rus.understanding_data_storage

Alexandr Kirilov (https://github.com/alexandrkirilov)

Понимание data storage.

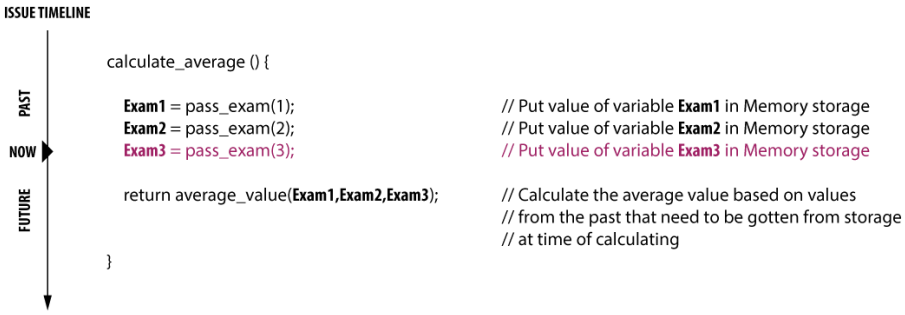
Для того, что бы проектировать и разрабатывать хранилища данных (data storage) нужно разобраться в причине, по которой мы должны хранить те или иные данные. Именно причина хранения данных будет определять то, на сколько эти данные важны, как они могут быть использованы, на сколько быстро они должны быть извлечены из хранилища и т.д.

При описании процессов в данной статье часто используется понятия "прошлое", "настоящее" и "будущее" - в этом нет никакого скрытого смыла. Это просто наиболее понятное описание последовательности действий, точно так же как 1-2-3. С точки зрения data storage это означает:

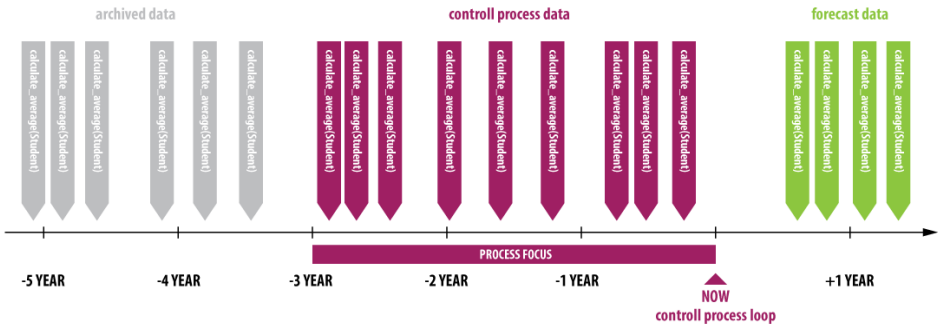
- 1 - прошлое: данные сохранены
- 2 - настоящее: над данными производятся какие-то действия или они находятся в сохраненном состоянии для будущего использования
- 3 - будущее: с данными будут производиться какие-то действия

Во всех случаях причина по которой нужно хранить данные одна - что-то не решено или не завершено В РЕАЛЬНОМ МИРЕ (этот пункт специально отмечен) и потребует передачи информации о прошлом в будущем, то о чем вы не можете забыть по каким-то причинам, например:

- Получение оценки студентом или учащимся на экзамене: неразрешенной ситуацией является вычисление среднего балла для диплома или сертификата в будущем который будет рассчитываться с учетом текущего экзамена и прошлых экзаменов. На основании этой неразрешенной ситуации - вычисление среднего балла, вы обязаны хранить все оценки студента по всем экзаменам за период который определен учебной программой, где время от начала обучения до получения сертификата или диплома время обязательного хранения данных.



- Система контроля качества обучения: неразрешенной ситуацией является постоянный процесс контроля качества на основании данных за последние, для примера, 3 года (если процесс включает в себя анализ промежуточных экзаменов, то вы должны хранить данные о промежуточных экзаменах на основании которых был вычислен промежуточный бал), где время обязательного хранения вычисляется от текущего времени процесса до границы в прошлом определенной фокусом процесса - в данном случае 3 года назад.



- Система учета проданных товаров: неразрешенные ситуации в данном случае две - уплата налогов и стратегическое планирование (эти два пункта взяты для примера, количество процессов может быть различным). Все эти процессы используют данные из прошлого для процесса вычисления или прогнозирования для какого-то действия в будущем.
- Система контроля доступа к объекту: неразрешенная ситуация - жизнь объекта к которому ограничен доступ черед постоянный процесс контроля и анализ возможных попыток проникновения в будущем на основании данных из прошлого при процессе анализа в настоящем.
- и т.д.

Неразрешенные ситуации или процессы в реальном мире могут быть лимитированными по времени (в IT мы называет это функциями и процедурами) или не лимитированными по времени (в IT мы называем это службами, процессами или демонами). Процессы могут включать в себя субпроцессы и в обратном порядке быть частью других процессов. Для понимания возможных структур процессов можно использовать понимание принципов описанных в статье "Понимание Blockchain".

При проектировании data storage крайне важно понимать взаимосвязи данных процессов для которых вы собираетесь организовывать хранилище, потому что на основании этих данных вы будете проектировать необходимый размер хранилища, а соответсвенно и затраты на строительство и обслуживание - размер данных в одну единицу времени умноженный на время жизни процесса даст вам минимальное значение размера хранилища которое вам необходимо. А структура самого процесса и его участники определят на сколько быстро и как должны быть извлечены данные из хранилища и в каком формате.

```
Примечание автора.
Основная проблема Big Data заключается именно в
большом количестве неразрешенных задач или процессов в реальном
мире и как результат этого появляется необходимость хранения
большого объема данных в Big Data Storage. Понятие "Big Data" не
возникло само по себе, это результат. Может начать с понимания
процессов и оптимизации в реальном мире и только после этого
начинать решать проблему Big Data если таковая останется после
оптимизации процессов в реальном мире?
Может и не понадобится Big Data Storage?
```

Основываясь на опыте разработки data storage, всегда нужно держать максимальный размер хранилища с запасом. Для одно-юнитных хранилищ это 30%-40%, для распределенных хранилищ это значение может варьироваться от 3% до 25%. Значение запаса в размере напрямую зависит от возможностей обеспечить отказоустойчивость в случае лавинообразного роста количества объектов хранения. Чем больше вам нужно времени на увеличение объема хранилища относительно роста объема данных - тем больше вам нужен запас. В каждом случае вычисляется отдельно.

Процесс разработки архитектуры data storage, обычно, на 90% состоит из решения задач не имеющих к IT никакого отношения. В большинстве случаев это должна быть работа как минимум двух специалистов - один IT специалист, а другой специалист из отрасли для которой вы разрабатываете data storage.