

В качестве характеристик эффективности обслуживания, в зависимости от условий задачи и целей исследования, могут применяться различные величины и функции: среднее количество заявок, которое может обслужить [2] в единицу времени; средний процент заявок, получающих отказ и покидающих СМО необслуженными; вероятность того, что проступившая заявка немедленно будет принята к обслуживанию; среднее время ожидания в очереди; закон распределения времени ожидания; среднее количество заявок, находящихся в очереди; закон распределения числа заявок в очереди; доход, приносимый СМО в единицу времени, и т. д. [2].

**Заключение.** Использование методов теории массового обслуживания показывает связь между объемом клиентов, количеством сотрудников торгового предприятия и их производительностью, а также эффективностью работы непосредственно расчетно-кассового узла.

Применяя метод теории массового обслуживания, решаются задачи по оптимизации планирования, качества и количества обслуживания, эффективной организации обслуживающих систем магазина. Все это приводит к минимизации затрат и издержек, увеличению конкурентоспособности и лучшему функционированию всей системы по обслуживанию покупателей.

#### Список цитируемых источников

1. Кошуняева Н. В. Теория массового обслуживания (практикум по решению задач) / Н. В. Кошуняева, Н. Н. Патронова ; САФУ имени М.В. Ломоносова. — Архангельск : САФУ, 2013. — 107 с.
2. Системы массового обслуживания и их применение в логистике [Электронный ресурс] — 2018 — Режим доступа: [http://www.kt-lospo.com/study/L\\_3\\_5.htm](http://www.kt-lospo.com/study/L_3_5.htm) — Дата доступа: 08.10.2018.

УДК 004.75

И. В. Яковюк

*Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», Минск*

### БЕЗОПАСНОСТЬ ХРАНЕНИЯ ДАННЫХ С СЕРН

**Введение.** Серн — свободная программная объектная сеть хранения, обеспечивающая как файловый, так и блочный интерфейс доступа.

Система не обладает узкими местами в безопасности и единими точками отказа, что положительно характеризует её с точки зрения безопасности хранения данных. Система легко масштабируема при достаточном наличии физических ресурсов. Кластеры узлов выполняют различные функции, обеспечивая хранение и репликацию данных, а также распределение нагрузки, чем достигается высокая надёжность и доступность.

**Основная часть.** Серн обеспечит сохранность данных при выходе из строя диска, узла или кластера узлов. После замены повреждённых дисков система восстановит данные с других дисков. При корректно настроенной конфигурации это не повлечёт за собой недоступность и потерю данных.

Система программно определяемая и работает поверх стандартных файловых систем и сетевых уровней. Для настройки кластера необходимы сервера / виртуальные машины с операционной системой Linux (Red Hat Enterprise Linux, CentOS, Fedora, Debian, Ubuntu), соединённые сетью между собой. Можно использовать внешнюю сеть, но для обеспечения более высокой скорости можно настроить по второй сетевой карте на каждом сервере и соединить сервера второй сетью [1].

Поверх операционной системы работают службы Серн, в документации их перечислено три: Mon — служба монитора; OSD — служба хранилища; MDS — сервер метаданных.

Если речь идёт об относительно небольшом кластере, то один сервер может совмещать несколько ролей.

Как правило, под OSD подразумевается один физический диск, однако это может быть также RAID-массив или iSCSI-устройство.

Кластер может иметь один или несколько пулов данных разного размера, пулы делятся на плейсмент-группы для хранения объектов, к которым обращаются клиенты. За каждой плейсмент-группой закреплён один главный диск, несколько дисков-реплик [2].

Физически диски могут находиться на разных узлах.

Фактор репликации — это уровень избыточности данных; количество копий данных, которое будет храниться на разных дисках. Пул — это логический абстрактный контейнер для организации хранения данных пользователя. Любые данные хранятся в пуле в виде объектов. Каждый пул имеет ряд настраиваемых параметров: фактор репликации, количество плейсмент-групп, минимальное количество живых реплик объекта, необходимое для работы, и другие параметры.

Плейсмент-группа — это связующее звено между физическим уровнем хранения (диски) и логической организацией данных (пулы). Число PG должно быть кратно степени «2».

Каждый объект на логическом уровне хранится в конкретной плейсмент-группе. На физическом же уровне он лежит в нужном количестве копий на разных OSD (физический диск, RAID-массив или iSCSI-устройство), которые в эту плейсмент-группу включены.

