ИСПОЛЬЗОВАНИЕ БЕССЕРВЕРНЫХ ТЕХНОЛОГИЙ ДЛЯ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Волосович С. В.
Кафедра информатики, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: svalasovich@gmail.com

Введение

Данные играют важную роль в машинном обучении, анализе данных, науке о данных, их представление и качество являются первостепенной заботой, поэтому обработка и управление данными играют важную роль.

Обработка данных, чаще всего, представляет собой высоконагруженный процесс, который требует высокие мощности, дорогостоящее оборудование, установку и настройку инструментов для реализации распределенной обработки данных (например Apache Spark, Apache Flink, Apache Hadoop, Apache Samza, Nemo), настройку и обслуживание серверов. При пакетной обработке данных в большинстве случаев работа выполняется по расписанию, либо другой сервис является триггером для запуска процесса обработки, это означает, что часть времени кластер простаивает без работы.

Для таких приложений можно использовать подход, при котором не требуется приобретать и настраивать оборудование, а использовать облачные бессерверные сервисы, где оплата производится только за использование ресурсов.

I. Бессерверные вычисления, приложения и особенности бессерверных сервисов

Бессерверные вычисления — естественная для облака архитектура, которая позволяет передать большую часть операционной ответственности на провайдера. Бессерверные вычисления позволяют создавать и запускать приложения и сервисы, не беспокоясь о серверах. Они устраняют необходимость заниматься вопросами управления инфраструктурой – такими, например, как выделение серверов или кластеров, необходимых ресурсов, а также установка исправлений и обслуживание операционной системы.

Бессерверное приложение — приложение которое не требует выделения, обслуживания и администрирования серверов для таких компонентов, как вычислительные ресурсы, базы данных, хранилища, компоненты обработки потоков и организации очередей сообщений и т. д. Заботиться об обеспечении отказоустойчивости и доступности приложения не требуется, этим занимается поставщик услуг.

Провайдеры облачных услуг предоставляют набор полностью управляемых бессерверных сервисов (сервисы для вычислений, хранилища, хранилища данных, оркестраторы, сервисы для аналитики и т.д.), которые можно использовать для создания и работы бессерверных приложений. Особенности бессерверных сервисов:

- Абстракция. Не требуется управление сервером на котором запускается сервис. Клиент ничего не знает про операционную систему, сетевые настройки и прочее;
- эластичность. Поставщик бессерверных сервисов предоставляет необходимое количество вычислительных ресурсов, в зависимости от того, какая нагрузка приходится на ваше приложение;
- эффективная стоимость. Если ваше приложение не выполняет работу, то вы не платите, т.к. оно не потребляет вычислительные ресурсы;
- ограниченый жизненный цикл. Если ваше приложение не используется, то сервис автоматически его останавливает.

II. Бессерверные сервисы для обработки больших данных

Приведем список бессерверных сервисов для обработки больших данных и их поставщиков:

- Dataflow — Google Cloud;
- AWS Glue – Amazon Web Services;
- Azure Data Factory — Microsoft Azure.

Для примера подробнее рассмотрим Dataflow. Google Cloud Dataflow – это управляемая бессерверная служба для выполнения пакетной или потоковой обработки данных. Для написания Dataflow задач используется Apache Beam SDK.

После написания pipeline с помощью Apache Beam, можно использовать Dataflow для его выполнения. Попав в сервис, pipeline становится Dataflow задачей. Dataflow сервис полностью управляет сервисами Google Cloud, такими как Compute Engine и Cloud Storage для выполнения задания Dataflow, автоматически выполняет все функции администрирования, включая создание кластера, масштабирования необходимых ресурсов, мониторинг результата и ведение журнала. Для выполнения задания, Dataflow разбивает задание на трансформации и строит граф выполнения (см. рис. 1), после чего согласно построенному графу выполняются все трансформации. По окончанию задания, Dataflow автоматически останавливает кластер и освобождает все ресурсы.

