D. Эламские архивы

У месопотамской цивилизации в III тысячелетии до нашей эры было два важных центра — Шумер и Элам. Шумер, расположенный в долине между реками Тигр и Евфрат, славился своими богатыми урожаями, а Элам был богат металлами и другими природными ресурсами. Контроль над этими ресурсами и торговыми путями приводил к конфронтации между двумя цивилизациями.

Уже тогда записи играли важную роль в управлении, хозяйстве, религиозной жизни и законотворчестве. Они изначально издавались в столице, после чего перевозились в один из архивов. При получении данных о скором нападении противника важные документы перераспределялись из архивов в «горячих точках» в более безопасные места.

Вы захотели сымитировать это историческое событие в современных цифровых реалиях. Каждый из архивов можно представить как кеширующую базу данных, где по ключу названия документа находится его содержимое. Все данные изначально создаются в главной базе, после чего вам необходимо переместить их в одну или несколько кеширующих реплик - на один и тот же ключ в главную базу данных должно быть не более одного запроса. Обращаться к базам можно по HTTP-интерфейсу. Реплики должны быть сбалансированы, иначе они не смогут вместить всю информацию.

Не более одного раза вы воспроизведёте нападение на какой-то из архивов — на запросы кеширующая реплика начнёт всегда отвечать HTTP-кодом 502.

Вам необходимо реализовать алгоритм, который сможет распределять данные и определять их местонахождение с учётом возможного нападения.

Формат ввода

Изначально главная база данных заполнена всеми возможными ключами и не поддерживает операцию редактирования. Сервера-реплики находятся в работоспособном состоянии и пусты.

Участнику предоставляется конфигурационный файл с URL базы данных и серверов-реплик — путь до файла передается в качестве первого параметра командной строки в решение участника. Конфигурационный файл имеет следующий формат:

{

"db\_url": "http://db\_host:db\_port/some\_path/",

"cache\_urls": [

"http://cache1\_host:cache1\_port/some\_path/",

"http://cache2\_host:cache2\_port/some\_other\_path/",

...

]

}

Количество серверов-реплик может быть от 2-х до 10.

На вход программе через stdin по одному передаются ключи. В ответ на каждый ключ, полученный из stdin ожидается запись значения для этого ключа в stdout в виде строки, которая должна заканчиваться символом \n (сами значения не содержат \n).

Ключи представляют собой строковые литералы формата [\w-]+ и длиной не больше 20 символов. Количество уникальных ключей — не более 20000. Строковые значения, которые соответствуют ключам являются ASCII строками.

Дополнительные файлы, которые понадобятся вам для решения задачи можно найти [здесь](https://yastatic.net/s3/contest/cup/2023-public-files/backend_final_championship-40_8cec158d02ee7c692ad4f417c7b2841b.zip).

Формат вывода

| stdin | Ожидаемое поведение | stdout |
| --- | --- | --- |
| key-1 | запрос за значением для key-1 в главную базу данных | value-1 |
| key-2 | запрос за значением для key-2 в главную базу данных | value-2 |
| key-2 | запрос за значением для key-2 в реплику | value-2 |
| key-1 | запрос за значением для key-1 в реплику | value-1 |

Примечания

В зависимости от выбранного метода или языка запись в stdout в вашем решении может буферизоваться. Убедитесь, что каждая команда принудительно отправляется из буфера в stdout, иначе проверяющая система может не получать ваши ответы.

Для отладки решения участникам доступен вывод в stderr. Он будет выводиться в соответствующем разделе отчета о тестировании.

Проверяющая система может эмулировать отказ, в том числе приложения участника. После отказа приложение участника перезапускается автоматически с самого начала.

Решение должно выполняться быстрее, чем за 15 секунд, и общая память, используемая решением, базой данных и кеширующими серверами не должна превышать 512 Мб.

Задача предполагает получение баллов за каждый отдельный тест. Максимальное количество баллов — 200.