

Задание 2
Репликация. Эпидемический протокол.
Крайний срок - 12 ноября

Общая формулировка

Реализовать систему хранения данных с репликацией. Тип репликации: главный-подчинённые. Все записи идут через главную реплику. Репликация синхронная: главная реплика ожидает, чтобы новые значения записались на подчинённые.

Архитектура

Процессы, участвующие в системе:

- главная реплика
- подчинённые (их количество — параметр запускающего скрипта)
- клиент

Главную и подчинённые реплики запускает скрипт, клиент запускается пользователем отдельно. Каждая реплика запускается в своём окне

Функциональные требования

На каждой реплике хранятся три ключа — x, y, z. Их значения — строки.

Предполагается, что количество реплик постоянно, отказов нет. Каждая реплика знает, как связаться со всеми остальными репликами

Клиент принимает команды вида: <имя переменной>, <новое значение>. Отсылает их главной реплике. Далее обновление распространяется с помощью эпидемического протокола (gossip)

Эпидемический протокол

Реплика (главная либо подчинённая), которая получила новое значение переменной, должна передать его дальше. Для этого, она случайным образом выбирает k соседей (k может быть параметром или зашитым значением). Если кого-то из этих соседей нет нового значения данной переменной — передаёт ей обновление.

Для того, чтобы отличать новое/старое значение — необходимо присваивать значению номер версии (этим занимается главная реплика)

Каждая реплика, которая получила новое значение:

- сообщает об этом главной реплике
- распечатывает новое значение своей копии переменной

После того как все реплики сообщат главной о том, что значение переменной обновлено, главная реплика распечатывает соответствующее сообщение.