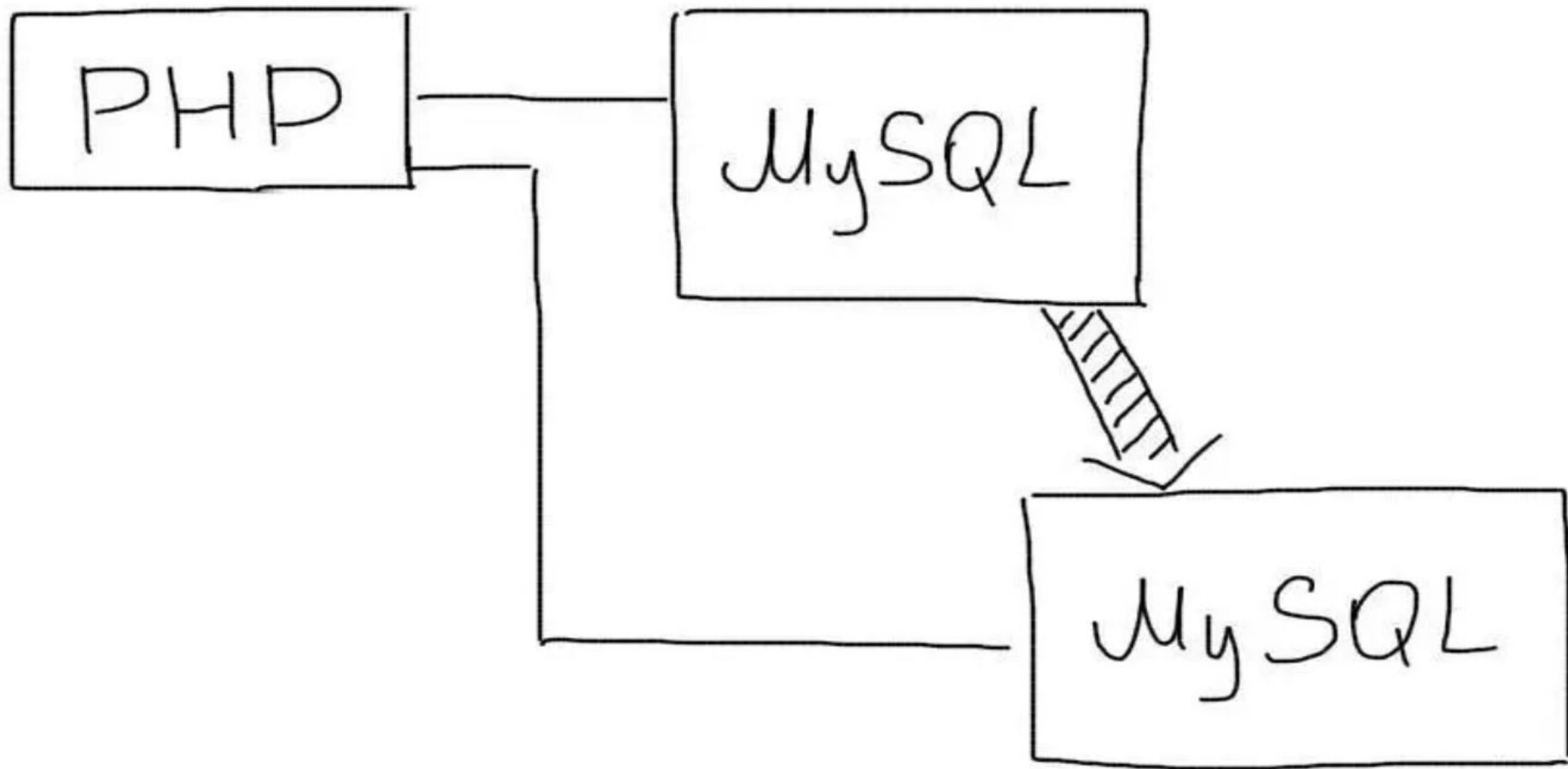


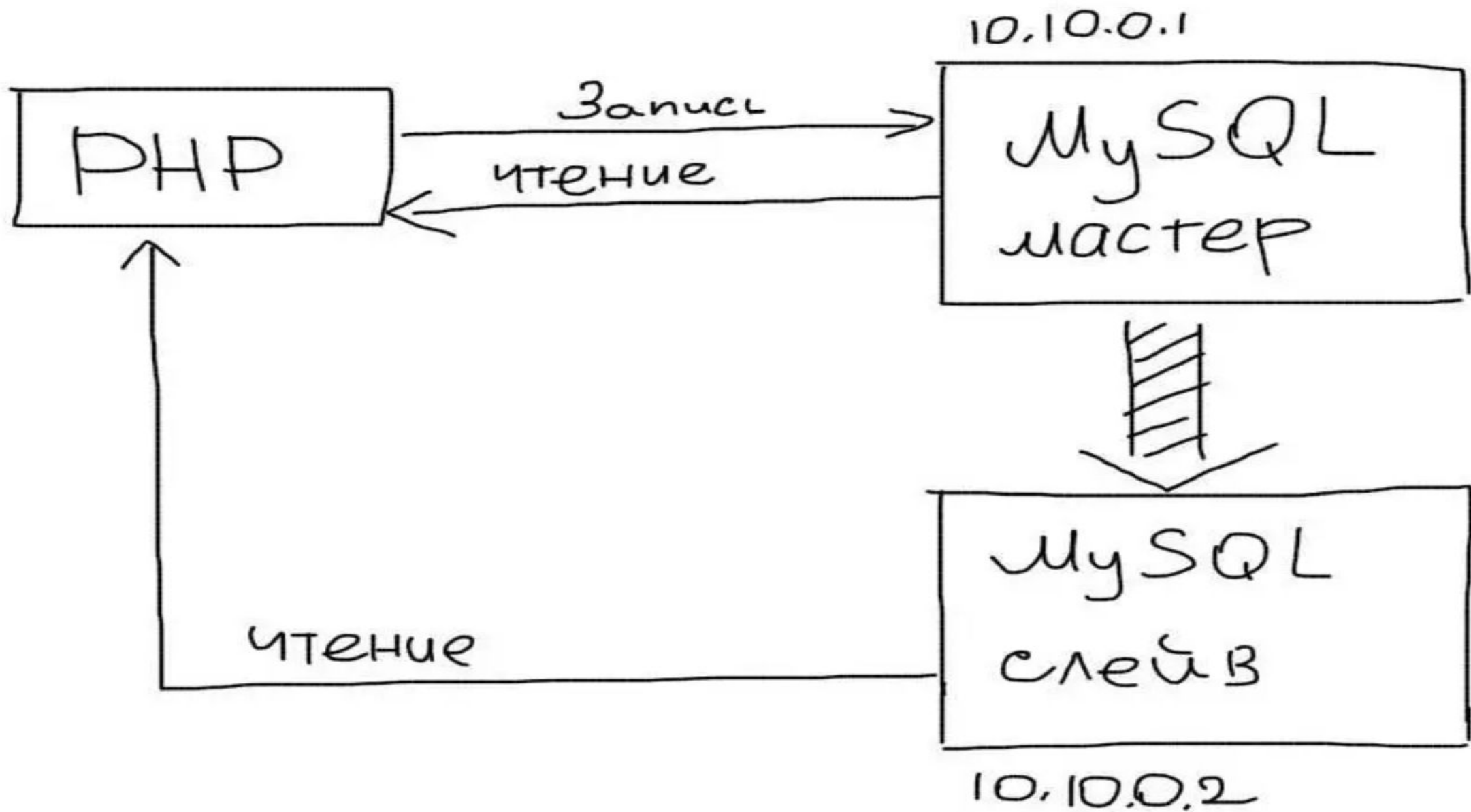
Репликация данных

- **Репликация** — одна из техник масштабирования баз данных. Состоит эта техника в том, что данные с одного сервера базы данных постоянно копируются (реплицируются) на один или несколько других (называемые репликами). Для приложения появляется возможность использовать не один сервер для обработки всех запросов, а несколько. Таким образом появляется возможность распределить нагрузку с одного сервера на несколько.