Монитор — это служба, выполняющая роль координатора, с которого начинается кластер. Как только появляется хотя бы один рабочий монитор, появляется Ceph-кластер. Монитор хранит информацию о здоровье и состоянии кластера, обмениваясь различными картами с другими мониторами. Клиенты обращаются к мониторам, чтобы узнать, на какие OSD писать/читать данные. При разворачивании нового хранилища первым делом создается монитор (или несколько). Кластер может прожить на одном мониторе, но рекомендуется делать три или пять мониторов во избежание падения всей системы по причине падения единственного монитора. Необходимо, чтобы количество мониторов было нечетным [1].

OSD (Object Storage Device) — это юнит хранилища, который хранит сами данные и обрабатывает запросы клиентов, обмениваясь данными с другими OSD. Обычно это диск. И обычно за каждый OSD отвечает отдельный OSD-демон, который может запускаться на любой машине, на которой установлен этот диск. Это второе, что нужно добавлять в кластер при первичной настройке.

В основе механизма децентрализации и распределения лежит CRUSH-алгоритм (Controlled Replicated Under Scalable Hashing), играющий важную роль в архитектуре системы. Этот алгоритм позволяет однозначно определить местоположение объекта на основе хеша имени объекта и определенной карты, которая формируется исходя из физической и логической структур кластера (датацентры, залы, ряды, стойки, узлы, диски). Карта не включает в себя информацию о местонахождении данных [1]. Путь к данным каждый клиент определяет сам, с помощью CRUSH-алгоритма и актуальной карты, которую он предварительно спрашивает у монитора. При добавлении диска или падении сервера карта обновляется.

Ceph предусматривает несколько способов увеличения производительности кластера методами кеширования: Primary-Affinity, вынос журналов на SSD, кеш-тиринг.

Клиент может смонтировать файловую систему CephFS, если у него Linux с версией ядра 2.6.34 или новее. Если версия ядра старше, то можно смонтировать ее через FUSE (Filesystem in User Space). Для того чтобы клиенты могли подключать Ceph как файловую систему, необходимо в кластере поднять хотя бы один сервер метаданных (MDS) [3].

Удаление данных непосредственно с диска — это довольно ресурсоемкая задача, в продвинутых системах удаление делается отложено или не делается вообще. Ceph — тоже продвинутая система, и в случае с RGW при удалении s3-объекта соответствующие RADOS-объекты не удаляются с диска сразу [3]. RGW помечает s3-объекты как удаленные, а отдельный gc-поток занимается удалением объектов непосредственно из RADOS-пулов и, соответственно, с дисков отложено.

**Заключение.** Так как Ceph лёгок в настройке и обеспечивает постоянный доступ к информации, часто используется хостинг-провайдерами, в таком случае клиент не беспокоится о проблемах с недоступностью и потерей данных. Для клиента важным является то, что данные не хранятся в одной точке, копии находятся на разных стойках (возможно даже в разных датацентрах).

#### Список цитируемых источников

1. Д'Атри, Э. Изучаем Ceph. / Э. Д'Атри. — 2-е изд. — М. : Модуль-Проекты, 2017.
2. Сингх, К. Книга рецептов Ceph / К. Сингх. — М. : Модуль-Проекты, 2016.
3. Фиск, Н. Полное руководство Ceph / Н. Фиск. — Packt Publishing Ltd, 2017.

УДК 004.657

**И. В. Яковюк**

*Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», Минск*

## МЕТОДЫ ОПТИМИЗАЦИИ ЗАПРОСОВ К БАЗЕ ДАННЫХ И ЗАЩИТА ОТ SQL-ИНЪЕКЦИЙ

**Введение.** Частой причиной медленной работы приложения, интернет-ресурса кроме ошибок, допущенных при настройке сервера, является наличие неоптимизированных, медленных запросов к базе данных. При долгом выполнении SQL-запроса конечный клиент долго ожидает ответа от сервера, к тому же в этот момент возрастает нагрузка на самом сервере. Для предотвращения указанных и многих других проблем необходимо провести оптимизацию.

**Основная часть.** SQL server management studio позволяет получать дополнительную информацию о выполнении запроса: план запроса получается функциями “Display Estimated Execution Plan” (оценочный план) и “Include Actual Execution Plan” (фактический план). Отличаются они тем, что оценочный план строится без выполнения запроса. Соответственно, информация о количестве обработанных строк в нем будет только оценочная. В фактическом плане будут как оценочные данные, так и фактические. Сильные расхождения этих величин говорят о неактуальности статистики.