и программных продуктов, поддерживающих масштабируемость и отказоустойчивость. Отслеживаемое состояние имеется только у двух компонент: базы данных и очереди проверки (Рис. 2).

В качестве базы данных используется нереляционная документоориентированная СУБД MongoDB [8]. Все файлы, необходимые для работы системы, хранятся в модуле Grid File System – модулем MongoDB для работы с файлами. Данный модуль позволяет эффективно работать с файлами, а также позволяет использовать стандартные механизмы базы данных для обеспечения отказоустойчивости и масштабирования.

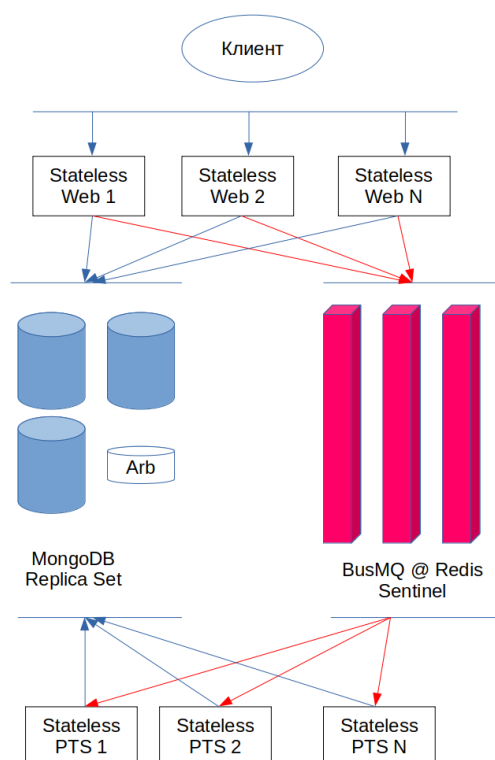


Рис. 2 Схема ЦОП Мирера. PTS — система проверки решений, в Web-сервисах находятся все системы, взаимодействующие с клиентами.

СУБД MongoDB является распределенной системой управления базой данных, поддерживающей механизм репликации данных — критического механизма для обеспечения функций отказоустойчивости. В терминах MongoDB реплика представляет собой группу нескольких экземпляров главного серверного процесса СУБД (mongod), каждый из которых хранит копию данных. При этом один из экземпляров объявлен главным, он ответственен за подтверждение операций записи данных и ведение журнала изменений набора данных. Остальные экземпляры являются вспомогательными, их задача — поддерживать собственные реплики данных в актуаль-

ном состоянии. В случае сбоя главного экземпляра группы вспомогательные экземпляры проводят выборы нового главного экземпляра. Дополнительно в группе может быть назначен арбитр — специальный экземпляр главного серверного процесса СУБД, на который не возложена задача хранения данных. Задача арбитра — обеспечение кворума для голосования при выборе нового главного экземпляра.

Помимо механизма репликации данных в СУБД MongoDB используется метод горизонтального масштабирования посредством техники сегментирования данных (sharding). В технике сегментирования хранимые данные разбиваются на диапазоны (сегменты) с помощью

ключа сегментирования. Каждый сегмент обслуживается своим экземпляром сервера СУБД. Техника горизонтального масштабирования обеспечивает дополнительный уровень доступности данных: при отказе одного из серверов СУБД теряется лишь порция данных, соответствующая одному сегменту.

Модуль Grid File System (GridFS) реализует файловую систему на базе СУБД MongoDB. В парадигме GridFS файл хранится не единой сущностью, но представляет набор частей (chunks), хранимых в виде отдельных документов (в терминологии MongoDB), т. е. иерархически организованных структур данных. При этом каждый файл представляется двумя коллекциями документов: одна коллекция хранит собственно части файла, другая — метаданные.

Очередь проверки построена на базе библиотеки BusMQ (модуля Node.js), в свою очередь основанной на сетевом журналируемом хранилище данных Redis [9]. Очередь BusMQ обеспечивает гарантированную доставку сообщений и поддерживает кластеризованный вариант сервера Redis - Redis Sentinel. Помимо этого, Redis Sentinel в случае отказа любой из систем сохраняет информацию об очереди, что гарантирует отсутствие потерь сообщений о тестировании.