Master-Slave репликация

- В этом подходе выделяется один основной сервер базы данных, который называется Мастером. На нем происходят все изменения в данных (любые запросы MySQL INSERT/UPDATE/DELETE). Слейв сервер постоянно копирует все изменения с Мастера. С приложения на Слейв сервер отправляются запросы чтения данных (запросы SELECT). Таким образом Мастер сервер отвечает за изменения данных, а Слейв за чтение.



- В приложении нужно использовать два соединения — одно для Мастера, второе — для Слейва:

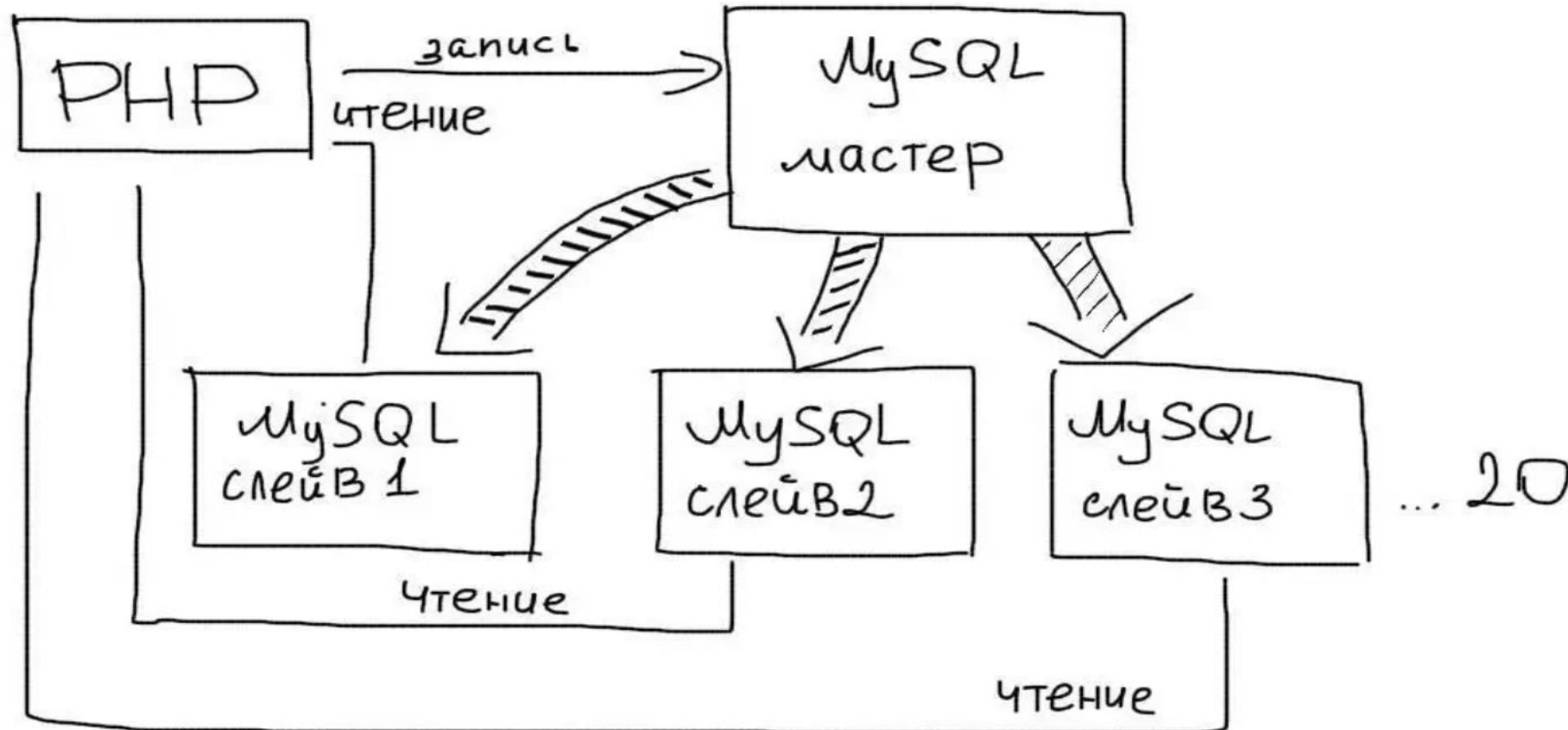
```
<?
$master = mysql_connect('10.10.0.1', 'root', 'pwd');
$slave = mysql_connect('10.10.0.2', 'root', 'pwd');

# ...
mysql_query('INSERT INTO users ...', $master);

# ...
$q = mysql_query('SELECT * FROM photos ...', $slave);
```

