Необходимо реализовать систему сервисов генераторов и подписчиков, которые общаются посредством общей шины данных и пишут результат в реляционную базу данных. Система состоит из следующих частей:

1. 2 сервиса генератора, один из которых генерирует по порядку числа Фибоначчи, а второй генерирует по порядку ПРОСТЫЕ числа. Сервисы пишут данные в шину в разные каналы. Данные в каналах идут точно в таком же порядке как были сгенерированы.
2. Шина данных представляет собой либо сервер очередей, либо базу данных (не реляционную) которая работает в оперативной памяти. Например, amqp, zeromq, memcache, redis
3. 2 сервиса подписчика, каждый из которых получает данные из шины со своего канала и складывает результаты в реляционную базу данных. Подписчики в каждой итерации должны работать с одним числом из шины и в этой же итерации делать запись вычислений в базу данных.
4. База данных состоит из одной таблицы с одной записью, состоящей из 3 полей.

Подробные условия для каждой из частей:

1. Первый сервис генератор представляет собой скрипт в котором генерируются и отправляются в шину данных числа Фибоначчи. При старте скрипта задается 2 параметра. 1 - кол-во чисел, которые нужно сгенерировать и отправить в шину, 2 - время паузы в микросекундах перед генерацией следующего числа.
2. Второй сервис генератор представляет собой скрипт в котором генерируются и отправляются в шину данных ПРОСТЫЕ числа. При старте скрипта задается 2 параметра. 1 - кол-во чисел, которые нужно сгенерировать и отправить в шину, 2 - время паузы в микросекундах перед генерацией следующего числа.
3. Шина данных имеет 2 канала, один для чисел Фибоначчи, второй для ПРОСТЫХ чисел. Способ организации данных в каналах отвечает принципам FIFO.
4. Реляционная база данных (MySql) имеет всего одну таблицу, структура описана ниже. В таблице в любой момент времени находится ОДНА запись.
5. Функционал сервисов подписчиков идентичный, но каждый инстанс (отдельно запущенный скрипт) подключается к своему каналу в шине данных. На каждой итерации сервис читает число из канала и прибавляет его значение к значению в поле `sum` в БД, при этом в соответствующем поле count увеличивается значение на единицу. Оба скрипта подписчика запускаются одновременно и параллельно работают с шиной и базой данных.

Структура БД:

```
CREATE TABLE `test` (
 `sum` ,
```

`count_fib` ,
`count_prime`
) ENGINE= DEFAULT CHARSET=utf8;
Типы полей и движок таблицы на выбор исполнителя.
`sum` - суммарное значение обработанных чисел Фибоначчи и ПРОСТЫХ
`count_fib` - кол-во обработанных чисел Фибоначчи
`count_prime` - кол-во обработанных ПРОСТЫХ чисел

Требования по реализации:

1. PHP 7+
2. Без использования фреймворков
3. Желательно сделать отдельный файл конфигов для скриптов
4. Можно использовать дополнительные модули из PECL (они должны быть указаны в описании по запуску реализации)
5. В качестве шины данных можно использовать любое решение, которое подходит на ваш взгляд. Описать как запустить и настроить это решение. Хотелось бы посмотреть в основном на реализацию с Redis
6. В качестве реляционной базы данных нужно использовать MySQL
7. На выполнение задания отводится 16 часов.

В результате выполнения задания в простейшем виде должны быть получены 4 скрипта (2 генератора, 2 подписчика), файл конфигурации и текстовый файл (.txt, .md) с описанием и заметками по запуску. Любые дополнительные файлы должны быть описаны. Так же должен быть текстовый файл (result.txt), который будет содержать всего одно число, которое является суммой 2000 первых чисел Фибоначчи и 5000 первых ПРОСТЫХ чисел.

Пример:
Мы должны посчитать сумму 4 чисел Фибоначчи и 5 ПРОСТЫХ чисел.
1 генератор выдает числа по очереди 0 1 1 2
2 генератор выдает числа по очереди 2 3 5 7 11
Оба подписчика работают каждый со своим стеком данных и пишет в одну и ту же запись в БД
В БД в результате обработки этих данных будет результат
| sum | count_fib | count_prime |
| 32 | 4 | 5