Redis Sentinel предоставляет функции отказоустойчивости и высокой готовности для хранилища данных Redis. Следует отметить, что хранилище данных Redis само по себе содержит базовые элементы для построения отказоустойчивых решений. В частности, помимо механизмов поддержки транзакционных и пакетных операций над хранилищем, ведения снимков

данных и журналирования для обеспечения постоянного хранения данных, Redis поддерживает репликацию типа master-slave, когда в системе существует выделенный основной сервер (master), обрабатывающий текущие запросы, и один или несколько вспомогательных серверов (slave), которые содержат копию текущих данных. Технология Redis Sentinel имеет распределенную архитектуру без единой точки отказа. В системе может существовать несколько серверов Redis Sentinel, взаимодействующих между собой для обеспечения, как функции постоянной готовности, так и механизмов голосования при принятии решения об отказе основного сервера Redis, когда несколько экземпляров Redis Sentinel должны прийти к выводу об отказе.

4. Выводы

Проведенные исследования показывают, что существующие решения задач цифровизации образования при построении цифровых образовательных платформ не удовлетворяют требованиям надежности. Возникающие из-за этого проблемы могут иметь далеко идущие последствия вплоть до масштабных срывов образовательного процесса на уровне субъекта РФ или даже на федеральном уровне. Сформулированные авторами критерии могут стать основными критериями как при разработке геораспределенной системы Мирера, так и при построении других цифровых образовательных платформ.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН по теме № FNEF-2022-0010.

On Approaches for Building Reliable Digital Educational Platforms

M.S. Diachenko, M.A. Kuzmenko, A.G. Kushnirenko, G.O. Raiko, I.G. Raiko

Abstract. In our country, the process of digital transformation of education is actively underway, which puts forward the requirements for the creation of digital educational platforms that provide reliable access to a variety of digital educational resources. Such platforms are essentially distributed computing systems. The importance and criticality of the tasks performed by digital educational resources determine the high level of technical requirements for the platforms on which they are implemented. The article considers the criteria that ensure the reliability of platforms, analyzes the existing popular solutions in the field of ensuring the reliability of digital educational platforms, and analyzes the compliance of these solutions with the specified criteria. It is shown that the existing digital educational platforms are generally far from meeting the set requirements. The authors proposed a variant of a digital educational platform that meets the formulated criteria.

Keywords: digital educational platform, high-available cluster, geographically distributed cluster, digital educational platform Mirera

Литература

1. А.Г.Леонов, Ю.А.Первин. Качественные оценки эффективности методики обучения элементам информатики в пропедевтическом курсе. «Ярославский педагогический вестник», (2015), № 5.
2. Концептуальная записка: Образование в эпоху COVID-19 и в последующий период, Организация Объединенных Наций, август 2020 г. [Электронный ресурс] https://www.un.org/sites/un2.un.org/files/policy_brief_-_education_during_covid-19_and_beyond_russian.pdf (дата обращения: 29.08.2022).
3. Seth Gilbert, Nancy Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT News, Volume 33, Issue 2, June 2002
4. Steve Baber, David Hoelzeman, Becky Cunningham, Rick Massengale, Programming contest hosting: conference tutorial, Journal of Computing Sciences in Colleges, Volume 26, Issue 5, May 2011
5. Stefano Maggiolo, Giovanni Mascellani, Introducing CMS: A Contest Management System, Olympiads in Informatics, Vol.6, 2012
6. Ejudge – система для проведения различных мероприятий, в которых необходима автоматическая проверка программ. [Электронный ресурс] https://ejudge.ru/wiki/index.php/Система_ejudge (дата обращения: 29.08.2022).
7. Н.О.Бесшапошников, А.Г.Кушниренко, А.Г.Леонов, К.А.Прокин, Построение отказоустойчивых систем для проведения олимпиад по программированию, Труды НИИСИ РАН, Том 8, №6, 2018
8. MongoDB Operations Best Practices. [Электронный ресурс] https://s3-ap-southeast-1.amazonaws.com/tv-prod/documents%2Fnull-10gen-MongoDB_Operations_Best_Practices.pdf (дата обращения: 29.08.2022).
9. High availability with Redis Sentinel. [Электронный ресурс] <https://redis.io/docs/management/sentinel/> (дата обращения: 29.08.2022).