Несколько Слейвов

- Преимущество этого типа репликации в том, что Вы можете использовать более одного Слейва. Обычно следует использовать не более 20 Слейв серверов при работе с



Тогда из приложения Вы выбираете случайным образом один из Слейвов для обработки запросов:

```
<?
$master = mysql_connect('10.10.0.1', 'root', 'pwd');
$slaves = [
    '10.10.0.2',
    '10.10.0.3',
    '10.10.0.4',
];
$slave = mysql_connect($slaves[array_rand($slaves)], 'root', 'pwd');

# ...
mysql_query('INSERT INTO users ...', $master);

# ...
$q = mysql_query('SELECT * FROM photos ...', $slave);
```


Задержка репликации

- Асинхронность репликации означает, что данные на Слейве могут появиться с небольшой задержкой. Поэтому, в последовательных операциях необходимо использовать чтение с Мастера, чтобы получить актуальные данные:

```
<?
$master = mysql_connect('10.10.0.1', 'root', 'pwd');
$slave = mysql_connect('10.10.0.2', 'root', 'pwd');

# ...
mysql_query('UPDATE users SET age = 25 WHERE id = 7', $master);
$q = mysql_query('SELECT * FROM users WHERE id = 7', $master);

# ...
$q = mysql_query('SELECT * FROM photos ...', $slave);
```