Построение графа занимает некоторое время, поэтому для часто используемых заданий рекомендуется создавать Dataflow Template, при создании которого трансформации с построенным графом записываются в файл и может быть сохранено в Google Storage. После этого Dataflow задание может быть запущенно без необходимости повторного построения графа.

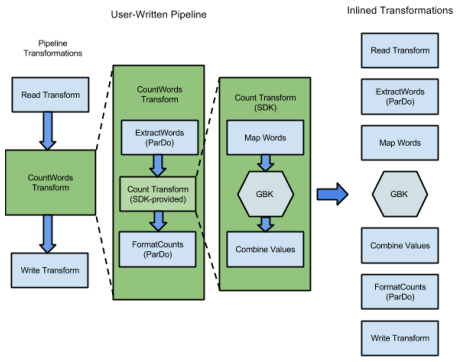

Рис. 1 – Потроение графа выполнения на примере подсчета слов

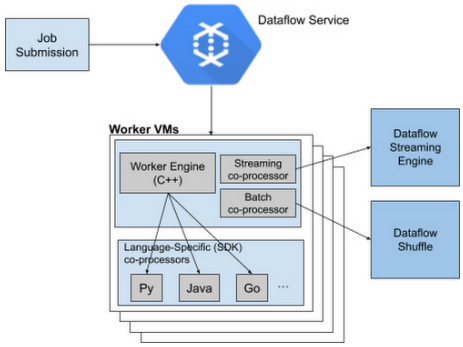

Рис. 2 – Архитектура worker'а

Вычислительным узлом в Dataflow является worker (см. рис. 2), который имеет виртуальные CPU (vCPU) и оперативную память в зависимости от заданного типа Compute Engine. Максимальное и стартовое количество worker'ов для выполнения задачи можно задавать при запуске.

III. Сравнение возможностей

Сравним бессерверные сервисы для обработки данных от разных поставщиков и представим в виде таблицы (см. таблицу 1). Для сравнения будем рассматривать следующие параметры:

1. Название сервиса;
2. возможность пакетной обработки;
3. возможность потоковой обработки;
4. учитываемые параметры для расчета стоимости выполнения задания;
5. наличие и тип SDK;
6. языки программирования для разработки задания по обработке данных;
7. визуализация выполняемых этапов задачи;
8. программное ядро.

Как видно из таблицы, базовые потребности в обработке данных удовлетворяет каждый сервис. Однако из-за разного программного ядра и разных политик ценообразования, в целях экономии и уменьшения времени исполнения, к каждому заданию требуется индивидуальный подход, для выбора необходимого сервиса.

Список литературы

1. AWS Glue Documentation [Electronic resource] / Amazon. – Amazon Web Services. – Mode of access: https://docs.aws.amazon.com/glue/. – Date of access: 18.10.2020.
2. Dataflow Documentation [Electronic resource] / Google. – Google Cloud. – Mode of access: https://cloud.google.com/dataflow/docs/. – Date of access: 18.10.2020.
3. Azure Data Factory Documentation [Electronic resource] / Microsoft. – Microsoft Azure. – Mode of access: https://docs.microsoft.com/en-us/azure/data-factory/. – Date of access: 18.10.2020.

Таблица 1 – Сравнительная характеристика

№ Параметра	Amazon Web Services	Google Cloud	Microsoft Azure
1	AWS Glue	Dataflow	Azure Data Factory
2	Да	Да	Да
3	Да	Да	Да
4	Время затраченное на выполнение задания, количество DPU (вычислительный узел)	Время затраченное на выполнение задания, количество worker'ов, тип worker'а, объем перемешивания (shuffle)	Оплата за вызов задания, время выполнения задания, используемое ядро для трансформаций и тип вычислительного узла
5	Apache Beam	Собственная реализация на базе Apache Spark	Собственная реализация
6	Java, Python, Go	Scala, Python	Code-free (не требует знания языков программирования), .NET, Python
7	Да	Да	Да
8	Собственная реализация на C++	Apache Spark	Зависит от выбранной среды исполнения трансформаций