

Отчет о проведении тестирования.

Время проведения: 24.05.2020

1. Цель теста

Бизнес-цель теста – получение навыков работы с индексами и нагрузочного тестирования.

Техническая цель теста – оценка производительности с использованием wrk/wrk2 с использованием СУБД MySQL 8.0.20.

В испытании участвовало 4 операций.

Таблица 1. Модель нагрузки.

1	Нагрузка с размером innodb buffer pool size = 32Mb. Без использования индексов.
2	Нагрузка с размером innodb buffer pool size = 32Mb. С использованием индексов.
3	Нагрузка с размером innodb buffer pool size = 1Gb. Без использования индексов.
4	Нагрузка с размером innodb buffer pool size = 1Gb. С использованием индексов.

Было произведено 4 эксперимента с разным количеством запросов в секунду.

Таблица 2. Эксперименты.

1	1 поток 1 соединение 1 запрос в секунду
2	4 потока 5 соединений 10 запросов в секунду
3	8 потоков 50 соединений 100 запросов в секунду
4	8 потоков 500 соединений 1000 запросов в секунду

2. Выводы

1. Максимальная производительность системы – 291 запроса в секунду при 8-ми потоковом тесте и 100 соединениях.
2. Время отклика / выполнения на уровне максимальной производительности по операциям составляло от 8,3 миллисекунд до 2,1 секунды. Так же были просадки после 90 перцентиля в диапазоне 4.4 секунды.
3. До применения индекса нагрузка в 1000 запросов в секунду практически не работала, запросы после 10 перцентиля закрывались по таймауту в 60 секунд, выставленному в R2DBC пуле соединений. После применения индекса, на минимальных ресурсах, запросы также отрабатывали в пределах 90 перцентиля и было всего 2 таймаута.

Настройки, запрос, создание индекса

4. При тестировании использовался docker image: mysql:8.0.20
5. Первые две операции проводились с настройками MySQL:
max_connections = 1100
innodb_flush_method = O_DIRECT
innodb_buffer_pool_size = 33554432
6. Последующие операции с настройками:
max_connections = 1100
innodb_flush_method = O_DIRECT
innodb_buffer_pool_size = 1073741824
7. Создание индекса проводилось следующим запросом:
CREATE INDEX user_info_first_name_sur_name_7045
USING BTREE ON user_info (first_name, sur_name);
8. Запрос и его план выполнения после создания индекса:
EXPLAIN FORMAT=JSON SELECT id, first_name, sur_name, age, sex, city
FROM user_info WHERE first_name LIKE "Aden%" AND sur_name LIKE "a%";

```
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "19067.66"
    },
    "table": {
      "table_name": "user_info",
      "access_type": "range",
      "possible_keys": [
        "user_info_first_name_sur_name_7045"
      ],
      "key": "user_info_first_name_sur_name_7045",
      "used_key_parts": [
        "first_name"
      ],
      "key_length": "774",
      "rows_examined_per_scan": 42372,
      "rows_produced_per_join": 4707,
      "filtered": "11.11",
      "index_condition": "((`db`.`user_info`.`first_name` like 'Aden%') and (`db`.`user_info`.`sur_name` like 'a%'))",
      "cost_info": {
        "read_cost": "18596.91",
        "eval_cost": "470.75",
        "prefix_cost": "19067.66",
        "data_read_per_join": "5M"
      },
      "used_columns": [
        "id",
        "first_name",
        "sur_name",
        "age",
        "sex",
        "city"
      ]
    }
  }
}
```
9. Составной индекс (по двум колонкам) был выбран так как в запросе использовалось условие поиска по этим двум колонкам, с логическим выражением AND.