Выход из строя

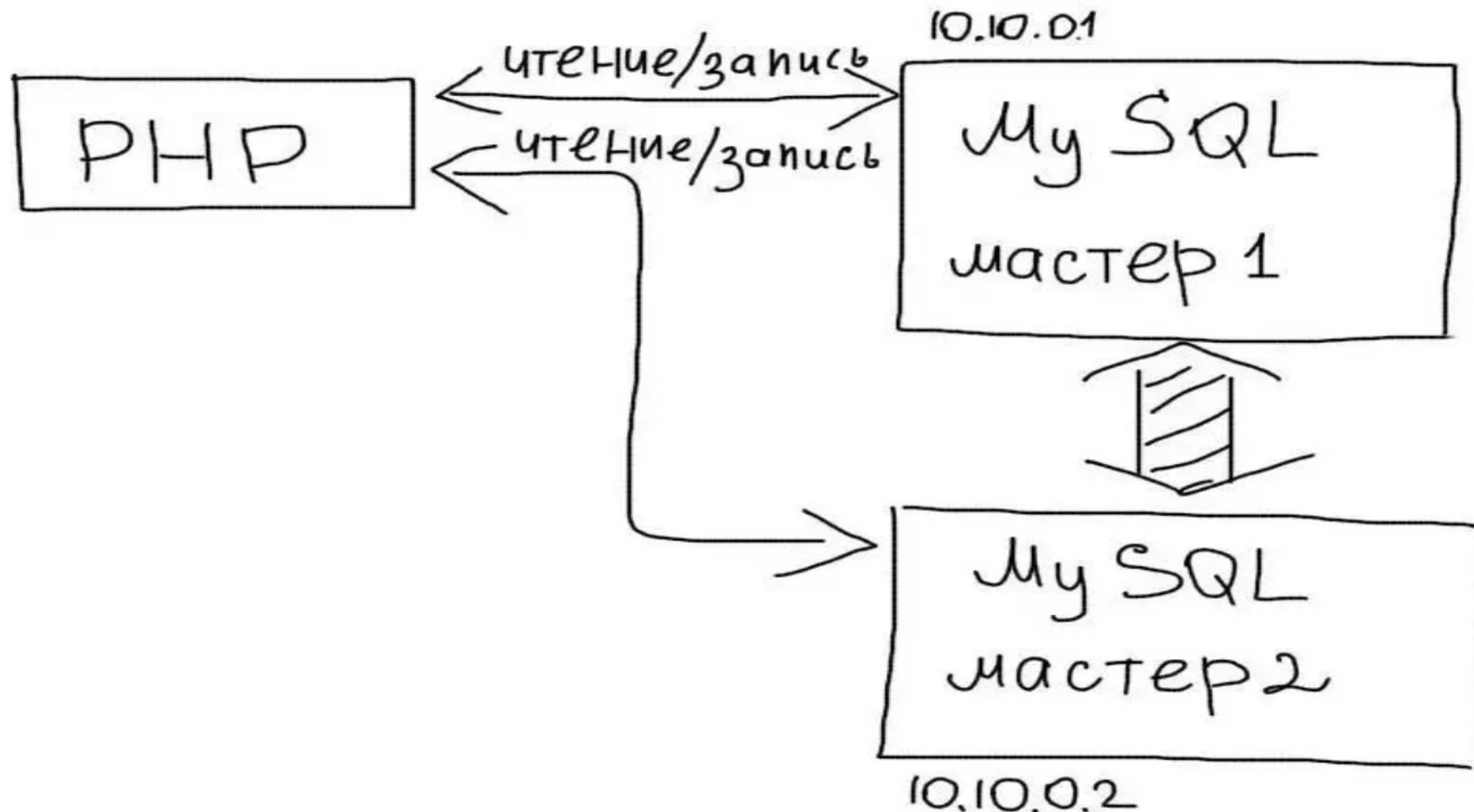
- При выходе из строя Слейва, достаточно просто переключить все приложение на работу с Мастером. После этого восстановить репликацию на Слейве и снова его запустить.
- Если выходит из строя Мастер, нужно переключить все операции (и чтения и записи) на Слейв. Таким образом он станет новым Мастером. После восстановления старого Мастера, настроить на нем реплику, и он станет новым Слейвом.

Резервирование

- Намного чаще репликацию Master-Slave используют не для масштабирования, а для резервирования. В этом случае, Мастер сервер обрабатывает все запросы от приложения. Слейв сервер работает в пассивном режиме. Но в случае выхода из строя Мастера, все операции переключаются на Слейв.

Master-Master репликация

- В этой схеме, любой из серверов может использоваться как для чтения так и для записи:

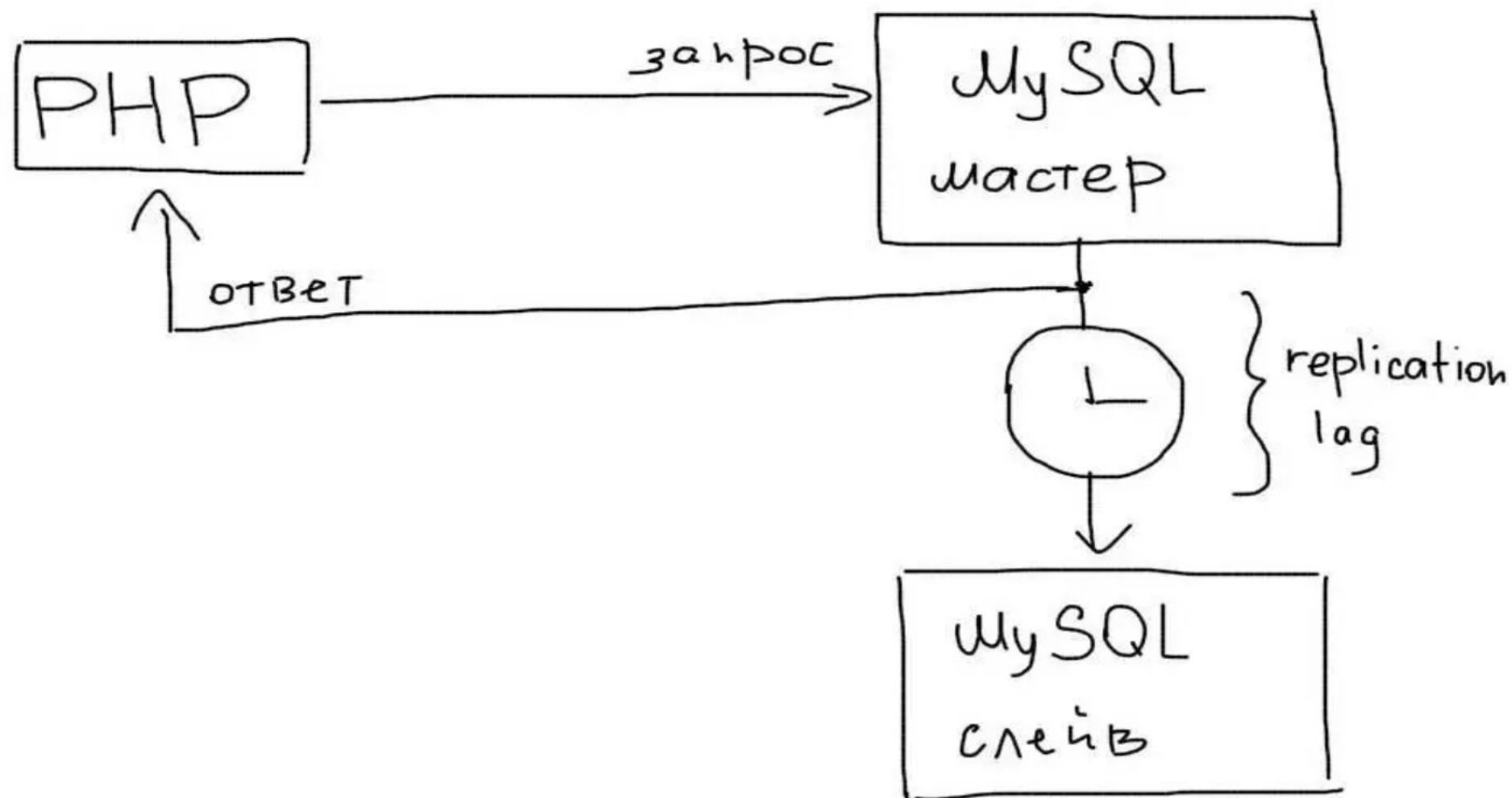


Выход из строя

- Вероятные поломки делают Master-Master репликацию непривлекательной. Выход из строя одного из серверов практически всегда приводит к потере каких-то данных. Последующее восстановление также сильно затрудняется необходимостью ручного анализа данных, которые успели либо не успели скопироваться.
- Используйте Master-Master репликацию только в крайнем случае.

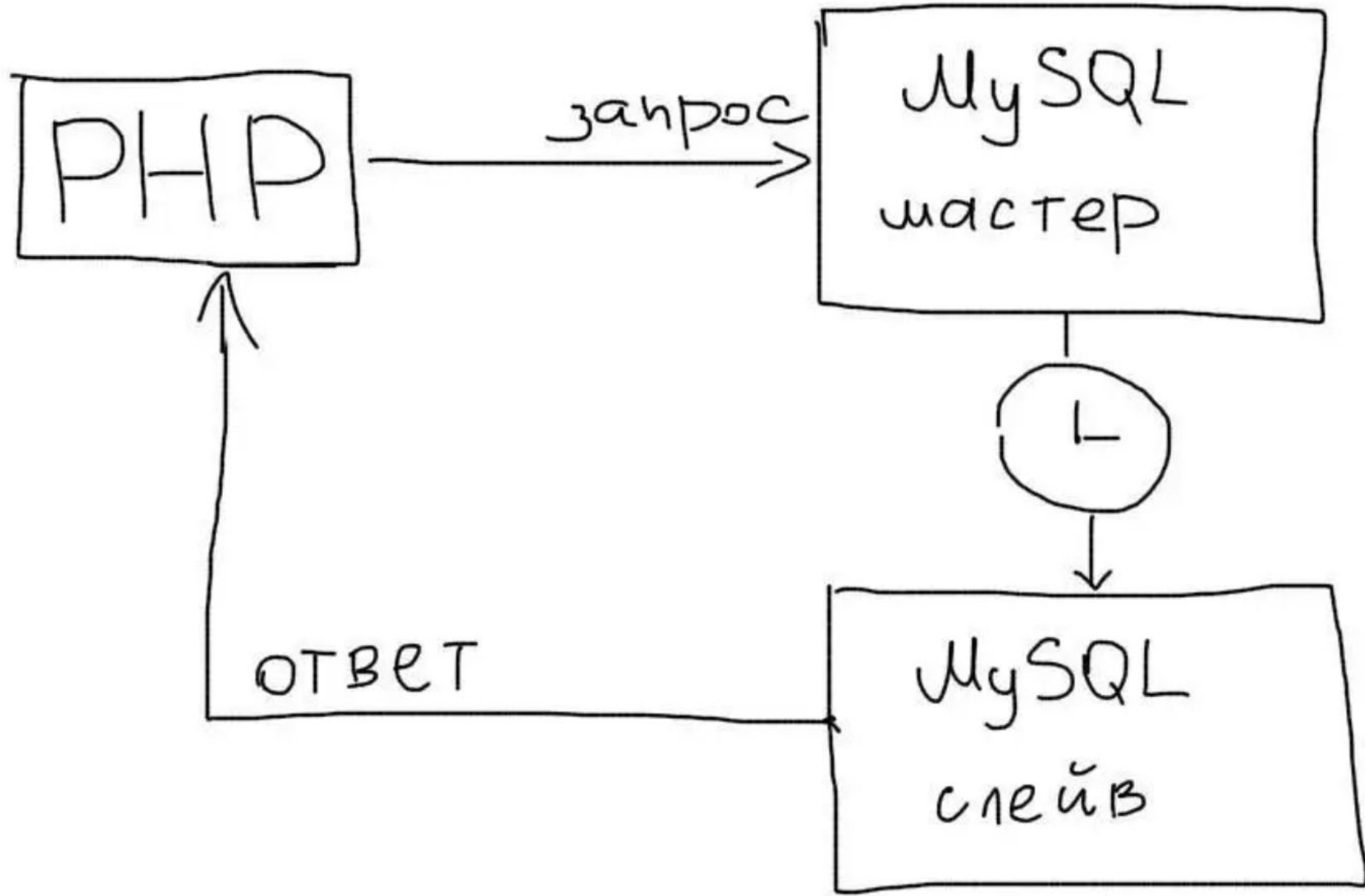
Асинхронность репликации

- В MySQL репликация работает в асинхронном режиме. Это значит, что приложение не знает, как быстро данные появятся на Слейве.