Добавлены индексы: surname (минимальный необходимый набор, причем колонка с большей селективностью) и surname first name.

```
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "67965.56"
    },
    "table": {
      "table_name": "user_info",
      "access_type": "range",
      "possible_keys": [
        "user_info_sur_name_first_name_9785"
      ],
      "key": "user_info_sur_name_first_name_9785",
      "used_key_parts": [
        "sur_name"
      ],
      "key_length": "774",
      "rows_examined_per_scan": 151034,
      "rows_produced_per_join": 16779,
      "filtered": "11.11",
      "index_condition": "((`db`.`user_info`.`first_name` like 'Aden%') and (`db`.`user_info`.`sur_name` like 'a%'))",
      "cost_info": {
        "read_cost": "66287.57",
        "eval_cost": "1677.99",
        "prefix_cost": "67965.56",
        "data_read_per_join": "18M"
      },
      "used_columns": [
        "id",
        "first_name",
        "sur_name",
        "age",
        "sex",
        "city"
      ]
    }
  }
}

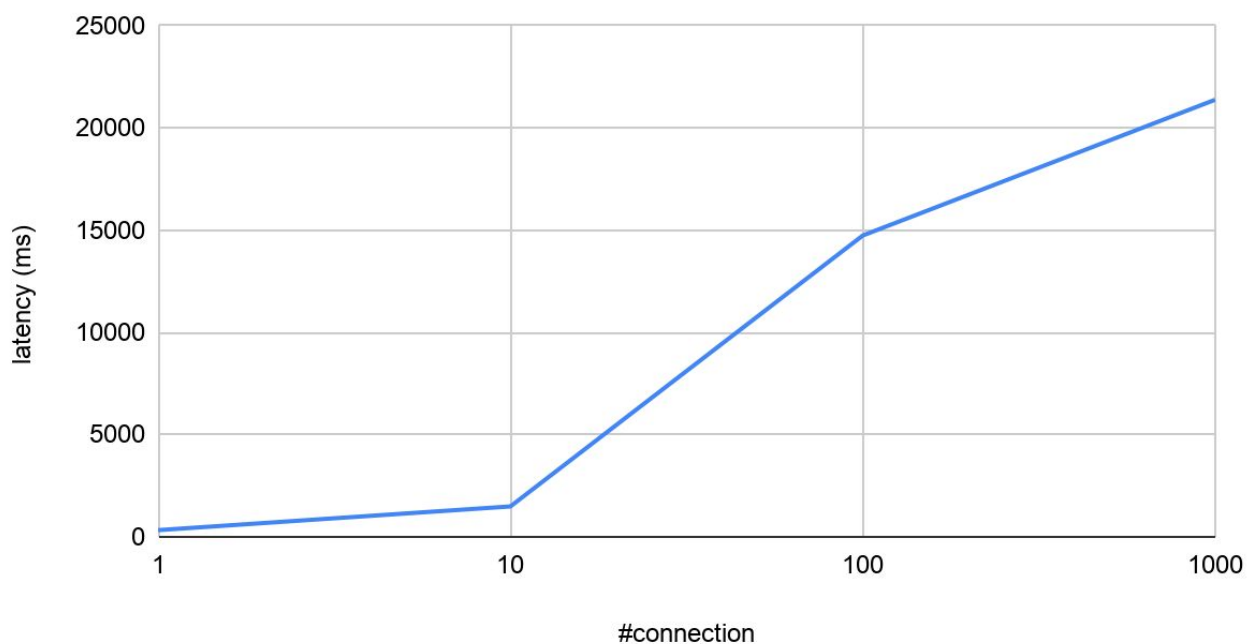
{
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "67965.56"
    },
    "cost_info": {
      [36/9341] "query_cost": "67965.56"
    },
    "table": {
      "table_name": "user_info",
      "access_type": "range",
      "possible_keys": [
        "user_info_sur_name_first_name_9785"
      ],
      "key": "user_info_sur_name_first_name_9785",
      "used_key_parts": [
        "sur_name"
      ],
      "key_length": "774",
      "rows_examined_per_scan": 151034,
      "rows_produced_per_join": 16779,
      "filtered": "11.11",
      "index_condition": "((`db`.`user_info`.`first_name` like 'Aden%') and (`db`.`user_info`.`sur_name` like 'a%'))",
      "cost_info": {
        "read_cost": "66287.57",
        "eval_cost": "1677.99",
```

```
    "prefix_cost": "67965.56",  
    "data_read_per_join": "18M"  
  },  
  "used_columns": [  
    "id",  
    "first_name",  
    "sur_name",  
    "age",  
    "sex",  
    "city"  
  ]  
}  
}
```

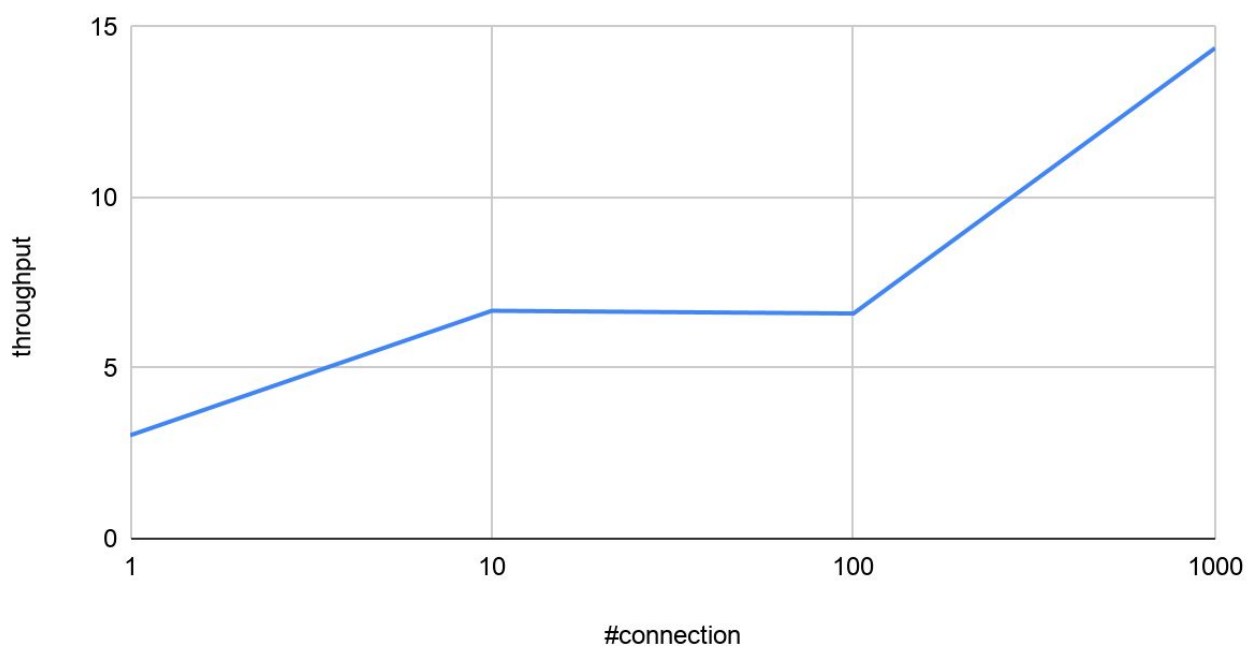
3. Графики

3.1. (по оси x- кол-во соединений, по оси y latency или throughput) графики до индекса.

latency (ms) vs. #connection. Before index.

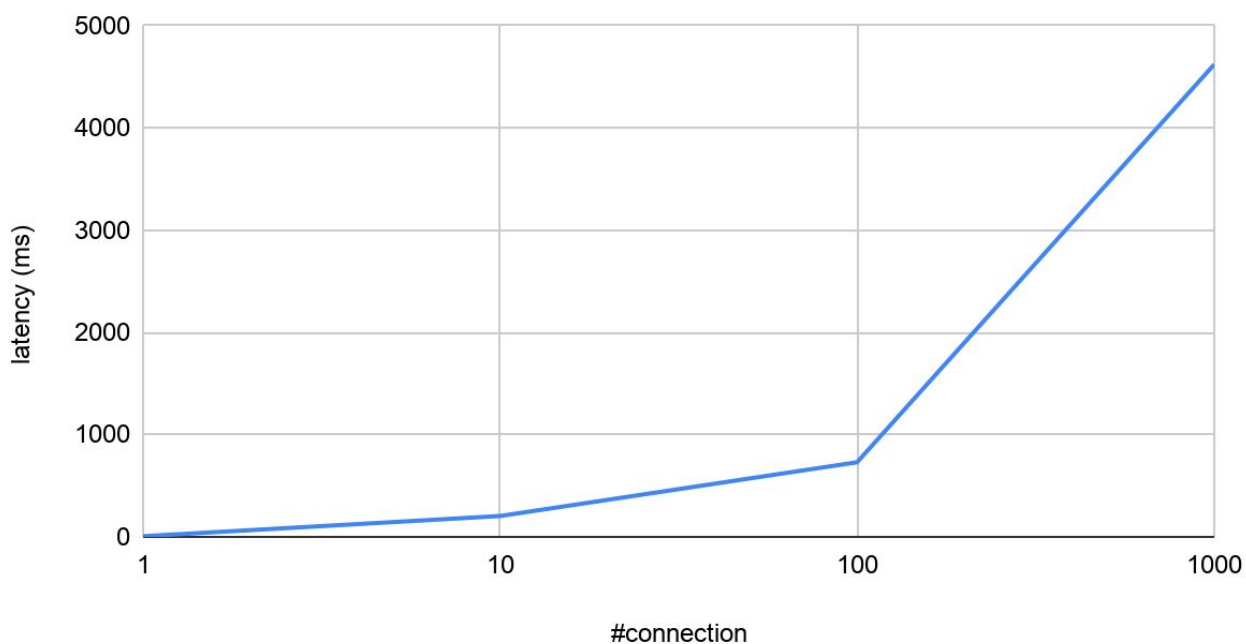


throughput vs. #connection. Before index.

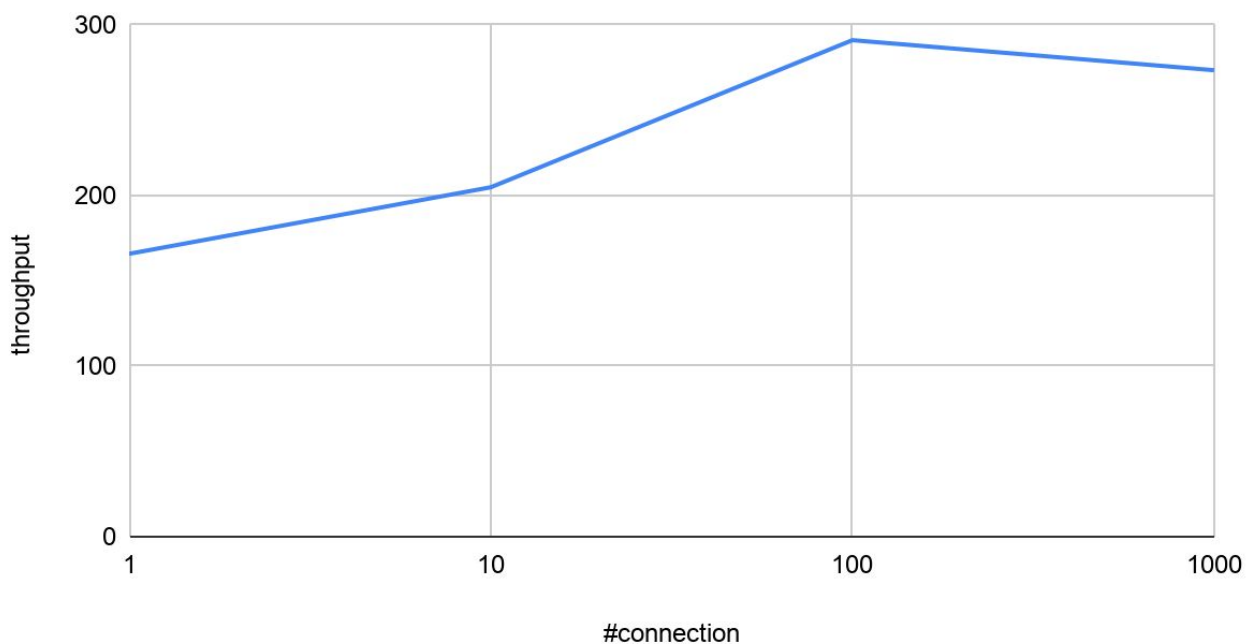


3.2. (по оси x- кол-во соединений, по оси y latency или throughput) графики с индексом.

latency (ms) vs. #connection. With index.



throughput vs. #connection. With index.



3.3. Графики до применения индекса

График 1. Нагрузки 1/10/100 при 32 Mb pool size.

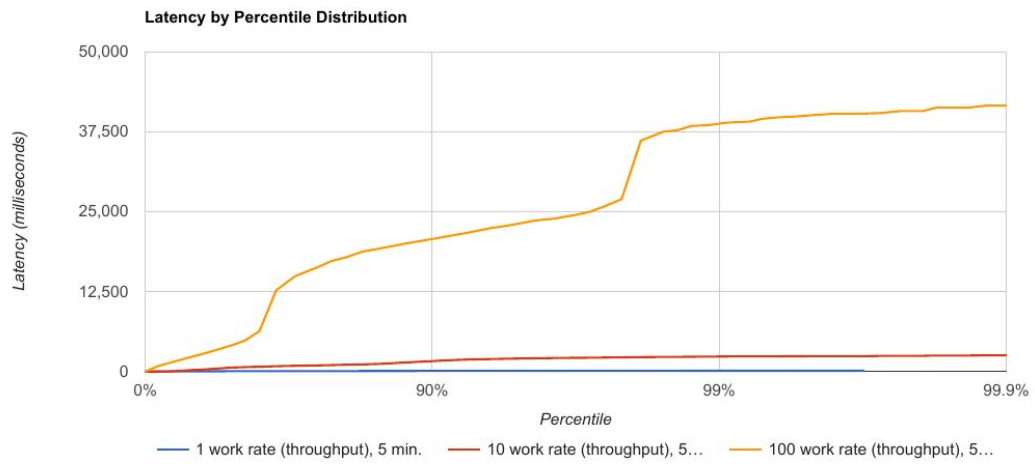


График 2. Нагрузки 10/100/1000 при 32 Mb pool size.

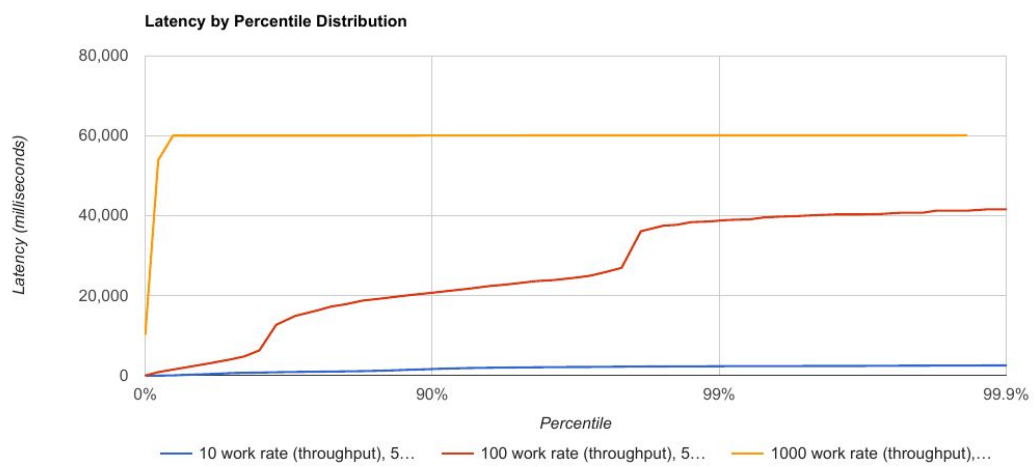


График 3. Нагрузки 1/10/100 при 1 Gb pool size.

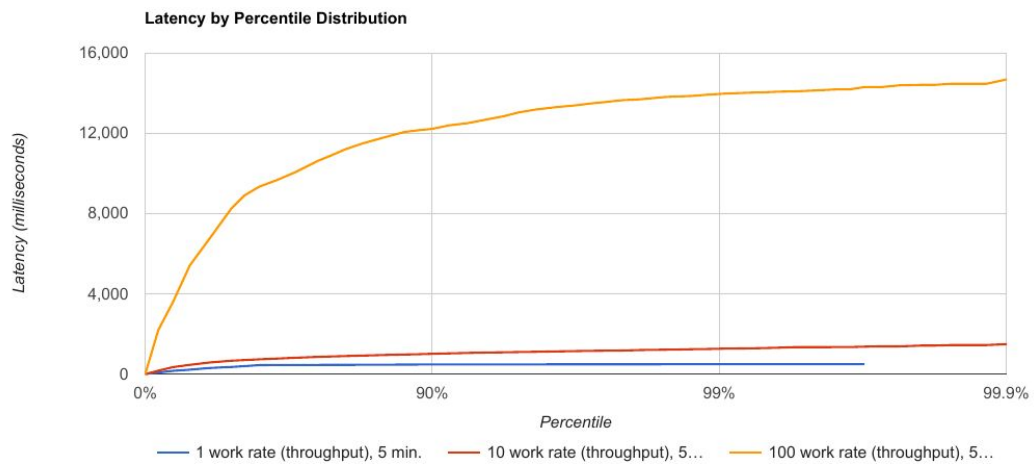
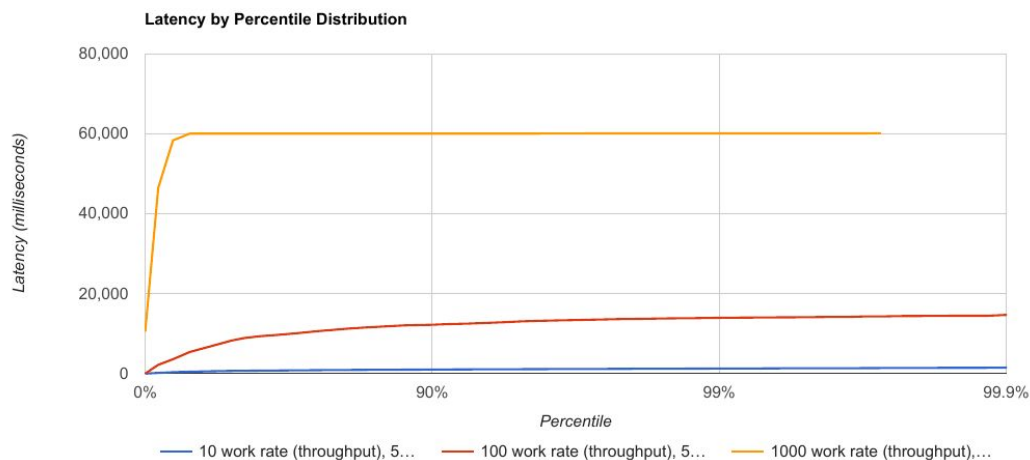


График 4. Нагрузки 10/100/1000 при 1 Gb pool size.



3.4. Графики после применения индекса

График 5. Нагрузки 1/10/100 при 32 Mb pool size.

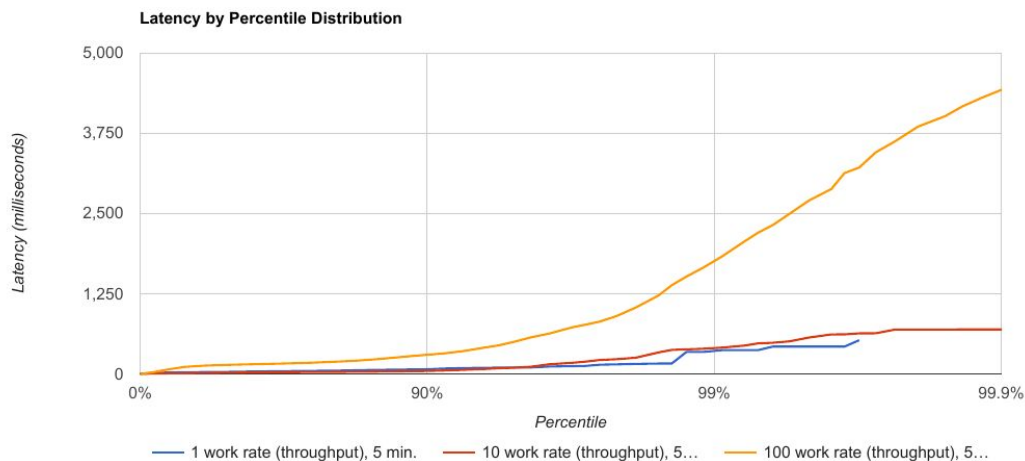


График 6. Нагрузки 10/100/1000 при 32 Mb pool size.

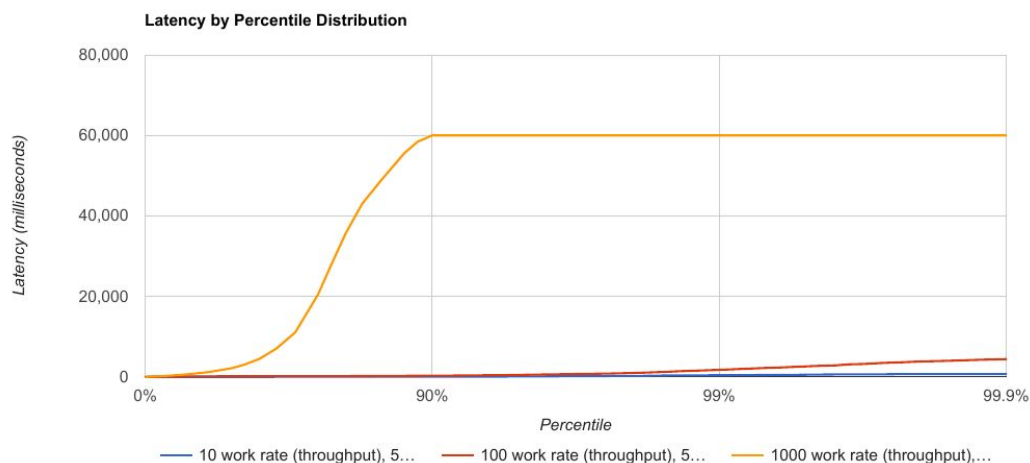


График 5. Нагрузки 1/10/100 при 1 Gb pool size.

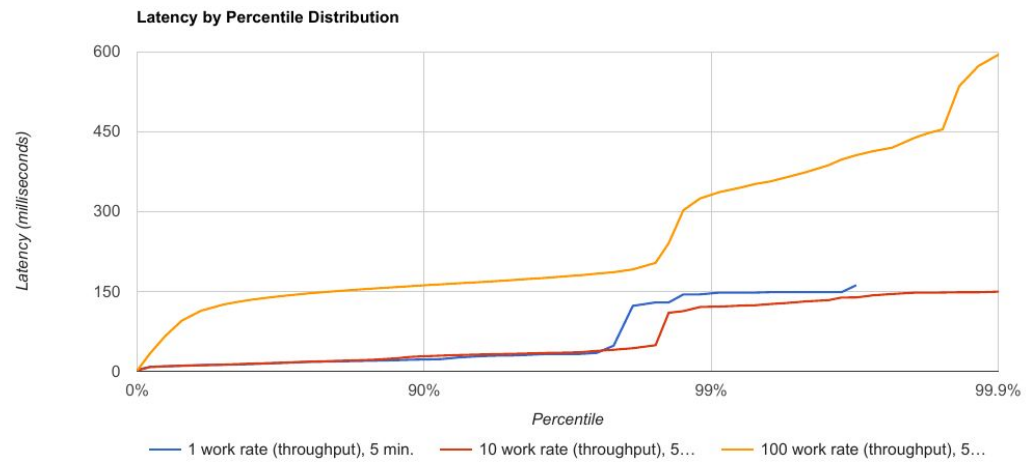


График 6. Нагрузки 10/100/1000 при 1 Gb pool size.