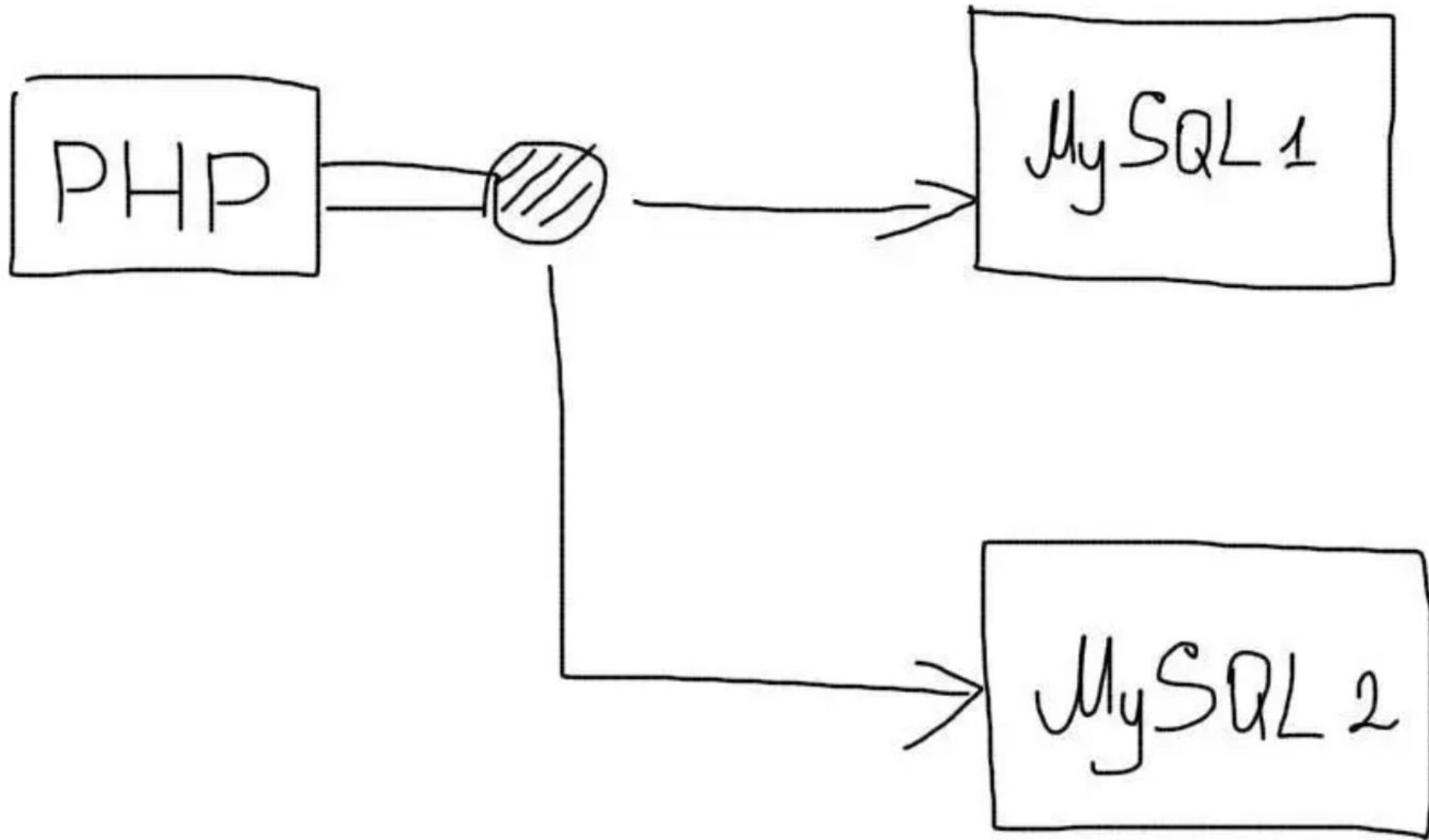
Синхронный режим

- [Синхронный режим](#) репликации позволит гарантировать копирование данных на Слейв.
- Это упростит работу в приложении, т.к. все операции чтения можно будет всегда отправлять на Слейв. Однако это может значительно уменьшить скорость работы MySQL. Синхронный режим не следует использовать в Web приложениях.



"Ручная" репликация

- Следует помнить, что репликация — это не технология, а методика. Встроенные механизмы репликации могут принести ненужные усложнения либо не иметь какой-то нужной функции. Некоторые технологии вообще не имеют встроенной репликации.
- В таких случаях, следует использовать самостоятельную реализацию репликации. В самом простом случае, приложение будет дублировать все запросы сразу на несколько серверов базы данных:



- При записи данных, все запросы будут отправляться на несколько серверов. Зато операции чтения можно будет отправлять на любой сервер. Нагрузка при этом будет распределяться по всем доступным серверам:

