

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 2.0
Дата 23.05.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

- 1.1. Определение максимальной производительности.
- 1.2. Соответствие системы предъявляемым требованиям.

2. Требования:

- Количество запросов и событий от модулей - до 30 в секунду
- Для пользователей допускаются задержки в интерфейсе до 5 секунд.
- Поддержка общего объёма базы данных ядра не менее 10 Гб.
- Скорость передачи данных через Ядро не менее 10 МБ/с

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность составила 33 запроса в секунду, что соответствует заявленным требованиям при общем объеме базы данных ядра 10 Гб. Время ответа запроса POST /api/v1/permission/state/module/rebuild превышает 5 секунд (таблица 1). Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU (Рис. 3). Скорость передачи данных через ядро равна 3 МБ/с (Рис. 8).

Запрос	50% line, ms	75% line, ms	95% line, ms
POST /api/v1/permission/state/module/rebuild	6875	10717	21506

Таблица.1. Перцентили времен ответа запроса при стабильной нагрузке

Большой разброс времен ответа по этому запросу связан с тем, что разные модули, участвовавшие в тестировании, содержат разное количество объектов. Чем больше объектов в модуле (сущностей, форм, событий и т. д.) тем больше время ответа.

Рекомендации: проанализировать возможность оптимизации медленного запроса, проанализировать причину высокой утилизации спу при помощи профилирования.

5. Модель нагрузки:

В тесте использовались следующие запросы:

Нагружаемые эндпоинты микросервиса:	%%
POST /api/v1/module/{moduleCode}/entity/{entityCode}	5
POST /api/v1/module/{moduleCode}/event/{eventCode}/export	5
POST /api/v1/module/{moduleCode}/event/{eventCode}/import	5
POST /api/v1/module/{moduleCode}/function/{functionCode}	5
POST /api/v1/relationship/entity/rebuild	1
GET /odata/v1/stateEntity	2.3
GET /odata/v1/stateFunction	1.7
POST /notification-api/api/v1/notification/send	10
GET /odata/v1/schemaregistry	5
GET /odata/v1/mappingregistry	5
POST /api/v1/eventproducer/message	5
POST /api/v1/eventconsumer/state	5

POST /api/v1/permission/state/form/rebuild	2
POST /api/v1/permission/state/module/rebuild	1.5
PUT /api/v1/permission/role/{role}	1.5
GET /api/v1/files/form/{formId}/manifest	10
POST /api/v1/mappingservice/state	15
POST /api/v1/businessProcess/{id}/execute	3.5
POST /api/v1/businessProcess/usertask/complete	6.5
GET /odata/v1/entity?\$filter=Id eq '{moduleEntity}'	1
GET /odata/v1/module?\$filter=Code eq '{moduleCode}'	1
GET /odata/v1/moduleevents?\$filter=Id eq '{moduleEvents}'	1
POST /api/v1/module/testFunction/{testfunctionId}/trigger	1
PUT /api/v1/module/testFunction/executionlog	1
POST /api/v1/module/manifest	-

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 24 запроса за 8 секунд. Запрос POST /api/v1/module/manifest (добавление манифеста) осуществлялся в один поток с интенсивностью один запрос в 4 минуты и обновлял существующий модуль «FM.NSS» с добавлением формы и сущности. Объем базы данных ядра 10Гб.

6. Использование заглушек:

Для работы микросервис «BP-management» взаимодействует с сервисом «Camunda». В ходе тестирования сервис «Camunda» был заглушен с использованием утилиты «Wiremock», а именно, следующие запросы:

- POST /engine-rest/task/{taskId}/complete
- POST /engine-rest/process-definition/key/{definitionKey}/start
- GET /engine-rest/history/task

6. Проведенные оптимизации:

Изменение в API:

POST /api/v1/relationship/entity/rebuild - перестроить связи сущностей. Добавлена модель данных (перестроение для модуля или сущности модуля). Убрана возможность перестроения всех сущностей в ядре.

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейная нагрузка на сервис согласно следующему профилю:

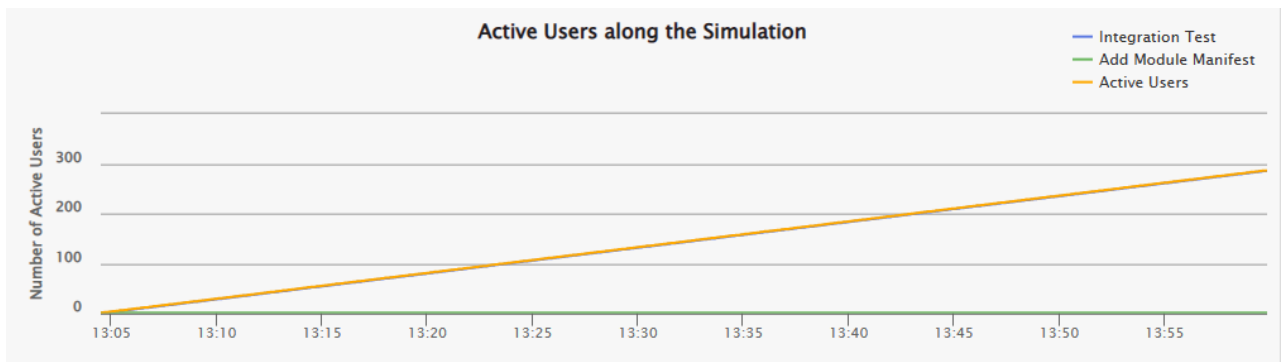


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную 33 запросам в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 при достижении этого значения перестали расти и ресурсы CPU были практически полностью утилизированы.

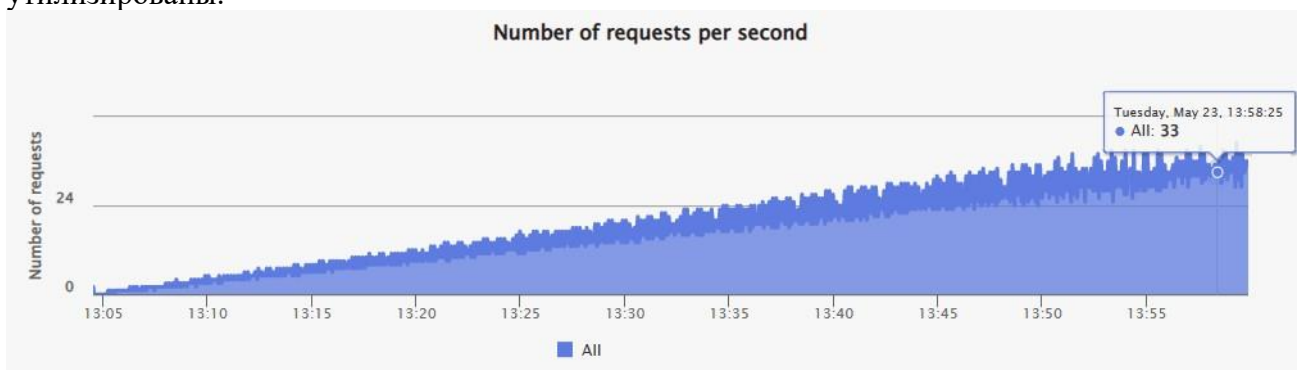


Рис.2. Пропускная способность для растущей нагрузки.

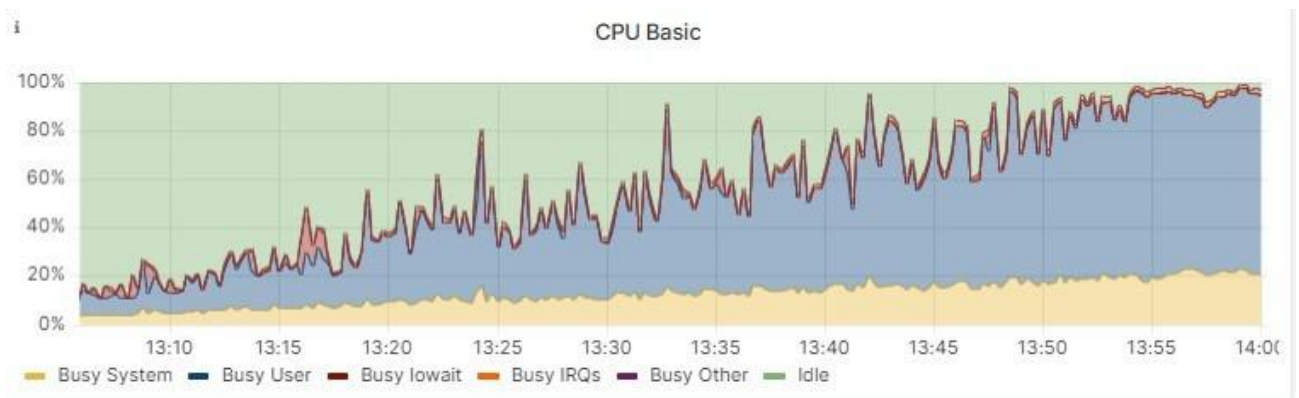


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

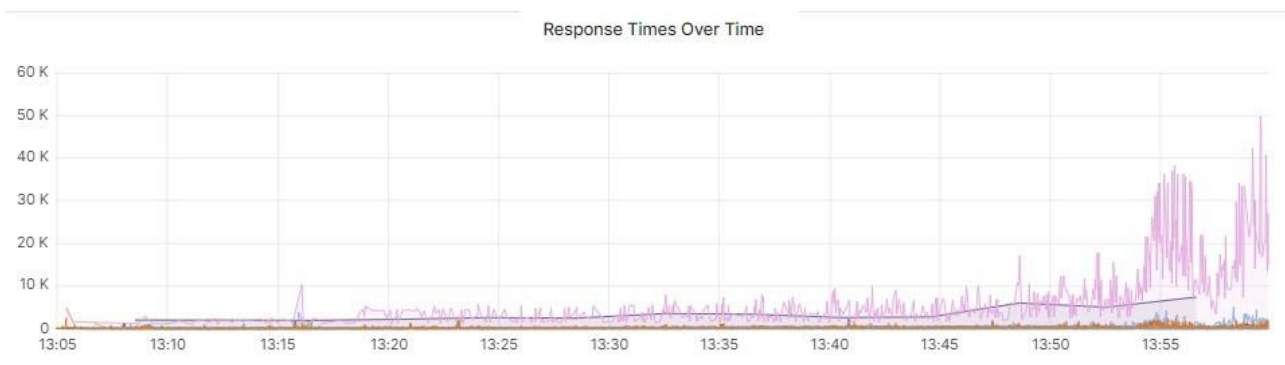


Рис.4. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки соответствующий 30 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

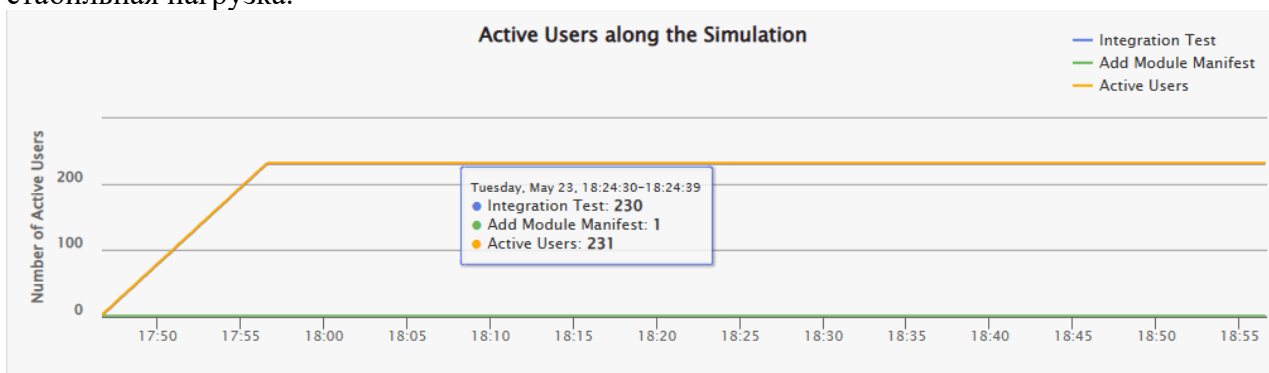


Рис.5 Профиль нагрузки для определения ключевых показателей.

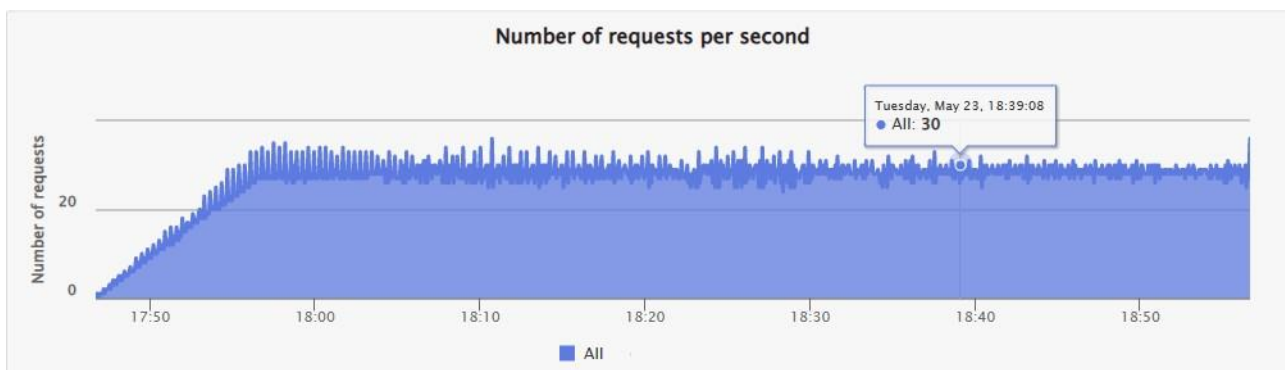


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

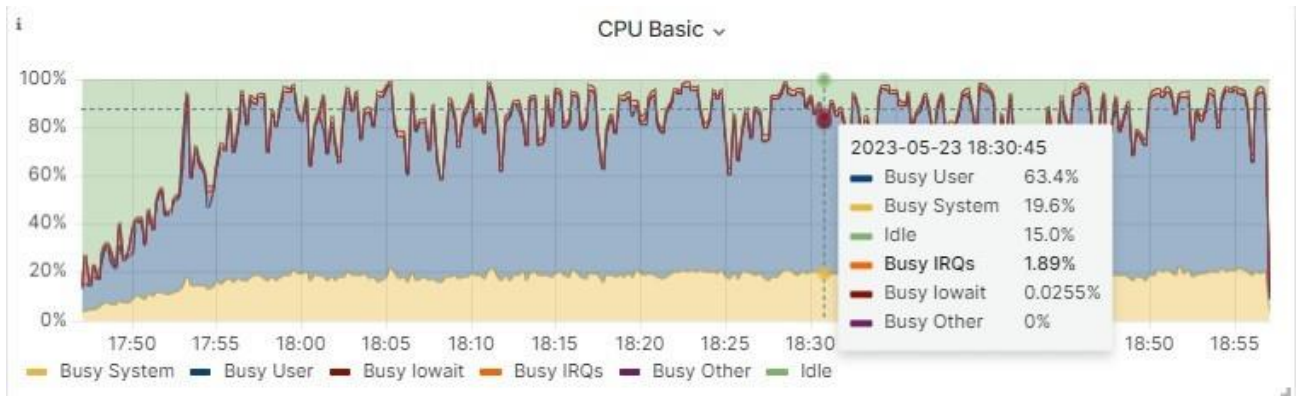


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 85%

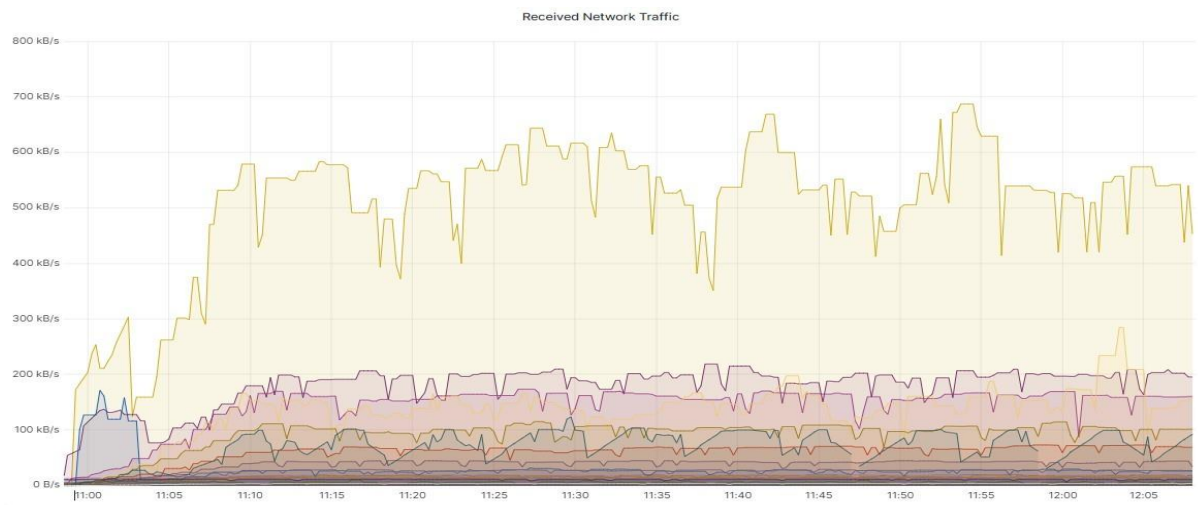


Рис 8. Скорость приема (receive) данных микросервисами при стабильной нагрузке.

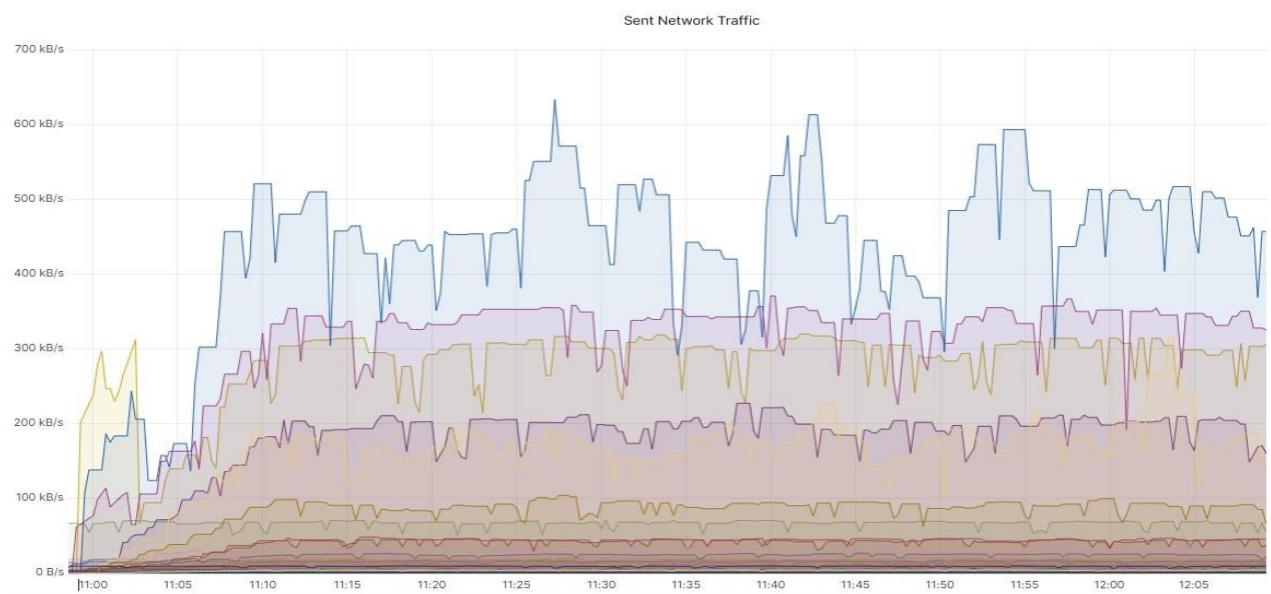


Рис 8. Скорость отправки (sent) данных микросервисами при стабильной нагрузке.

Суммарная скорость передачи данных микросервисами ядра около 3 МБ/с

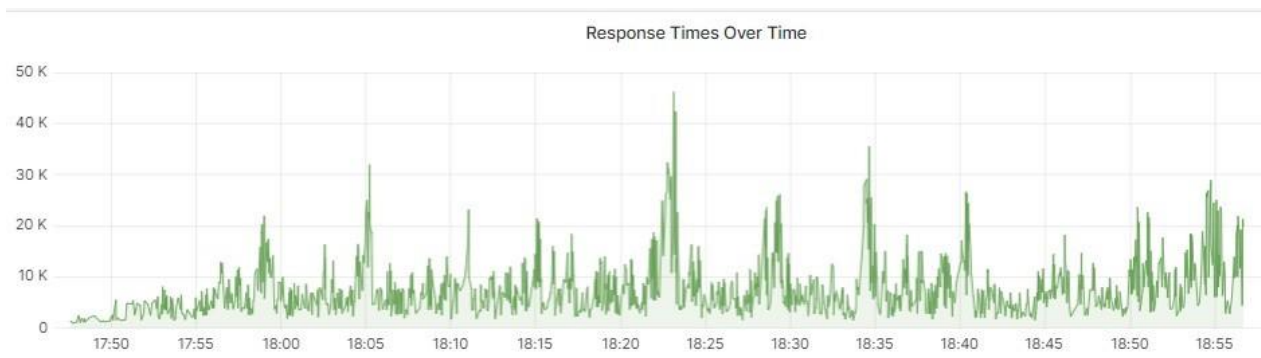


Рис.9. Времена ответа запроса POST /api/v1/permission/state/module/rebuild при стабильной нагрузке.

Наиболее длительные времена ответа (на графике Рис. 8) у запроса POST /api/v1/permission/state/module/rebuild



Рис.10. Времена ответа запроса POST /api/v1/module/manifest при стабильной нагрузке.

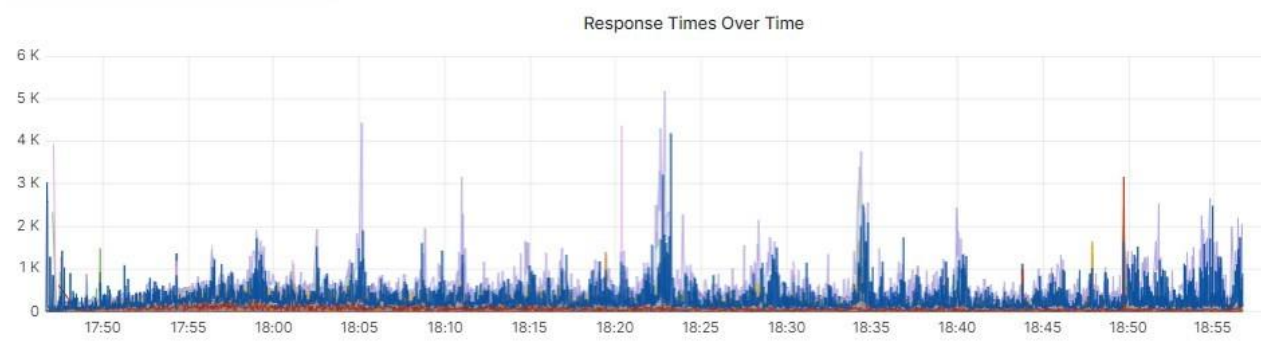


Рис.11. Времена ответа остальных запросов при стабильной нагрузке.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST /api/v1/module/{moduleCode}/entity/{entityCode}	5584	27	3179	100	81	125	203
POST /api/v1/module/{moduleCode}/event/{eventCode}/export	5603	56	2585	254	229	334	500
POST /api/v1/module/{moduleCode}/event/{eventCode}/import	5652	40	2937	218	191	289	449
POST	5642	26	4380	115	81	129	319

/api/v1/module/{moduleCode}/function/{functionCode}							
POST /api/v1/relationship/entity/rebuild	1104	53	2399	252	232	333	506
GET /odata/v1/stateEntity	2636	132	2624	345	336	414	536
GET /odata/v1/stateFunction	1895	77	2095	229	209	286	406
POST /notification-api/api/v1/notification/send	1103 8	40	3235	95	77	115	181
GET /odata/v1/schemaregistry	5675	11	2007	56	38	76	145
GET /odata/v1/mappingregistry	5635	12	2600	48	31	60	116
POST /api/v1/eventproducer/message							
POST /api/v1/eventconsumer/state	5629	21	2224	184	156	254	408
POST /api/v1/permission/state/form/rebuild	2244	126	6642	666	568	781	1447
POST /api/v1/permission/state/module/rebuild	1676	1036	46383	8474	6875	10717	21506
PUT /api/v1/permission/role/{role}	1639	15	3157	72	42	104	201
GET /api/v1/files/form/{formId}/manifest	11161	91	2326	331	302	426	609
POST /api/v1/mappingservice/state	1654 9	50	3650	350	307	453	728
POST /api/v1/businessProcess/{id}/execute	3977	10	1635	90	63	127	234
POST /api/v1/businessProcess/usertask/complete	7153	5	3110	26	13	26	64
GET /odata/v1/entity?\$filter=Id eq '{moduleEntity}'	1142	17	828	127	111	180	298
GET /odata/v1/module?\$filter=Code eq '{moduleCode}'	1089	10	1635	90	63	127	234
GET /odata/v1/moduleevents?\$filter=Id eq '{moduleEvents}'	1146	13	677	115	92	168	292
POST /api/v1/module/testFunction/{testfunctionId}/trigger	1113	35	951	150	130	202	319
PUT /api/v1/module/testFunction/executionlog	1179	5	2028	72	42	104	201
POST /api/v1/module/manifest	18	2281	9619	5074	4719	5911	8428

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО
ТЕСТИРОВАНИЯ
микросервиса
VR-Management

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 16.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Соответствие системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

- Количество запросов и событий от модулей - до 30 в секунду
- Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса “BP-management” составила 190 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. При такой производительности сервис стартует 63 бизнес процесса в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 95% составляют:

Запрос	95% line, ms
POST /api/v1/businessProcess/{id}/execute	440
POST /api/v1/businessProcess/usertask/complete	300

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис “BP-management”. В тесте использовался следующие запросы:

Нагружаемые эндпоинты микросервиса:	%%
POST /api/v1/businessProcess/{id}/execute	35
POST /api/v1/businessProcess/usertask/complete	65

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 3 запроса за 1 секунду.

6. Использование заглушек:

Для работы с бизнес процессами микросервис «BP-Management» взаимодействует с сервисом «Samunda». В ходе тестирования внешний сервис «Samunda» был заглушен с использованием утилиты «Wiremock», а именно, следующие запросы:

POST /process-definition/key/{key}/start

GET /history/task

POST /task/{id}/complete

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась ступенчатая нагрузка на микросервис согласно следующему профилю:

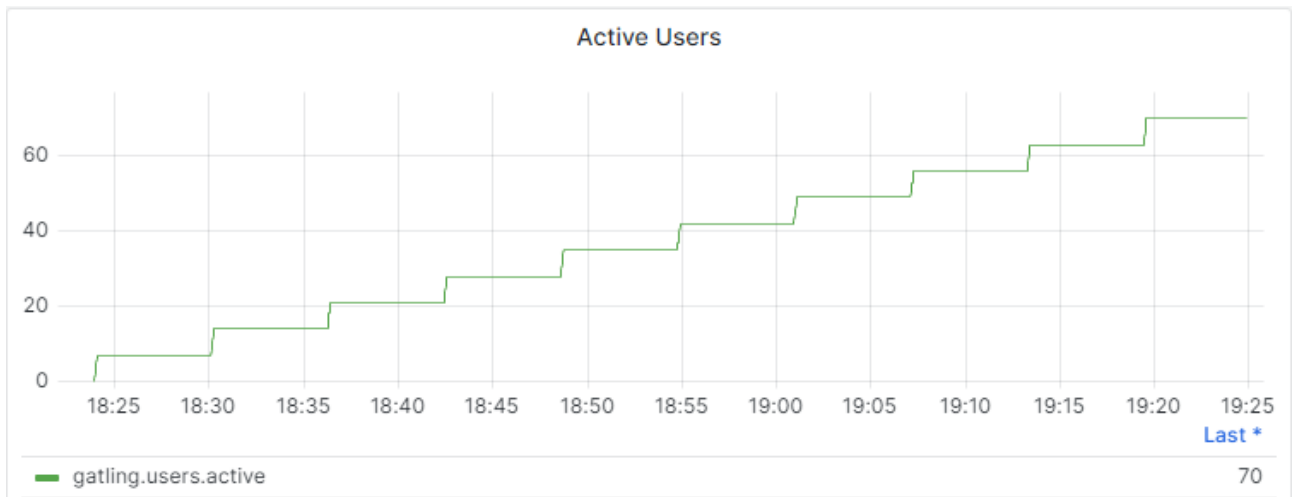


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность сервиса равную 190 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 перестали расти и ресурсы CPU были практически полностью утилизированы.

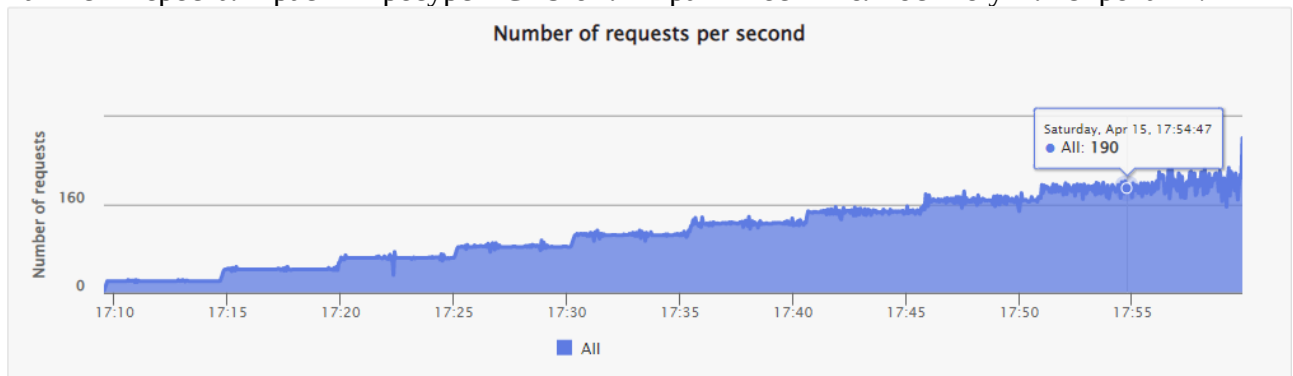


Рис.2. Пропускная способность для растущей нагрузки.

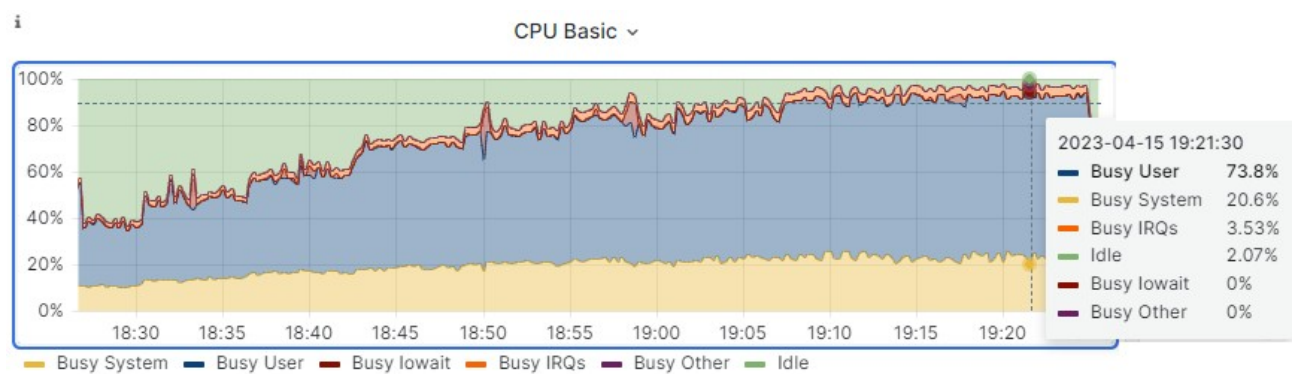


Рис.3. Утилизация CPU на хостовой виртуальной машине.

В момент достижения максимальной производительности сервис стартовал порядка 63 бизнес-процессов в секунду.



Рис.4. Количество стартовавших бизнес процессов в единицу времени

При достижении максимальной производительности времена ответов растут.

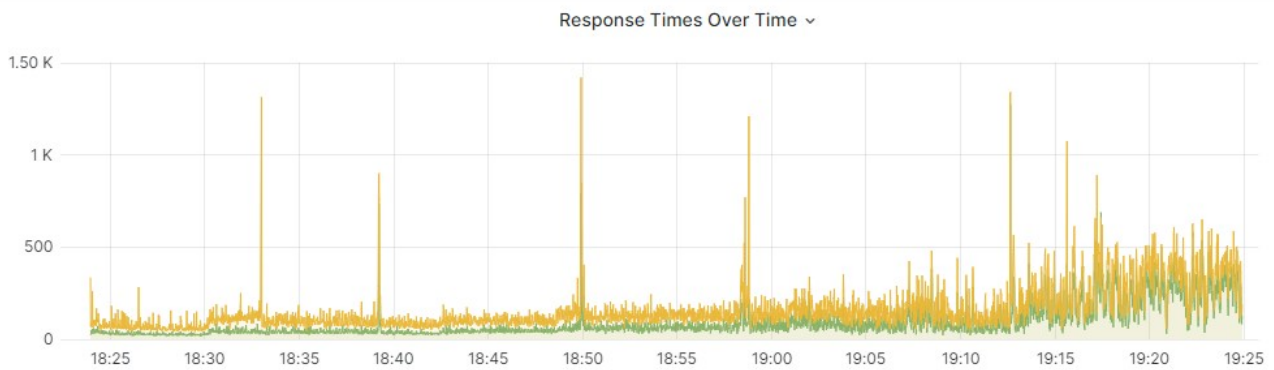


Рис.5. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 160 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

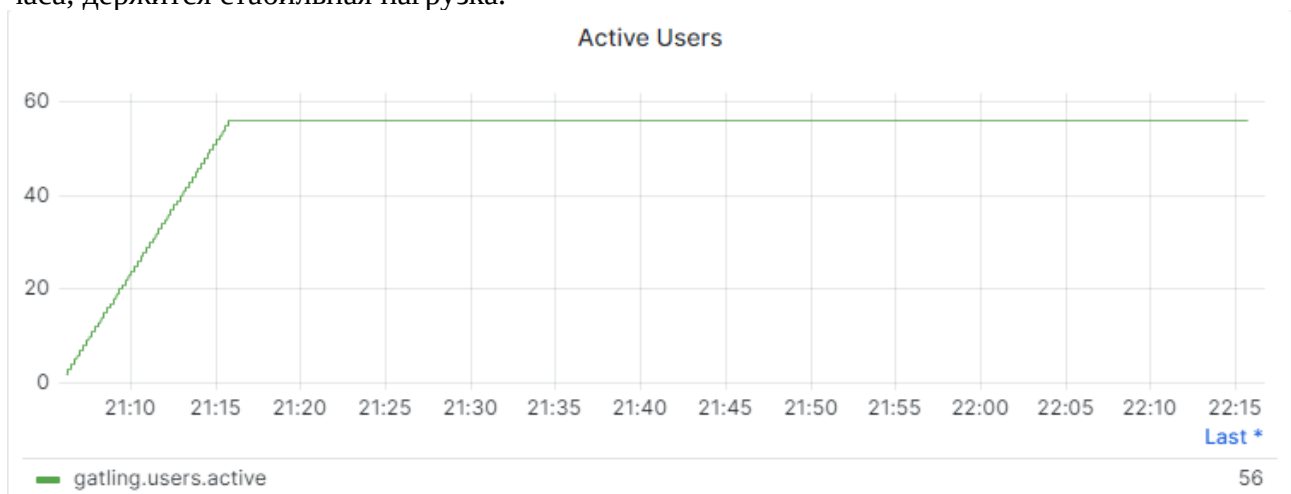


Рис.6. Профиль нагрузки для определения ключевых показателей.

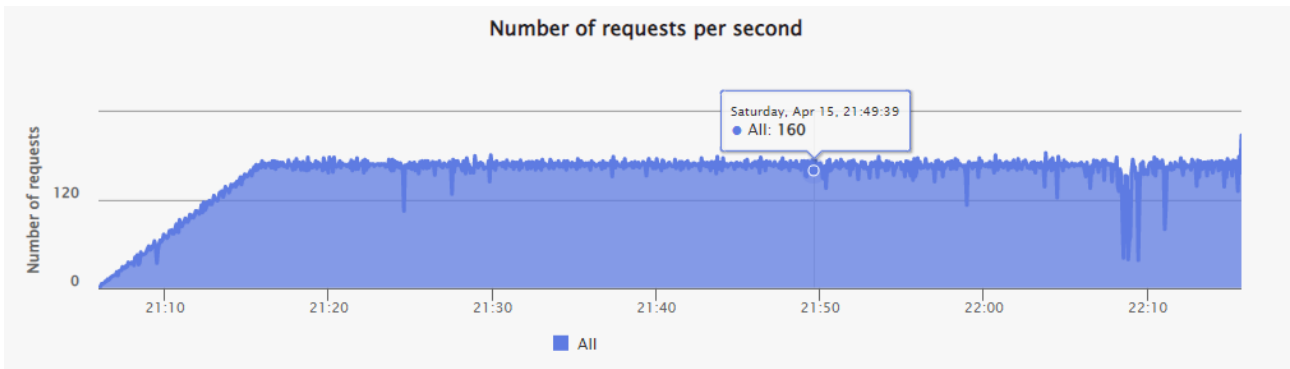


Рис.7. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

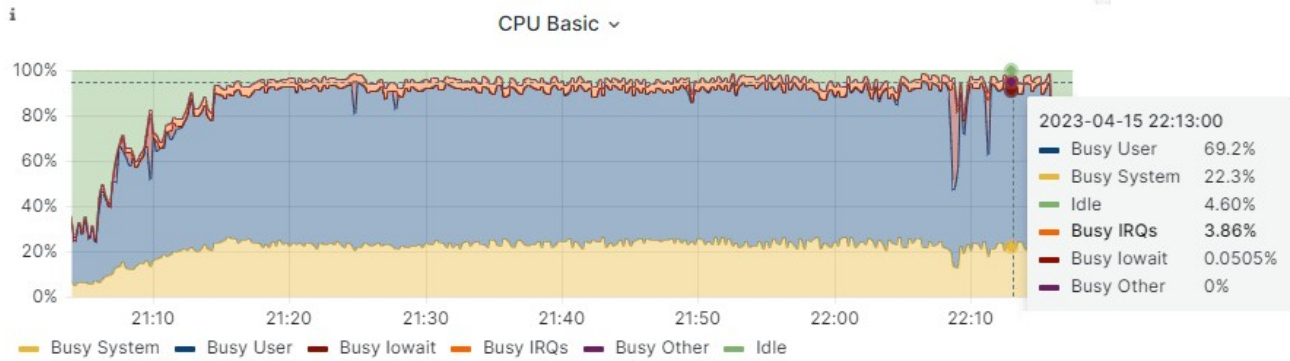


Рис. 8. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 95%

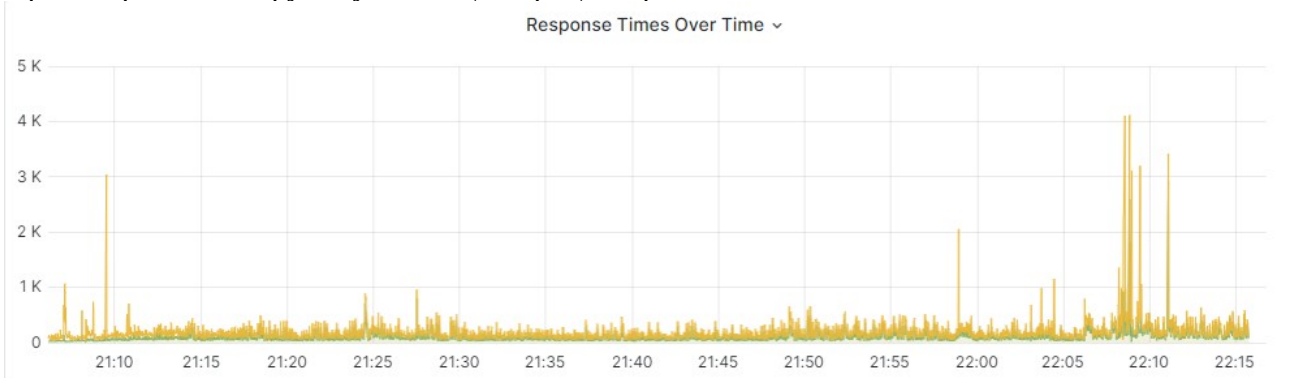


Рис.9. Времена ответа при стабильной нагрузке.

Времена ответа в целом стабильны на всем протяжении теста. В 22:08 наблюдается резкое увеличение времен ответа, связанное с увеличением утилизации дисковой подсистемы.

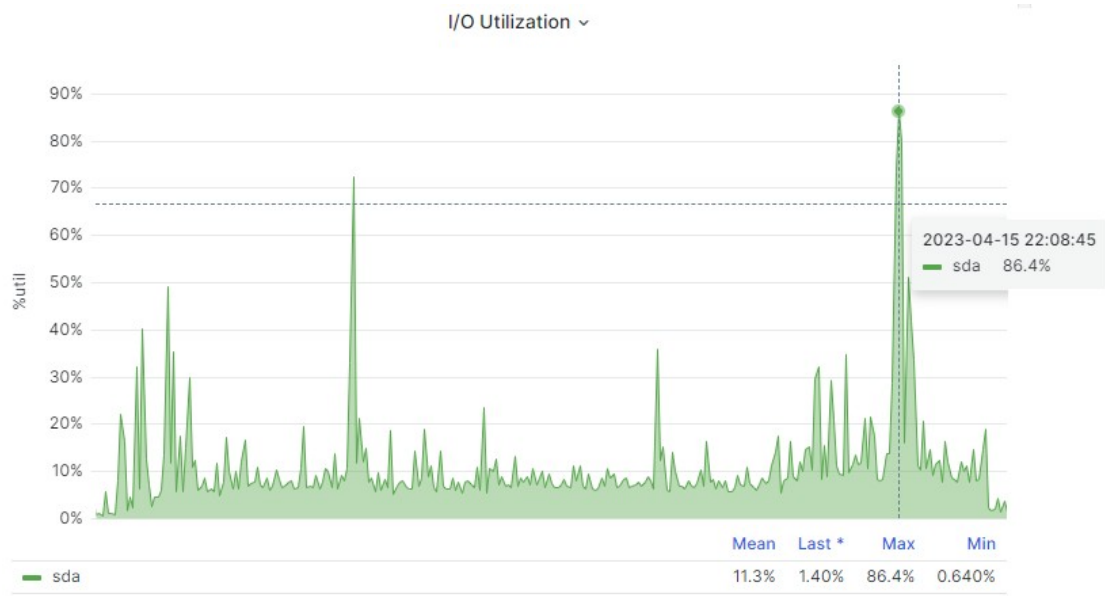


Рис.10. Утилизация дисковой подсистемы.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST /api/v1/businessProcess/{id}/execute	214952	13	6642	188	149	247	440
POST /api/v1/businessProcess/usertask/complete	429876	5	4213	110	80	152	300

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса DispatcherRelation

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 05.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

- 1.1. Определение максимальной производительности.
- 1.2. Соответствие системы предъявляемым требованиям.
- 1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

- Количество запросов и событий от модулей - до 30 в секунду (не более 70 запросов в минуту от одного модуля).
- Для пользователей допускаются задержки в интерфейсе до 5 секунд.
- Допустимые задержки получения данных между модулями до 30 сек.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 CPU, 16G RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса DispatcherRelation составила 50 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 80% составляют:

Запрос	95% line, ms
GET /odata/v1/stateFunction	1005
GET /odata/v1/stateEntity	1328
POST /api/v1/relationship/entity/rebuild	1046

5. Модель нагрузки:

Тестировался микросервис “DispatcherRelation”. В тесте использовались следующие запросы с распределением:

Нагружаемые эндпоинты микросервиса:	%%
POST /api/v1/relationship/entity/rebuild	20
GET /odata/v1/stateEntity	45
GET /odata/v1/stateFunction	35

В рамках одного потока нагрузка подавалась с интенсивностью 3 запроса в 2 секунды.

6. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась ступенчатая нагрузка на микросервис согласно следующему профилю:

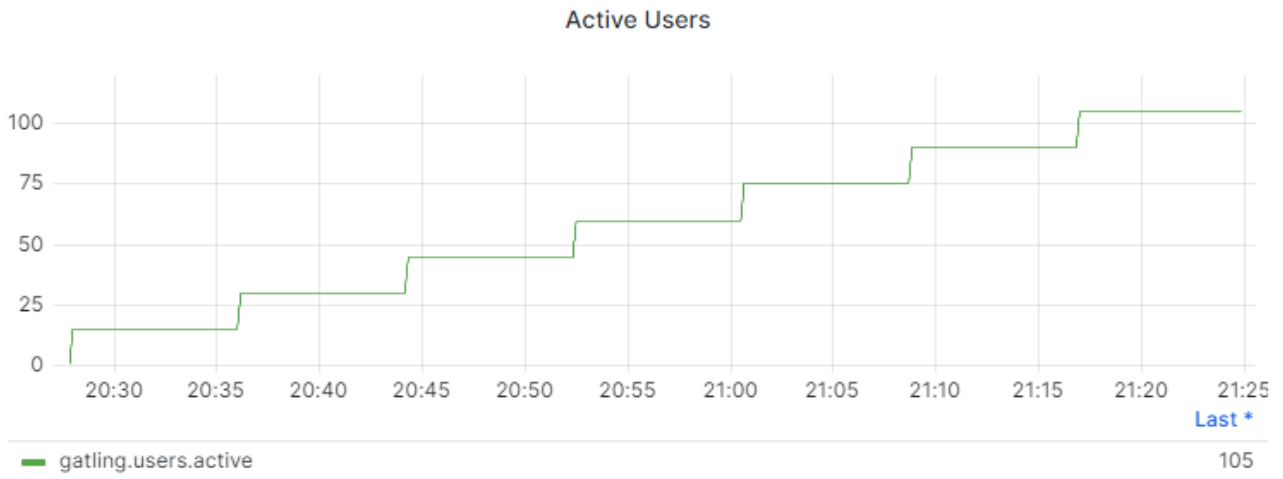


Рис.1 Профиль нагрузки с растущей нагрузкой.

В результате определили максимальную производительность равную 50 запросам в секунду. Такой вывод был сделан на основании того, что ресурсы CPU были практически полностью утилизированы.

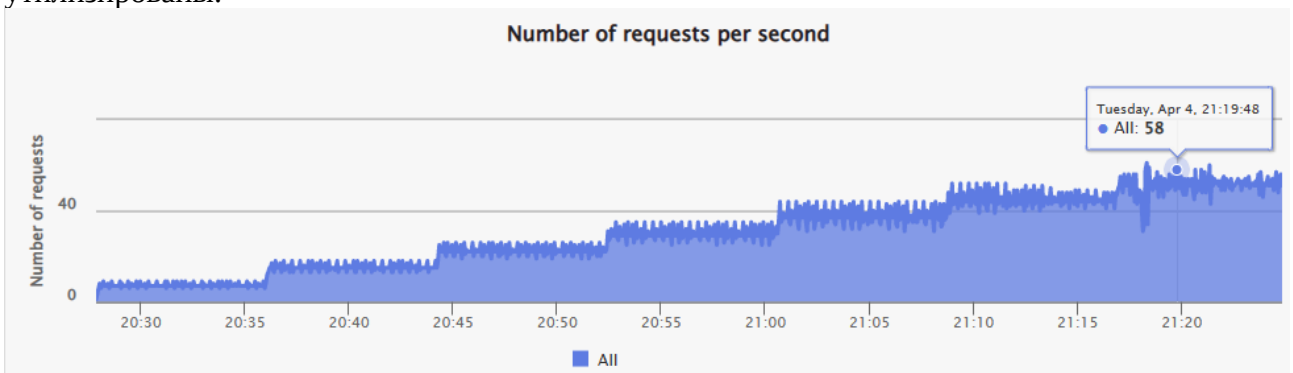


Рис.2. Пропускная способность для растущей нагрузки.

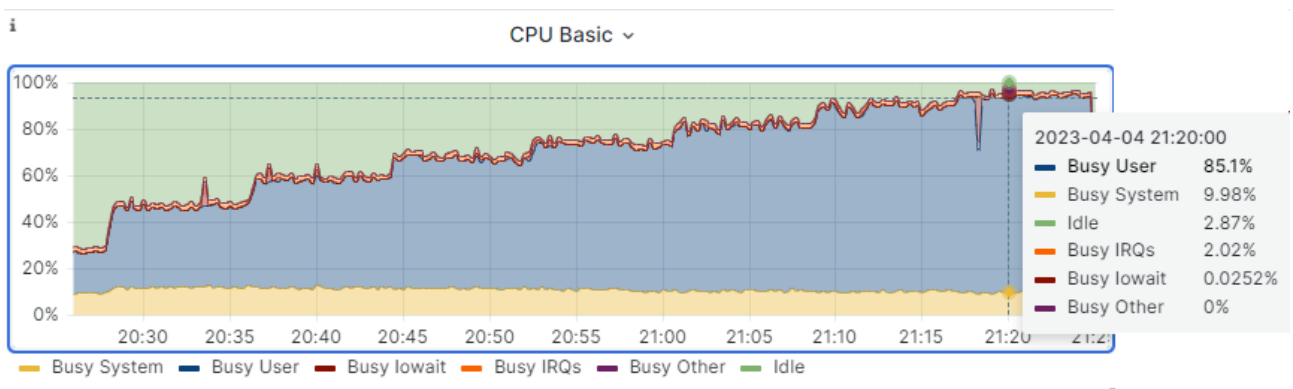


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

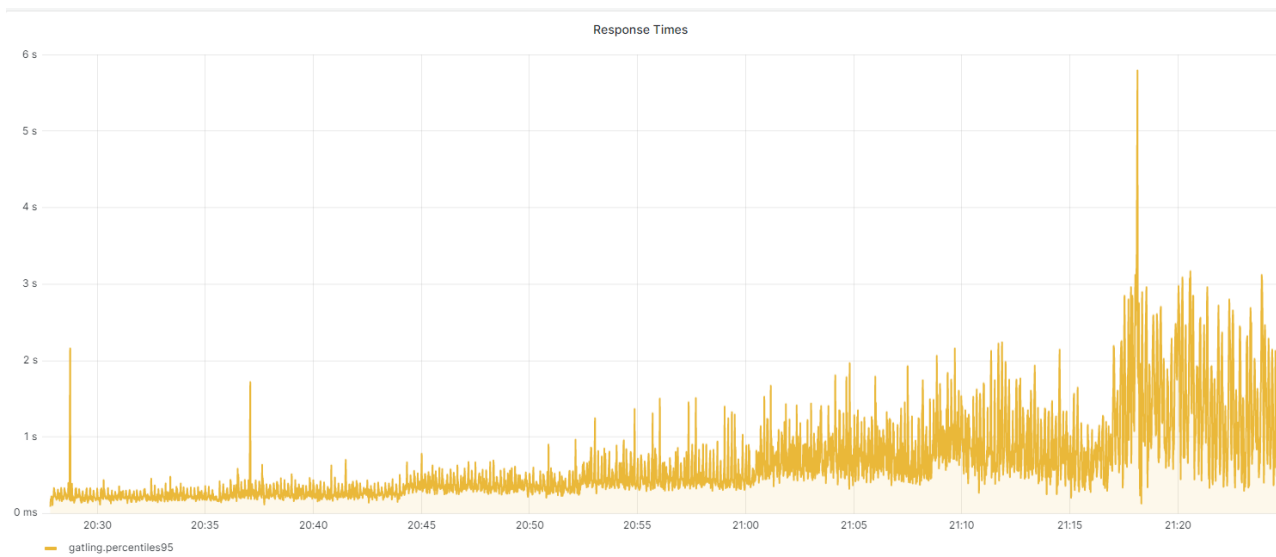


Рис.4. Динамика времен ответа 95% при растущей нагрузке.

За время проведения теста были ошибки 500 в запросе «POST /api/v1/relationship/entity/rebuild»

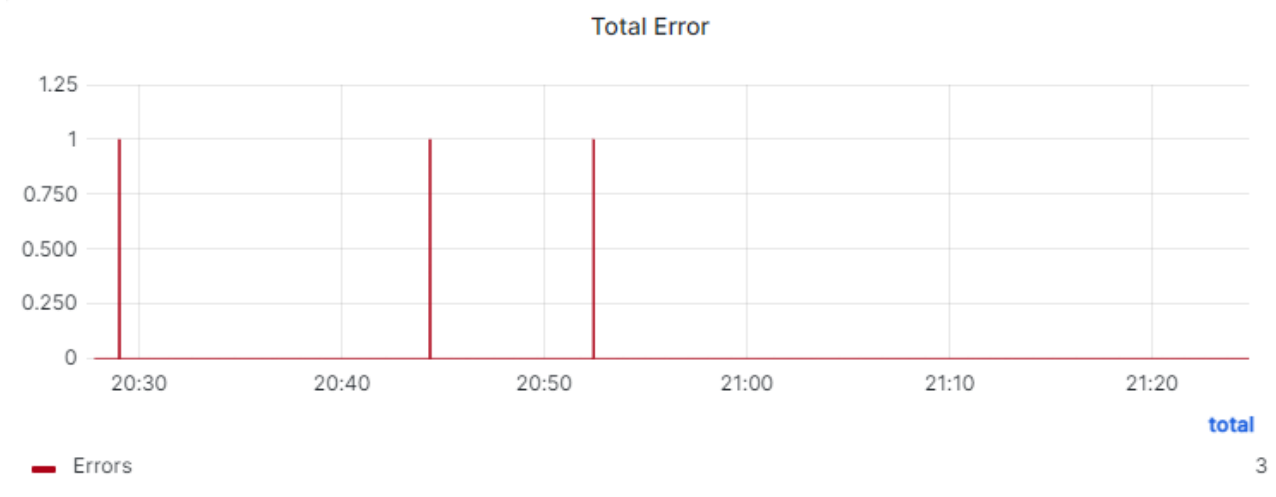


Рис.5. Ошибки при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки была выбрана предыдущая от максимальной ступени в 75 потоков. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

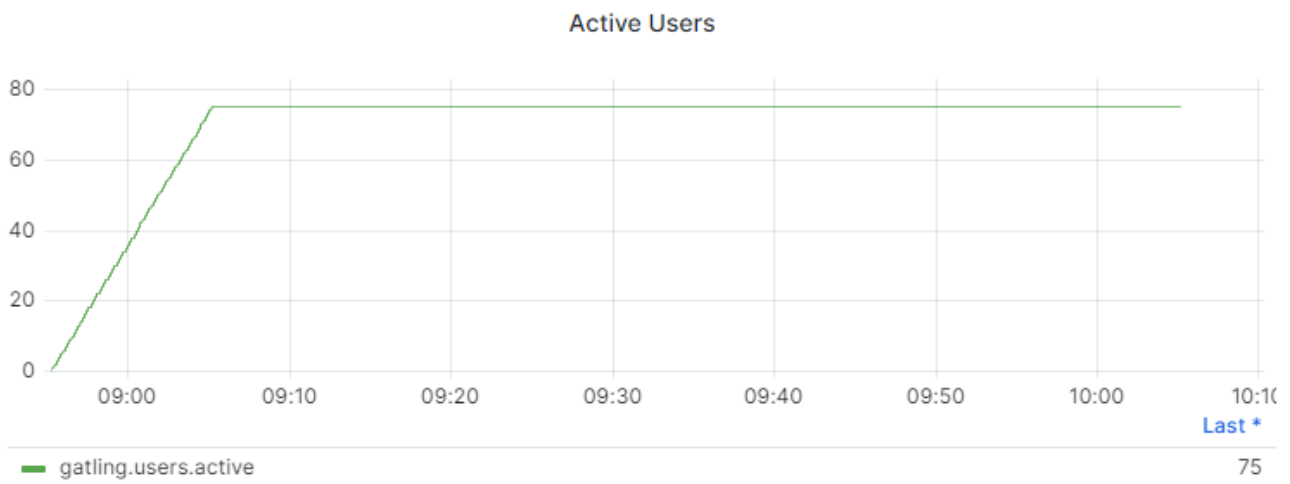


Рис.6 Профиль нагрузки с стабильной нагрузкой.

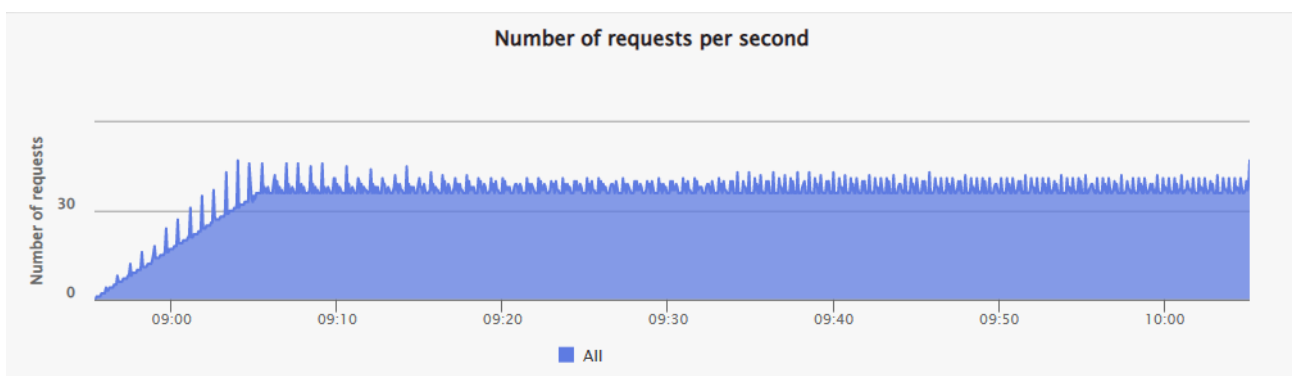


Рис.7. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

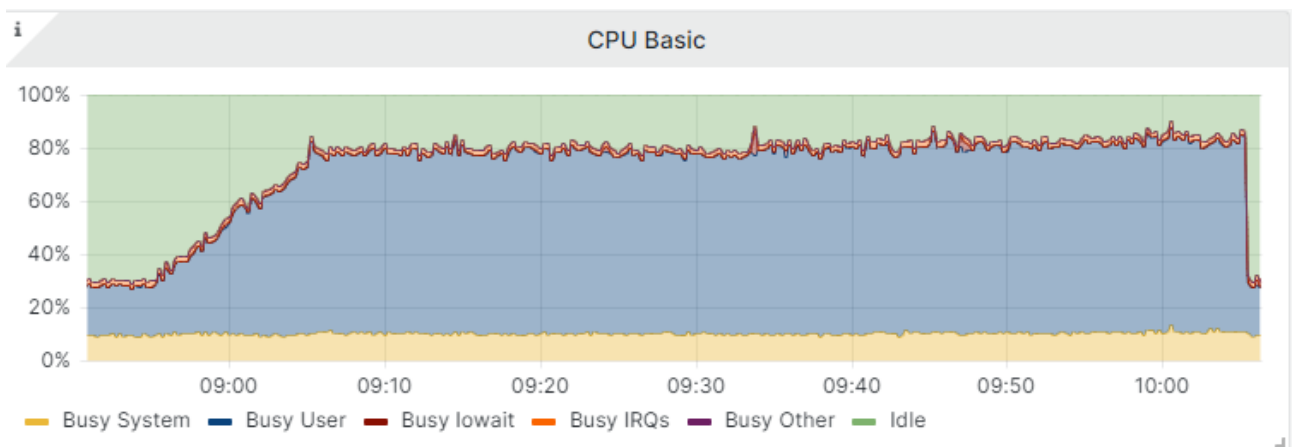


Рис 8. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила 80%

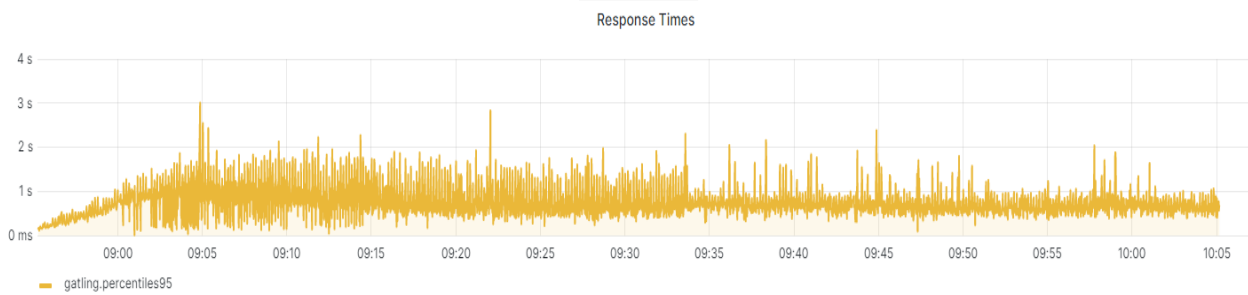


Рис.9. Времена ответа 95% при стабильной нагрузке.

После периода разгона наблюдаются флуктуации, которые связаны с быстрым ростом нагрузки. Через 30 минут система стабилизировалась.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
GET /odata/v1/stateFunction	50896	37	2841	470	418	590	1005
GET /odata/v1/stateEntity	65597	62	3420	656	601	819	1328
POST /api/v1/relationship/entity/rebuild	29611	4	2459	473	424	617	1046

Таблица 1. Времена ответа при стабильной нагрузке.

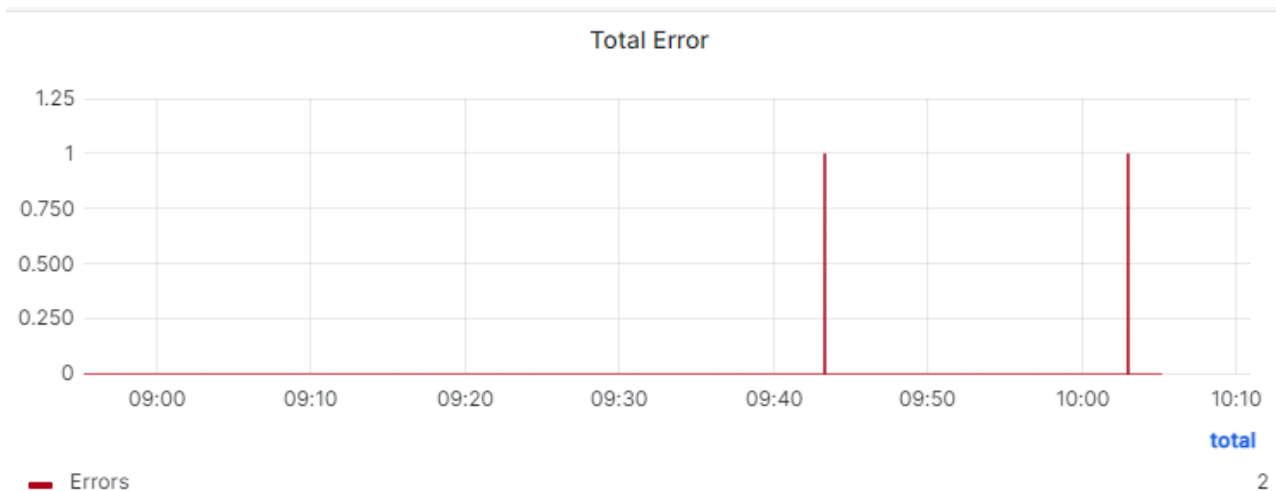


Рис.10. Ошибки при стабильной нагрузке.

Ошибки 500 в запросе POST /api/v1/relationship/entity/rebuild.

ОТЧЕТ НАГРУЗОЧНОГО
ТЕСТИРОВАНИЯ
микросервиса
DispatcherApi

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 02.05.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Определение соответствия системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

–Количество запросов и событий от модулей - до 30 в секунду

–Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса DispatcherApi составила 32 запроса в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 84% составляют:

Запрос	95% line, ms
POST/api/v1/module/{moduleCode}/entity/{entityCode}	934
POST /api/v1/module/{moduleCode}/event/{eventCode}/export	208
POST /api/v1/module/{moduleCode}/event/{eventCode}/import	213
POST /api/v1/module/{moduleCode}/function/{functionCode}	443

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис «DispatcherApi». В тесте использовался следующие запросы:

Нагружаемые эндпоинты микросервиса:	%%
POST/api/v1/module/{moduleCode}/entity/{entityCode}	30
POST /api/v1/module/{moduleCode}/event/{eventCode}/export	25
POST /api/v1/module/{moduleCode}/event/{eventCode}/import	25
POST /api/v1/module/{moduleCode}/function/{functionCode}	20

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в секунду.

6. Использование заглушек:

Для работы микросервис «DispatcherApi» взаимодействует с микросервисом «ManagementApi», «DispatcherRelationApi», «UserControlApi», «LicenseApi», «SchemaRegistryApi». В ходе тестирования эти сервисы были заглушены с использованием утилиты «Wiremock», а именно, следующие запросы:

```
GET /v1.0/invoke/core_management/method/odata/v1/Module?$filter=Code eq '{moduleCode}'
```

```
GET /v1.0/invoke/core_management/method/odata/v1/ModuleEvents?$filter=Code eq '{eventCodeExp}' and ModuleId eq '{moduleIdExp}'
```

```

GET /v1.0/invoke/core_management/method/odata/v1/ModuleEvents?$filter=Code eq
'{eventCodeImp}' and ModuleId eq '{moduleIdImp}'
GET /v1.0/invoke/core_schema_registry/method/odata/v1/SchemaRegistry?$filter=Code eq
'{eventCodeExp}' and ModuleCode eq '{moduleCodeExp}'&$top=1&$orderby=UpdatedAt desc
GET /v1.0/invoke/core_schema_registry/method/odata/v1/SchemaRegistry?$filter=Code eq
'{eventCodeImp}' and ModuleCode eq '{moduleCodeImp}'&$top=1&$orderby=UpdatedAt desc
GET /v1.0/invoke/core_dispatcher_relationships/method/odata/v1/StateEntity?$filter=Code
eq '{entityCode}' and ModuleCode eq '{moduleCodeEntity}'
GET /v1.0/invoke/core_dispatcher_relationships/method/odata/v1/StateEntity?$filter=Id eq
'{Id}' and true
GET /v1.0/invoke/core_dispatcher_relationships/method/odata/v1/StateFunction?
$filter=Code eq '{functionCode}' and ModuleCode eq '{moduleCode}'
GET /v1.0/invoke/core_user_control/odata/v1.0/user?%24filter=ExternalId+eq+'{userId}'
GET /v1.0/invoke/core_license/method/api/v1/license/status
GET /v1.0/invoke/core_dispatcher_relationships/method/odata/v1/$metadata
GET /v1.0/invoke/core_schema_registry/method/api/v1/schemaregistry/{Id}/setsendingtime

```

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейно нарастающая нагрузка на микросервис согласно следующему профилю:

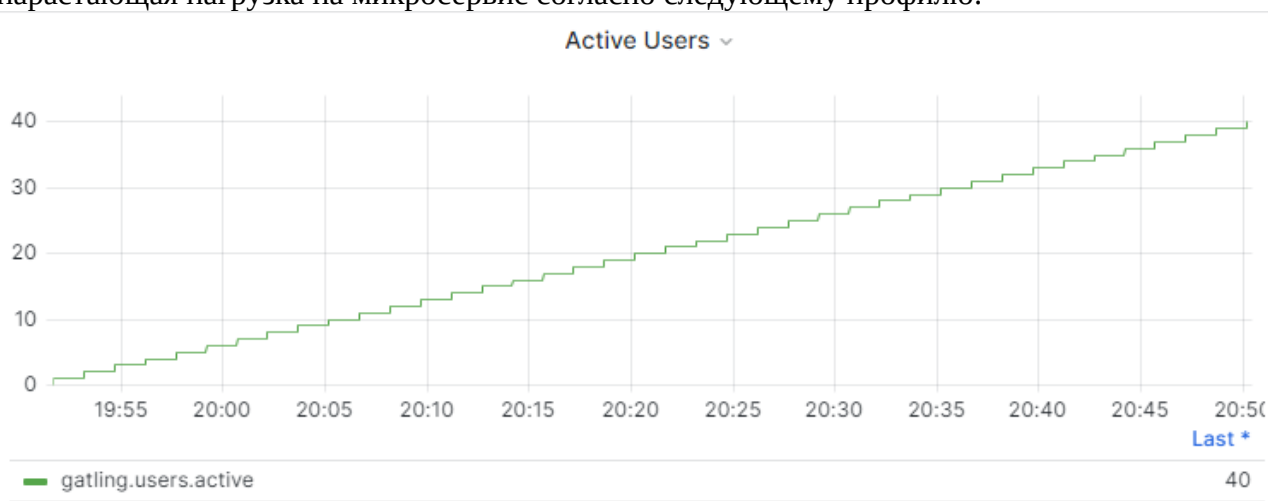


Рис.1 Профиль нагрузки для определения максимальной производительности.

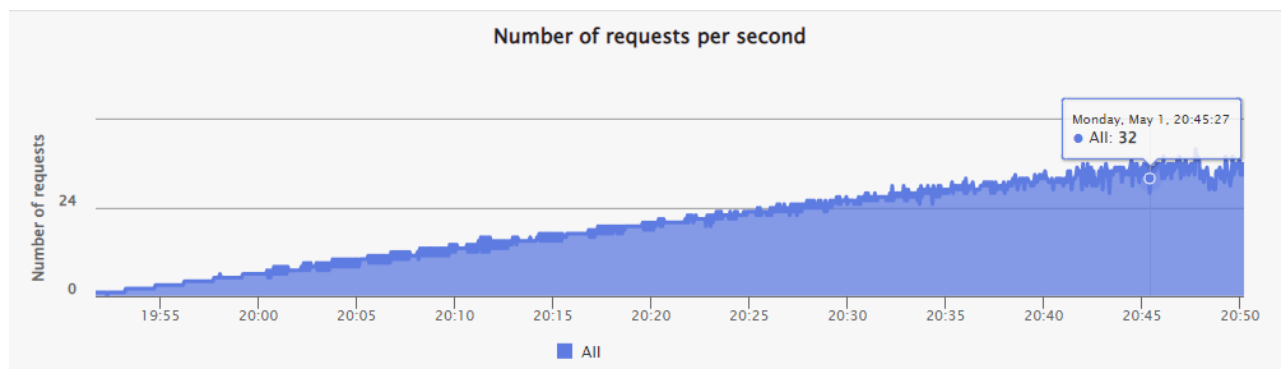


Рис.2. Пропускная способность для растущей нагрузки.

В результате определили максимальную производительность равную 32 запросам в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2

достигнув максимального значения перестали расти и ресурсы CPU и RAM были практически полностью утилизированы.

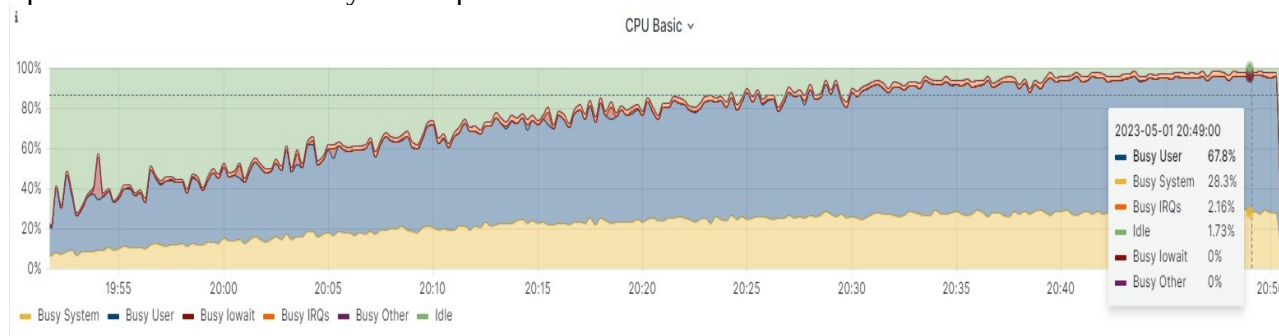


Рис.3. Утилизация CPU на хостовой виртуальной машине.

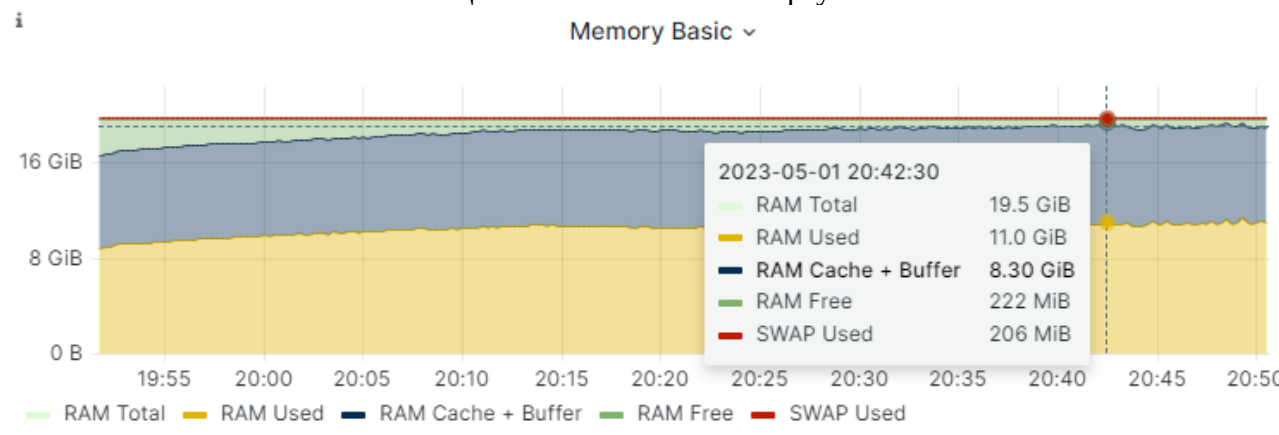


Рис.4. Утилизация RAM на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

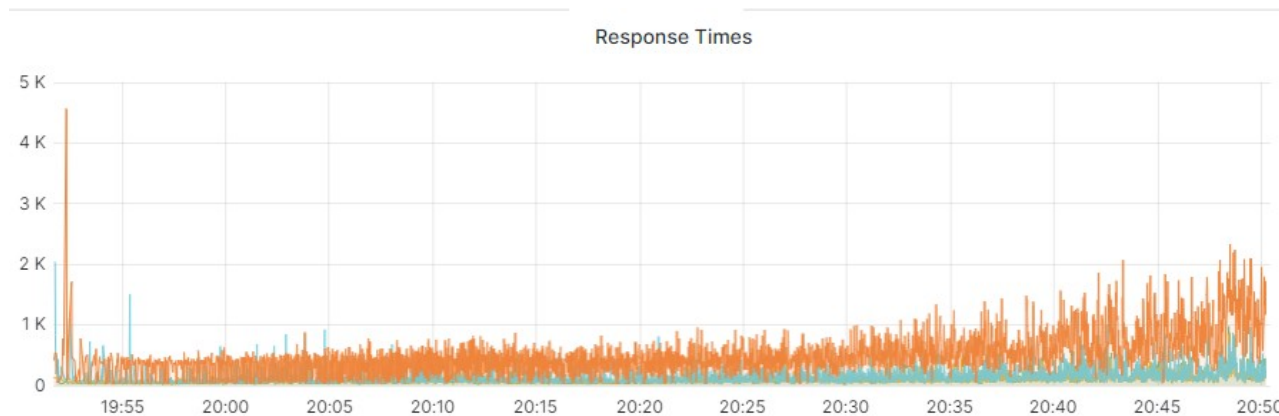


Рис.5. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 22 запроса в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

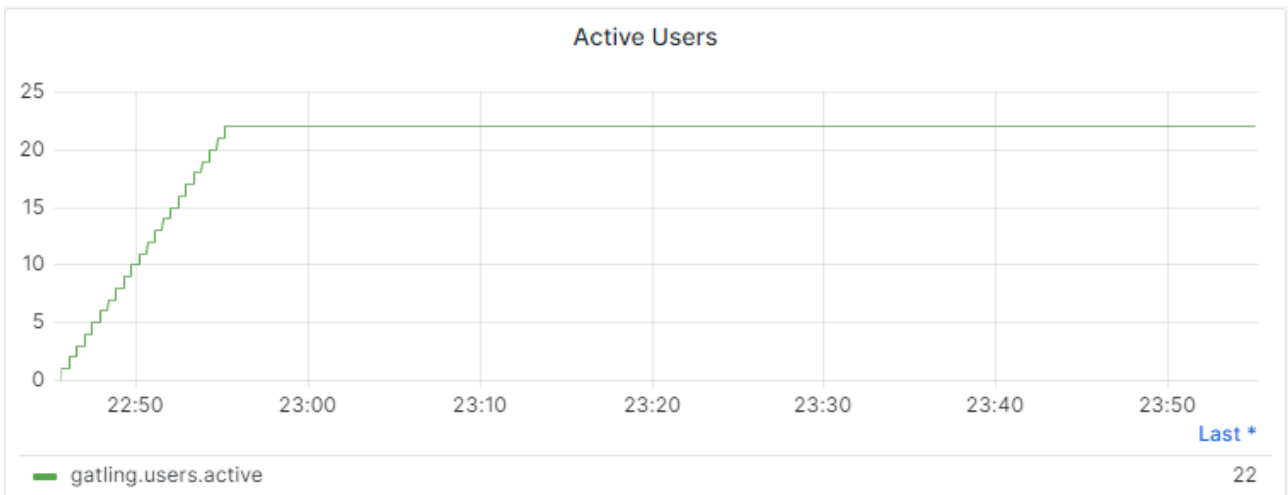


Рис.6. Профиль нагрузки для определения ключевых показателей.

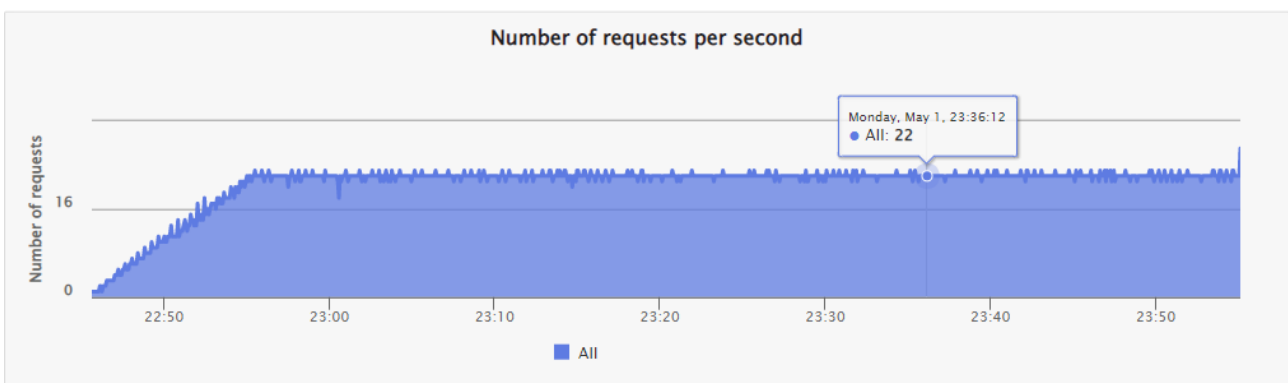


Рис.7. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

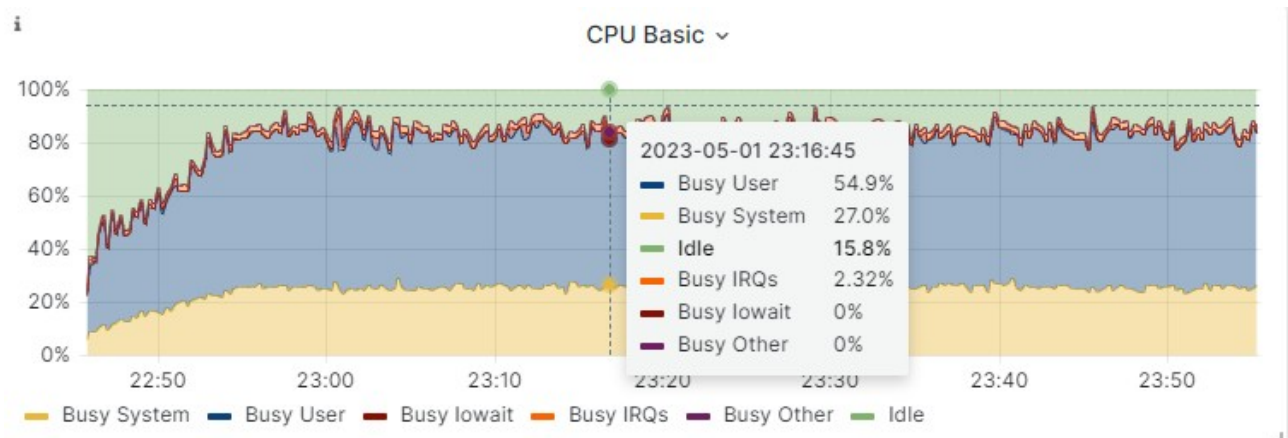


Рис 8. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 84%

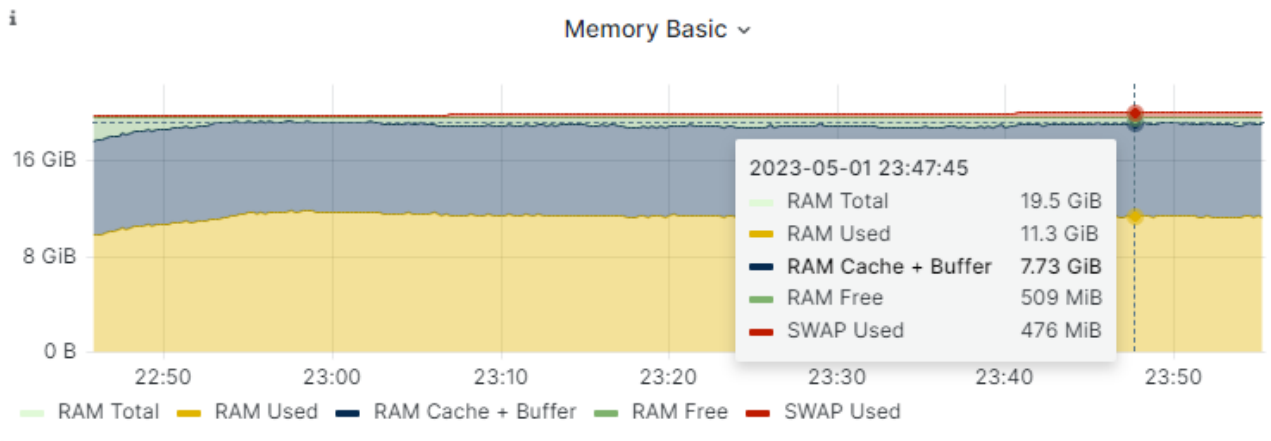


Рис 9. Утилизация оперативной памяти при стабильной нагрузке.

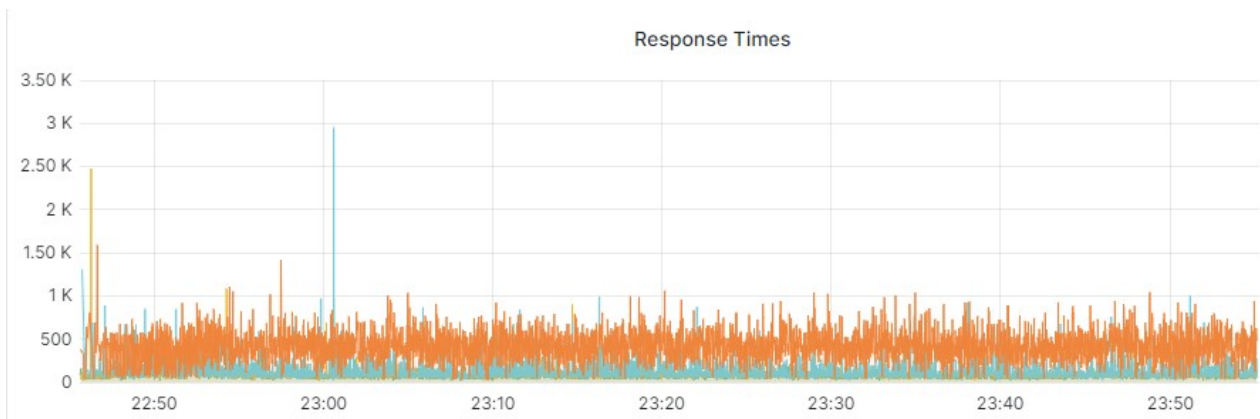


Рис.10. Времена ответа при стабильной нагрузке.

Времена ответов при стабильной нагрузке - стабильны

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST/api/v1/module/{moduleCode}/entity/{entityCode}	25603	13	1721	441	474	640	934
POST /api/v1/module/{moduleCode}/event/{eventCode} /export	21492	10	705	82	62	103	208
POST /api/v1/module/{moduleCode} /event/{eventCode}/import	21235	16	4871	98	79	116	213
POST /api/v1/module/{moduleCode} /function/{functionCode}	16961	13	6262	135	80	129	443

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса EventConsumer

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 21.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Определение соответствия системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

–Количество запросов и событий от модулей - до 30 в секунду

–Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса EventConsumer составила 90 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 80% составляют:

Запрос	95% line, ms
POST /api/v1/eventconsumer/state	391

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис «EventConsumer». В тесте использовался следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
POST /api/v1/eventconsumer/state	100

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в секунду.

6. Использование заглушек:

Для работы микросервис «EventConsumer» взаимодействует с микросервисом «ManagementApi». В ходе тестирования сервис «ManagementApi» был заглушен с использованием утилиты «Wiremock», а именно, следующие запросы:

```
GET /odata/v1/Module?$filter=Code eq '{code}'
```

```
GET /odata/v1/ModuleEvents?$filter=(ModuleId eq null or ModuleId eq '{moduleid'}) and
```

```
Is Active
```

```
GET /odata/v1/ModuleEvents?$filter=(EventId in
```

```
('{eventid1}','{eventid2}','{eventid3}','{eventid4}','{eventid5}','{eventid6}','{eventid7}')) and Is
```

```
Active
```

```
GET/odata/v1/Module?$filter=(Id in
```

```
('{id}','{id2}','{id3}','{id4}','{id5}','{id6}','{id7}','{id8}','{id9}')) and true
```

```
GET /odata/v1/ModuleEvents?$filter=(Id in ('{id}')) and Is Active
```

```
GET /odata/v1/Module?$filter=( Id in ('{id}')) and true
```

```
GET /odata/v1/ModuleEvents?$filter=ModuleId eq '{moduleId}' and IsActive
```

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейно нарастающая нагрузка на микросервис согласно следующему профилю:

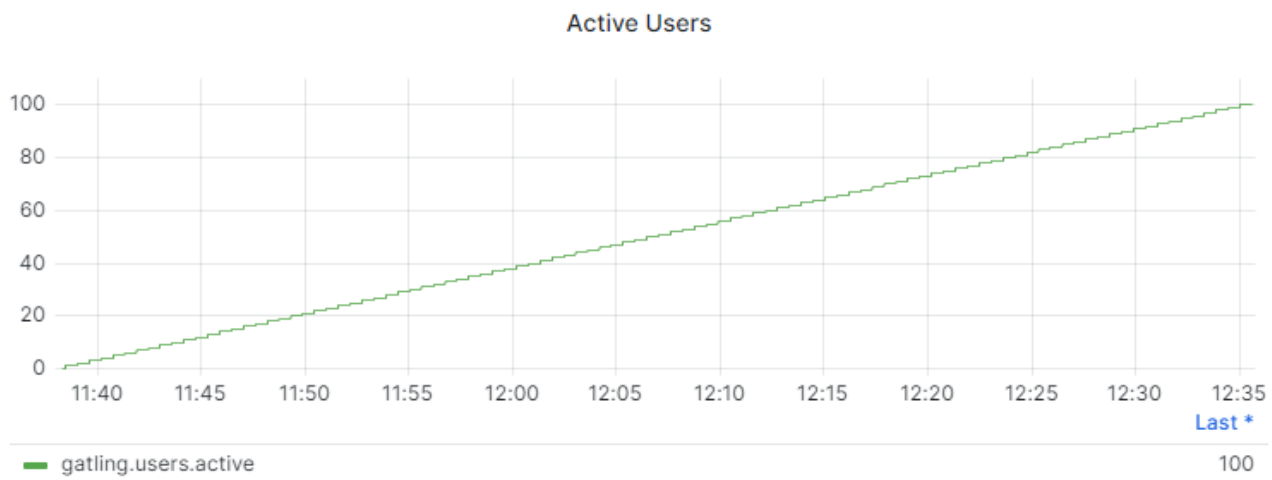


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную 90 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 достигнув максимального значения перестали расти и ресурсы CPU были практически полностью утилизированы.

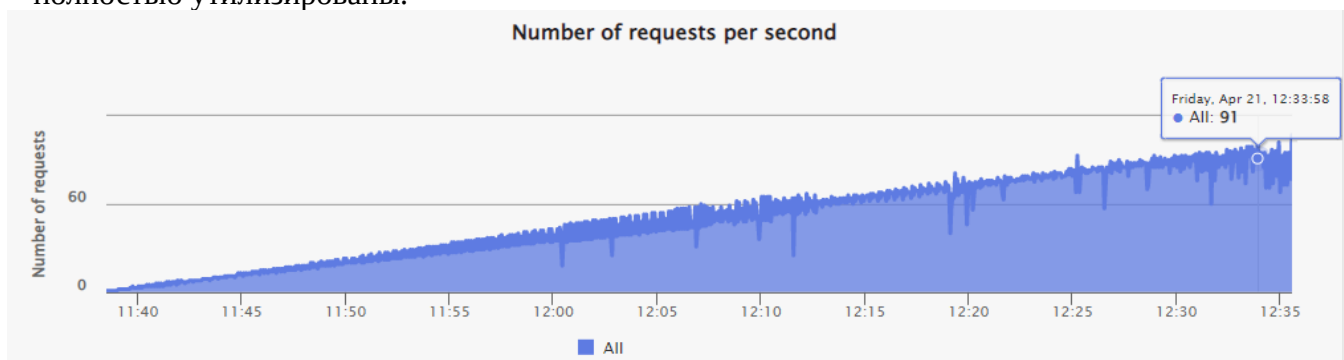


Рис.2. Пропускная способность для растущей нагрузки.

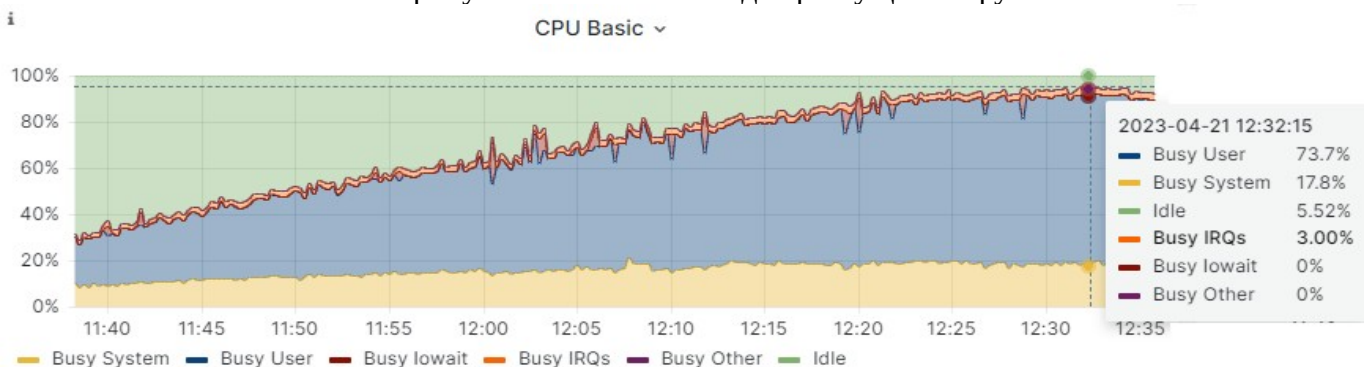


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

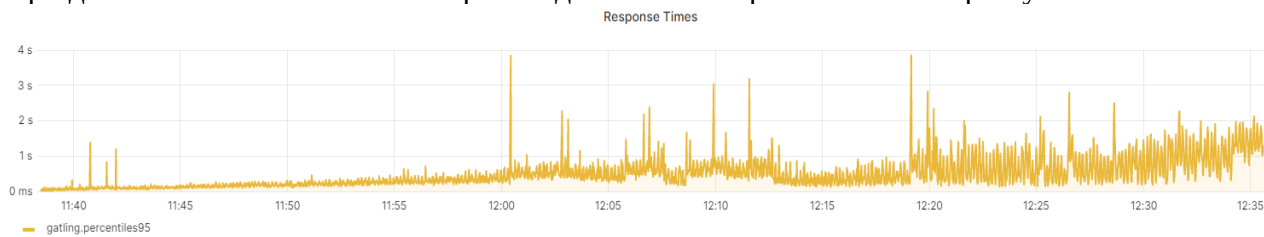


Рис.4. Динамика времен ответа 95% при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 65 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

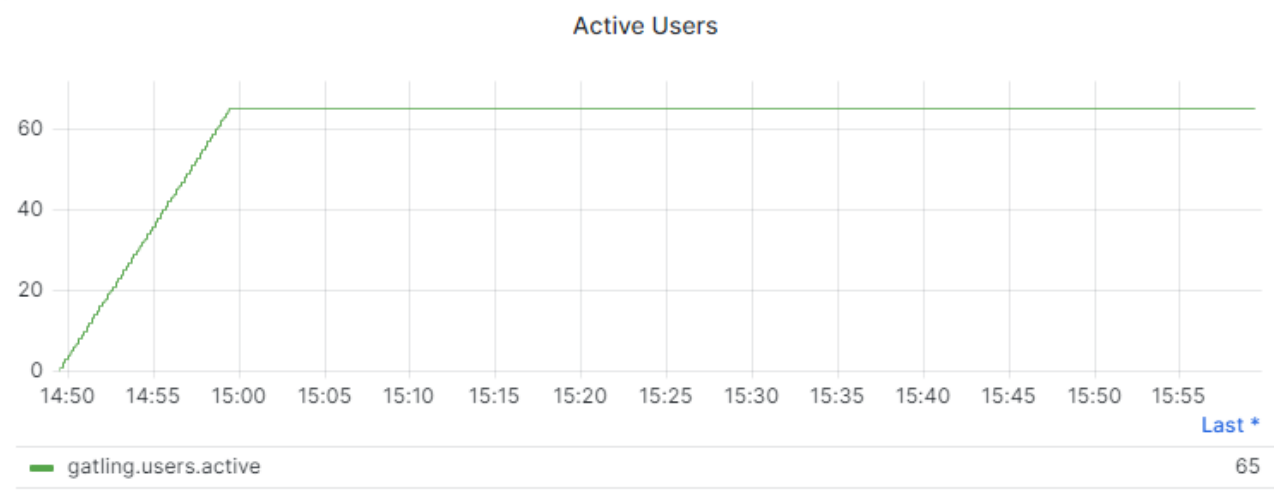


Рис.5. Профиль нагрузки для определения ключевых показателей.

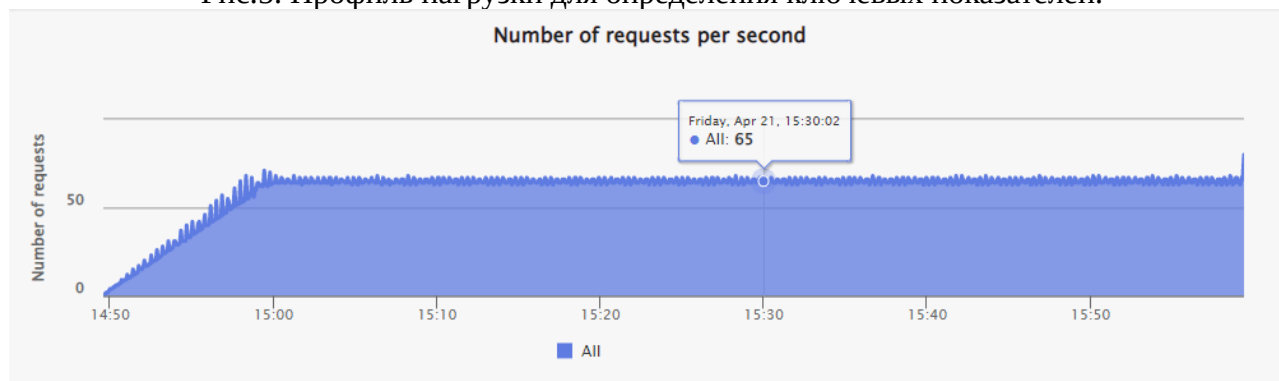


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

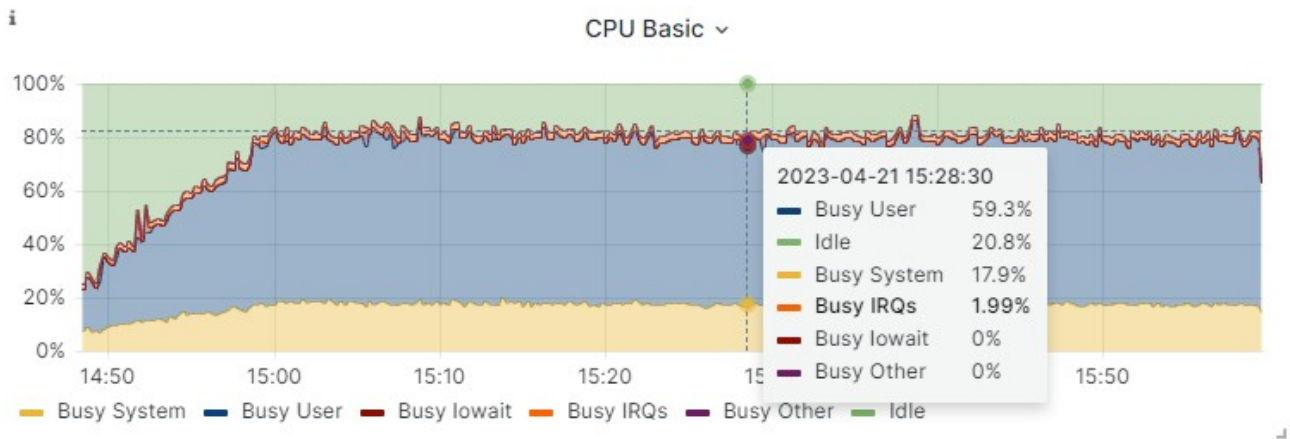


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 80%

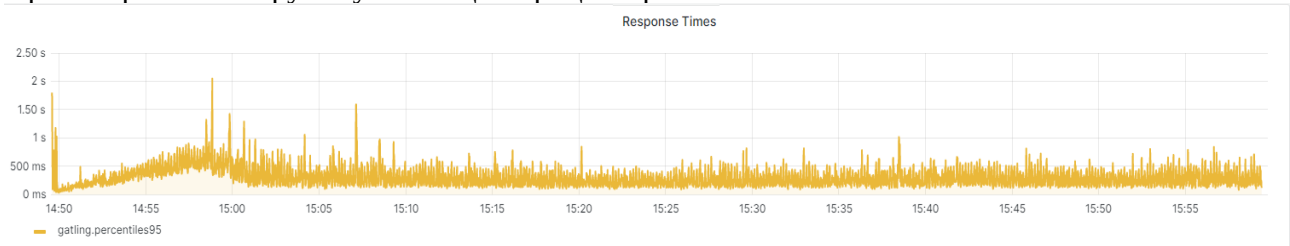


Рис.8. Времена ответа 95% при стабильной нагрузке.

Выброс в начале теста связан с прогревом системы и быстрым разгоном. Далее времена ответа снижаются. В остальном времена ответа стабильны при стабильной нагрузке.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST /api/v1/eventconsumer/state	253192	4	2351	129	85	172	391

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса EventProducer

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 12.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Соответствие системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

– Количество запросов и событий от модулей - до 30 в секунду.

– Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Тесты проводились с разным набором тестовых данных: с одним событием (eventCode) и с 33 событиями. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Ошибки от сервиса отсутствуют.

Тесты	MaxPerf, rps	95% line, ms
С одним eventCode	3700	492
С 33мя eventCode	2200	883

Таблица.1. Сравнения показателей производительности для разного набора тестовых данных

На ключевые показатели производительности влияет количество тестовых данных. При увеличении количества видов событий передаваемых сервисом, значения максимальной производительности уменьшаются, а времена ответов увеличиваются. При проведении тестов со стабильной нагрузкой для разного набора тестовых данных получили, что при более высоком уровне нагрузке в 2200 rps для одного события - времена ответов ниже чем при уровне нагрузки в 1500 rps для 33 видов событий. Тем не менее сервис удовлетворяет заявленным требованиям в 30 запросов в секунду от одного модуля.

Во время тестов при отправке запросов на <http://inka-lt.inka.konsom.ru> (на nginx реверс прокси) наблюдались ошибки 502 и 404. В связи с этим, требуется проведение конфигурационных тестов для увеличения пропускной способности реверс-прокси.

5. Модель нагрузки:

Тестировался микросервис "EventProducer". Запрос отправлялся непосредственно в микросервис, минуя nginx. В тесте использовался следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
POST /api/v1/eventproducer/message	100

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в секунду.

Для выявления зависимости производительности от количества видов событий обрабатываемых сервисом проводились тесты с разным количеством тестовых данных: с одним eventCode и с тридцатью тремя eventCode.

6. Ход нагрузки:

Тест на определение максимальной производительности с одним eventCode.

Для определения максимальной производительности подавалась линейная нагрузка на микросервис, согласно следующему профилю:

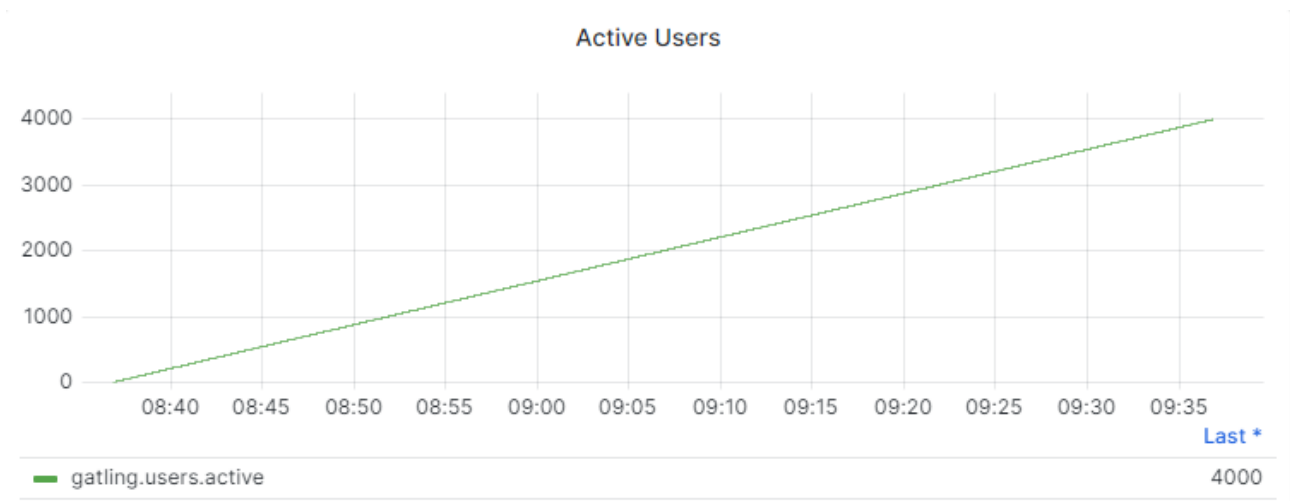


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную около 3700 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 вышли на насыщение и ресурсы CPU были практически полностью утилизированы.

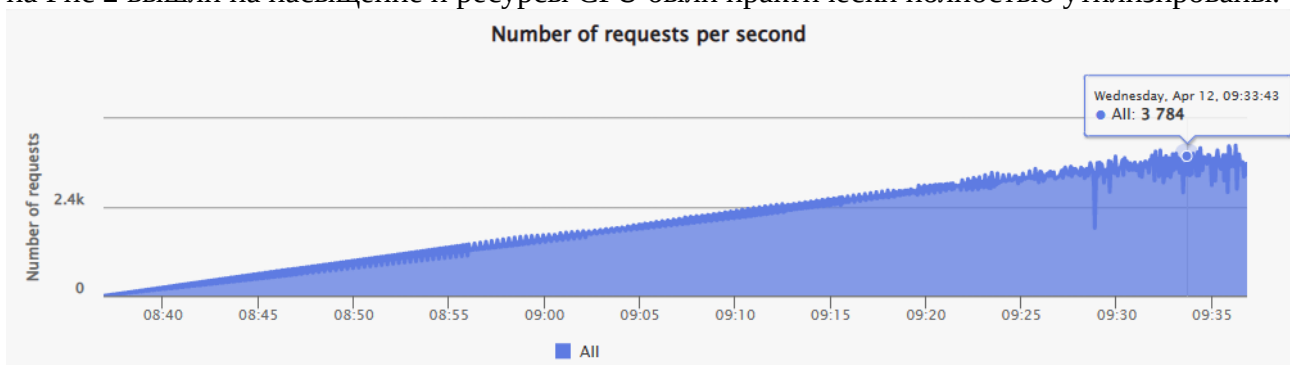


Рис.2. Пропускная способность для растущей нагрузки.

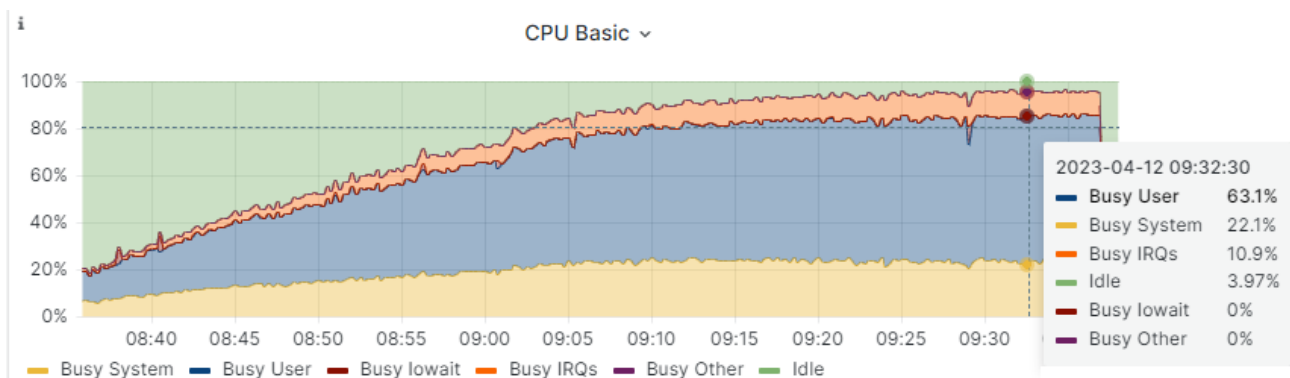


Рис.3. Утилизация CPU на хостовой виртуальной машине.

Работу сервиса можно проконтролировать через мониторинг публикации микросервисом событий в кафке.

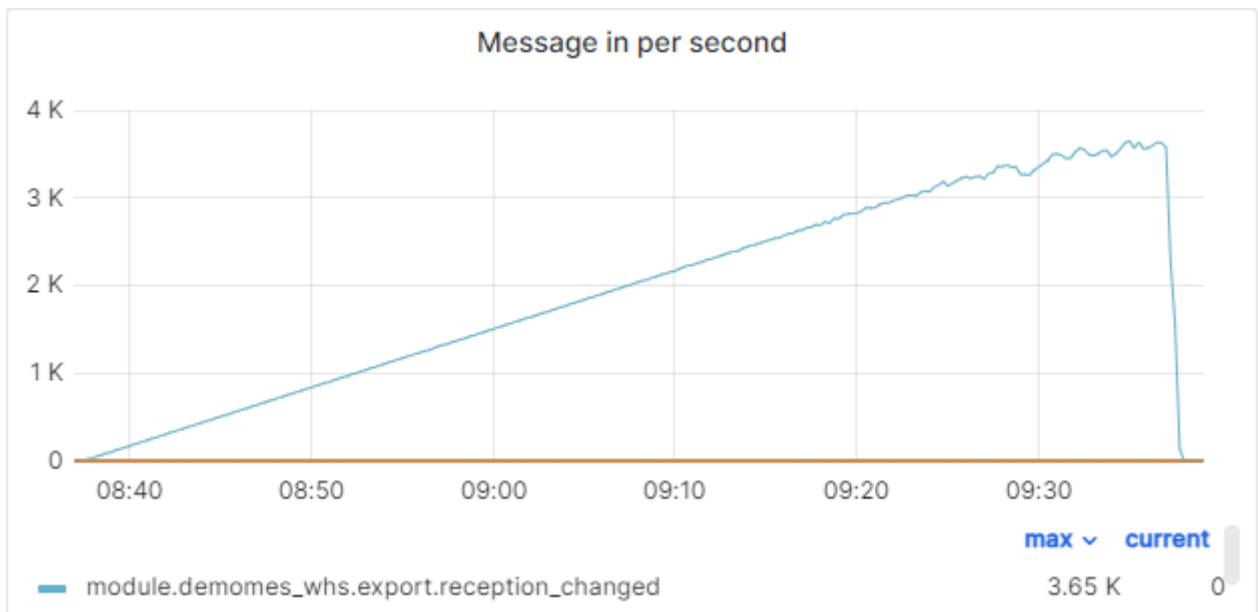


Рис.4. Публикация в единицу времени при отправке сообщений в один топик.

При достижении максимальной производительности сервиса график выходит на насыщение и фиксирует значение пропускной способности.

Динамика времен ответа для сервиса при растущей нагрузке показывает рост значений при достижении максимальной производительности.

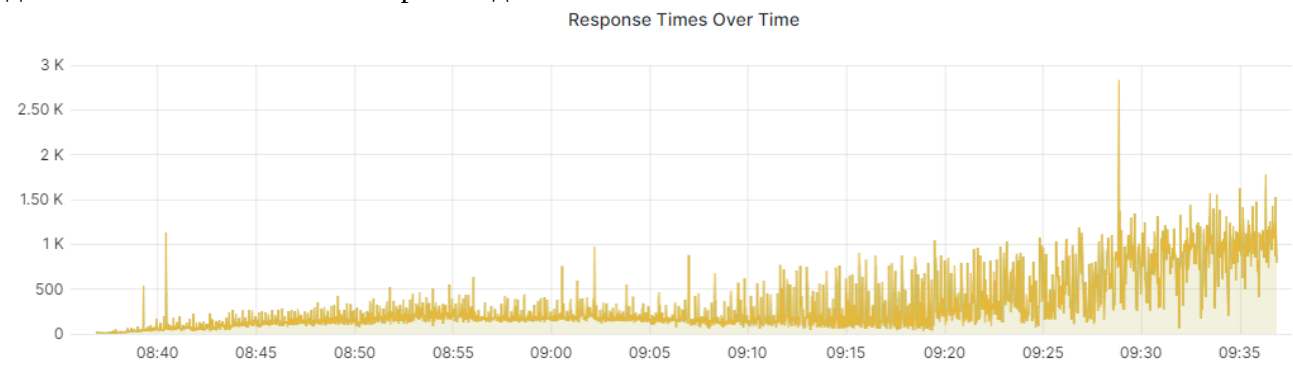


Рис.5. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей с 1 eventCode.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 2200 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

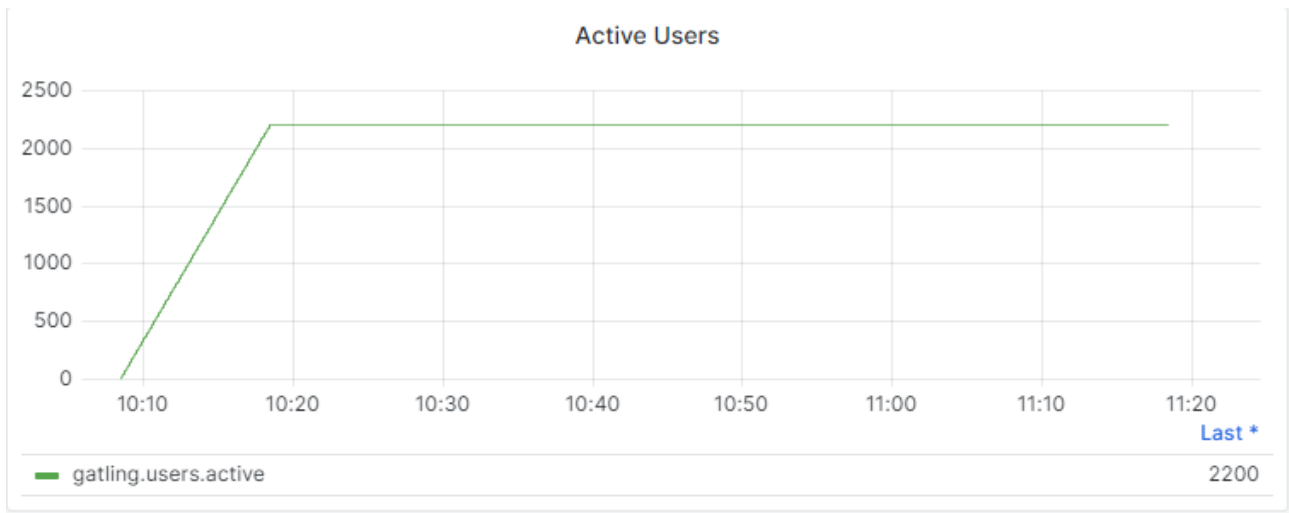


Рис.6 Профиль нагрузки для определения ключевых показателей.

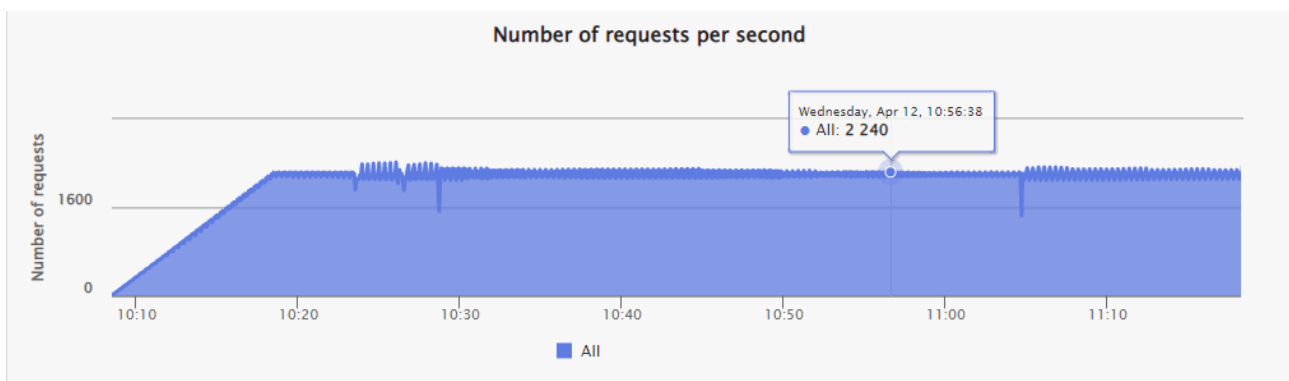


Рис.7. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

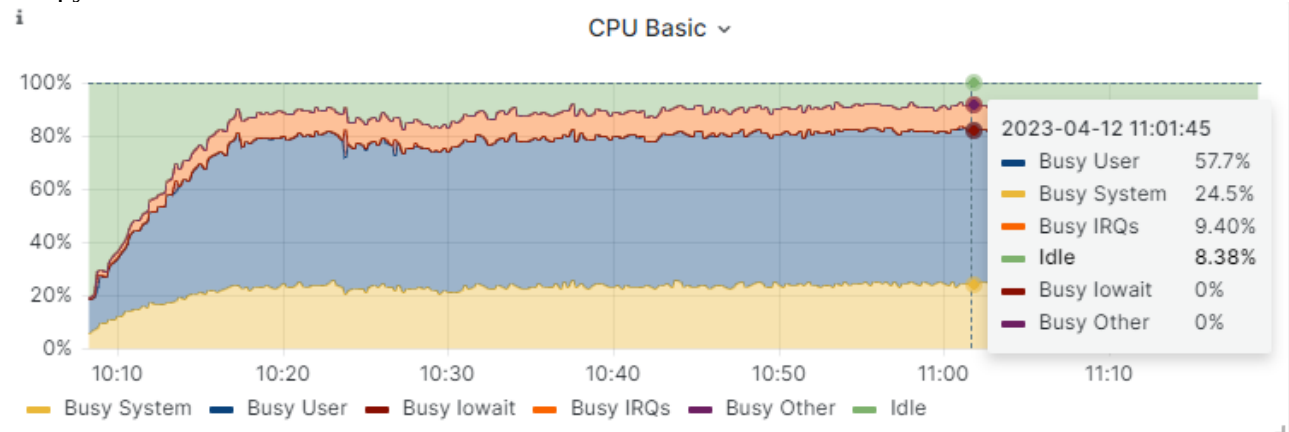


Рис 8. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 90%

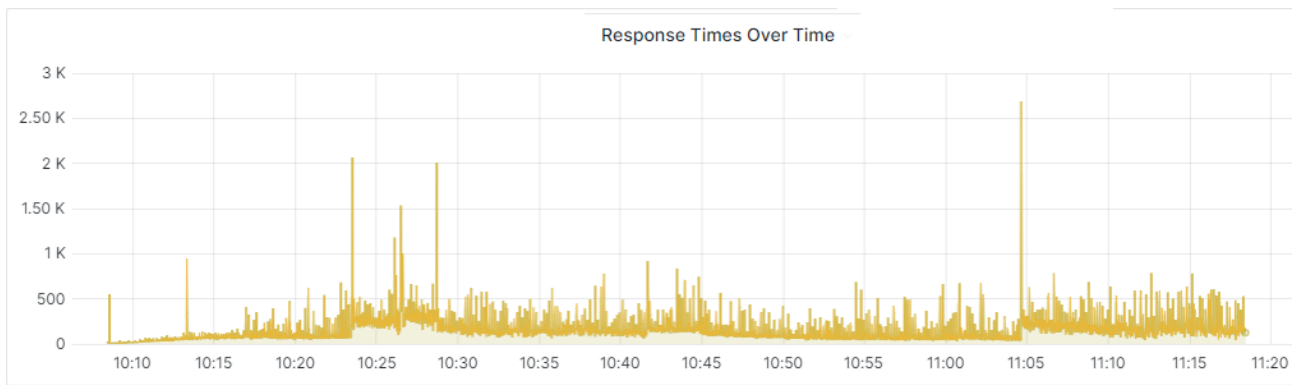


Рис.9. Времена ответа при стабильной нагрузке.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST /api/v1/eventproducer/message	8568066	1	2961	173	129	242	492

Таблица 3. Времена ответа при стабильной нагрузке.

Тест на определение максимальной производительности с 33 eventCode.

Для определения максимальной производительности подавалась линейная нагрузка на микросервис, согласно следующему профилю:

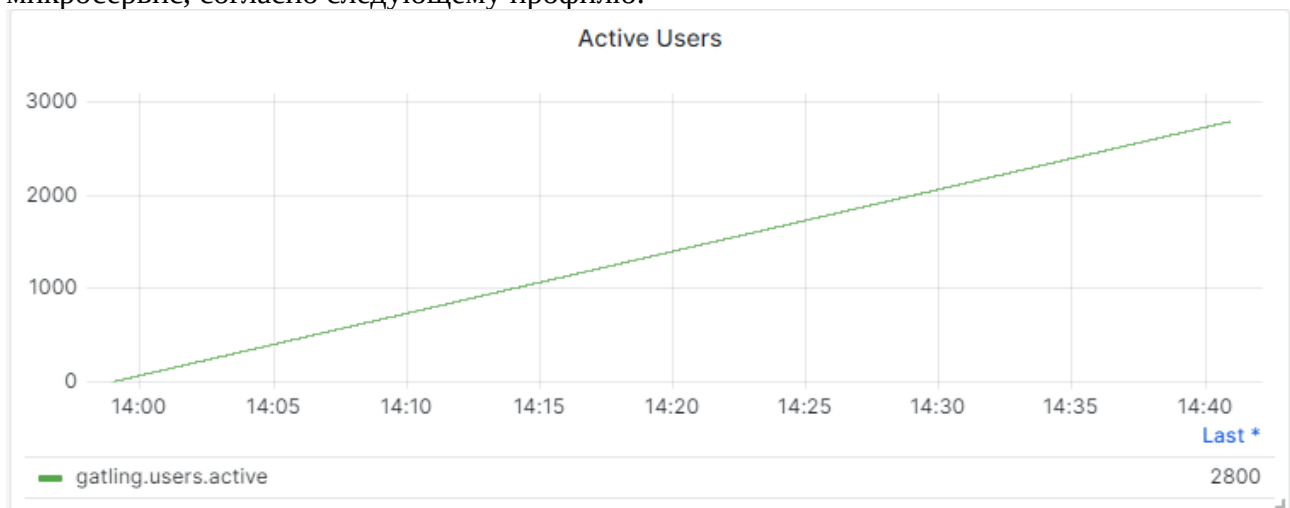


Рис.10. Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную около 2200 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 вышли на насыщение и ресурсы CPU были практически полностью утилизированы.

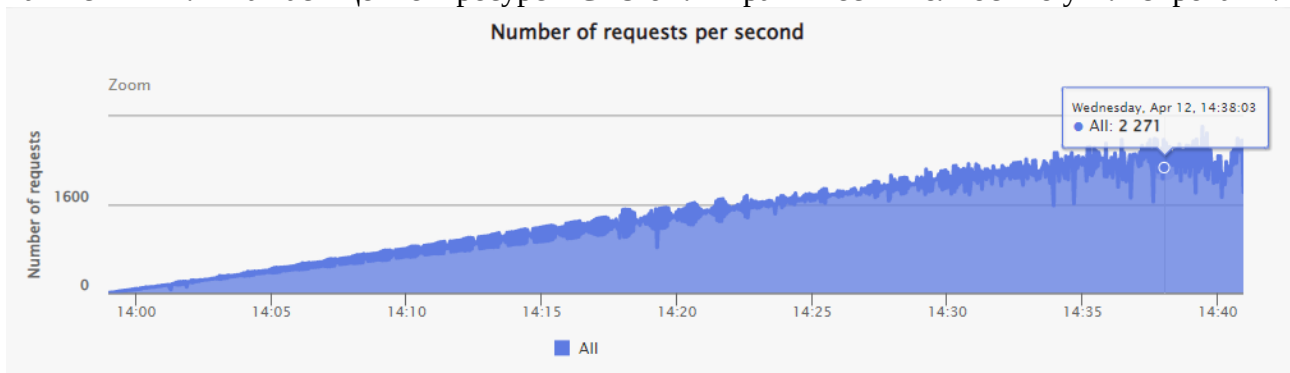


Рис.11. Пропускная способность для растущей нагрузки.

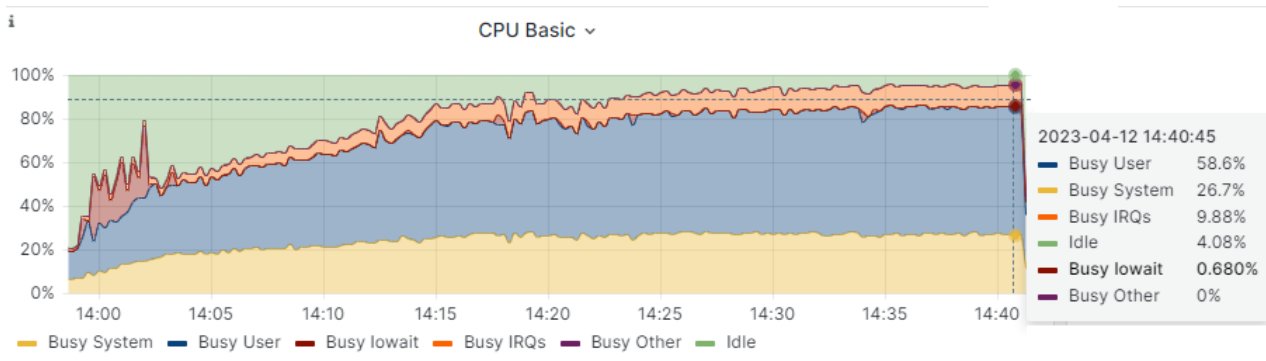


Рис.12. Утилизация CPU на хостовой виртуальной машине.

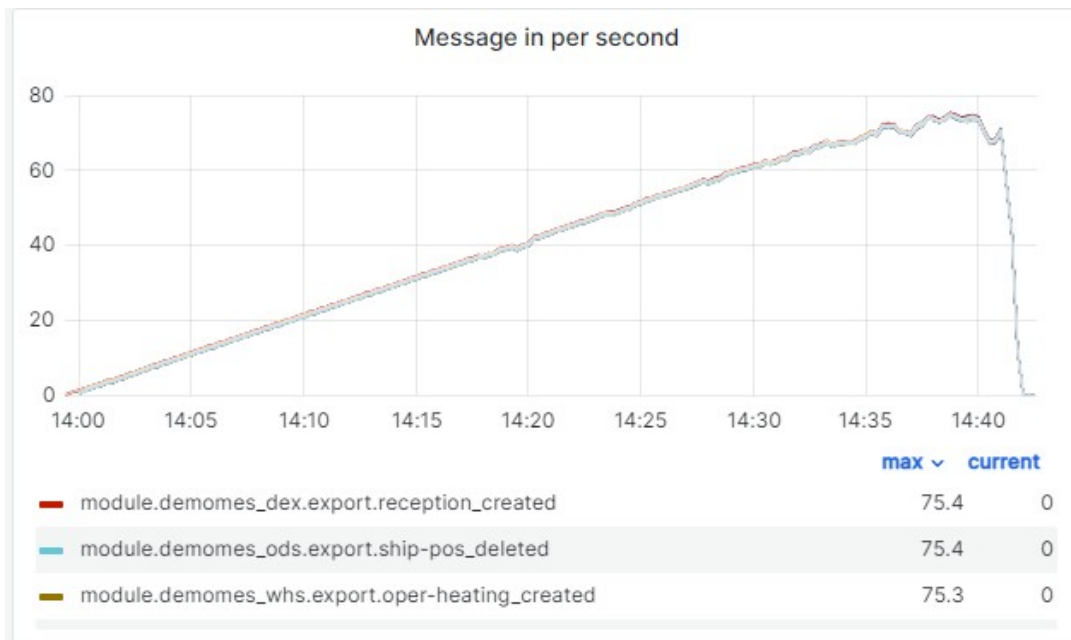


Рис.13. Публикация в единицу времени при отправке сообщений в 33 топика.

Количество публикуемых в кафку сообщений распределяется равномерно по всем топикам. Можно наблюдать, что при достижении максимальной производительности сервиса количество сообщений перестает расти.

Динамика времен ответа для сервиса при растущей нагрузке показывает рост значений при достижении максимальной производительности.

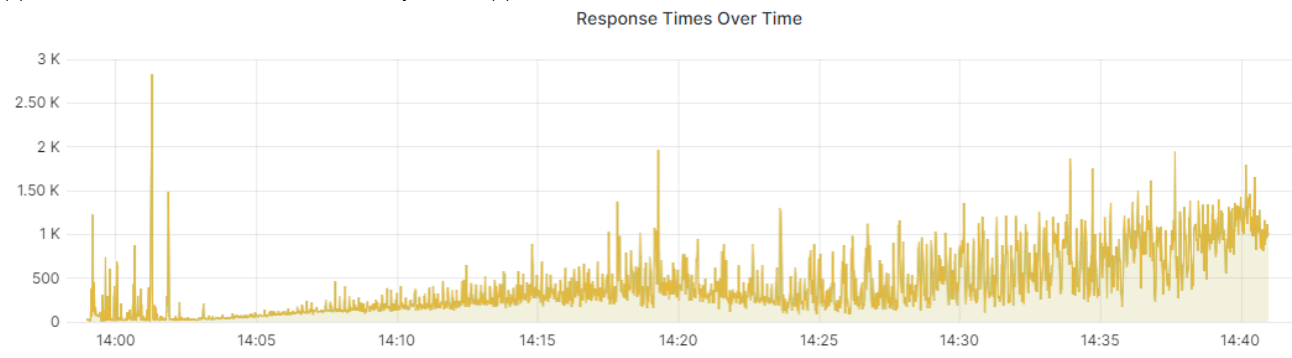


Рис.14. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей с 33 eventCode.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 1500 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

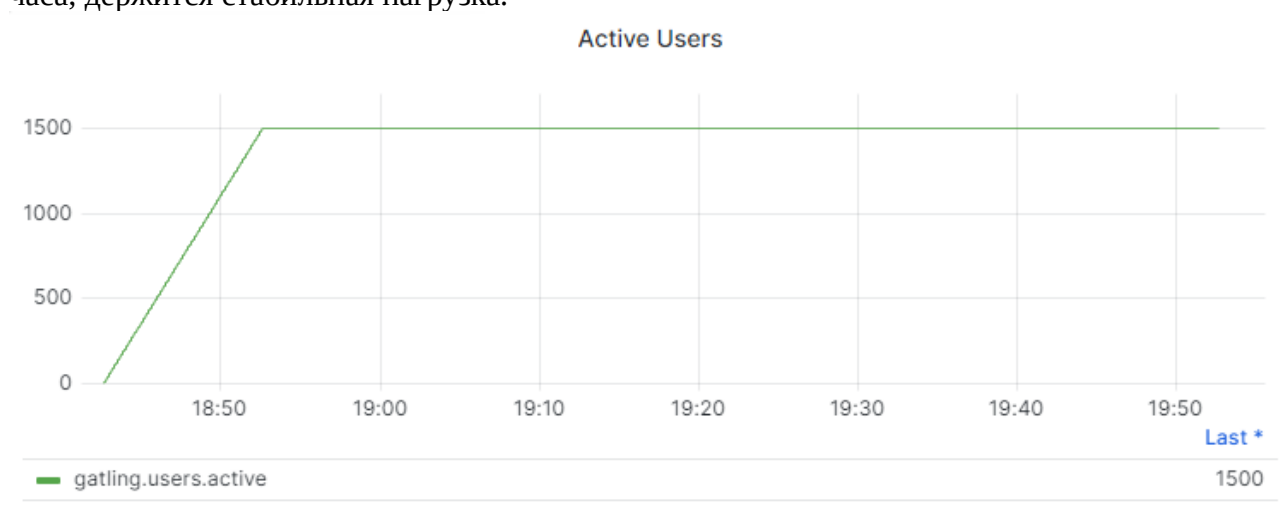


Рис.15 Профиль нагрузки для определения ключевых показателей.

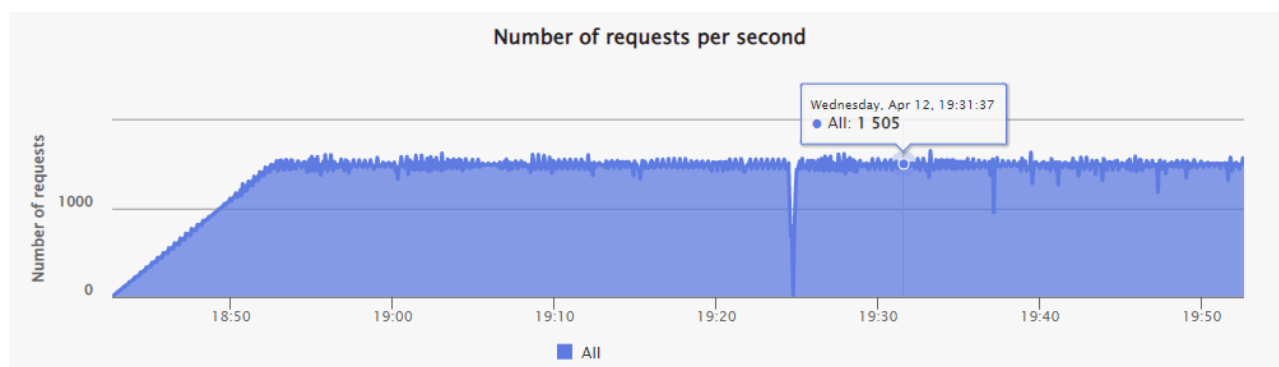


Рис.16. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

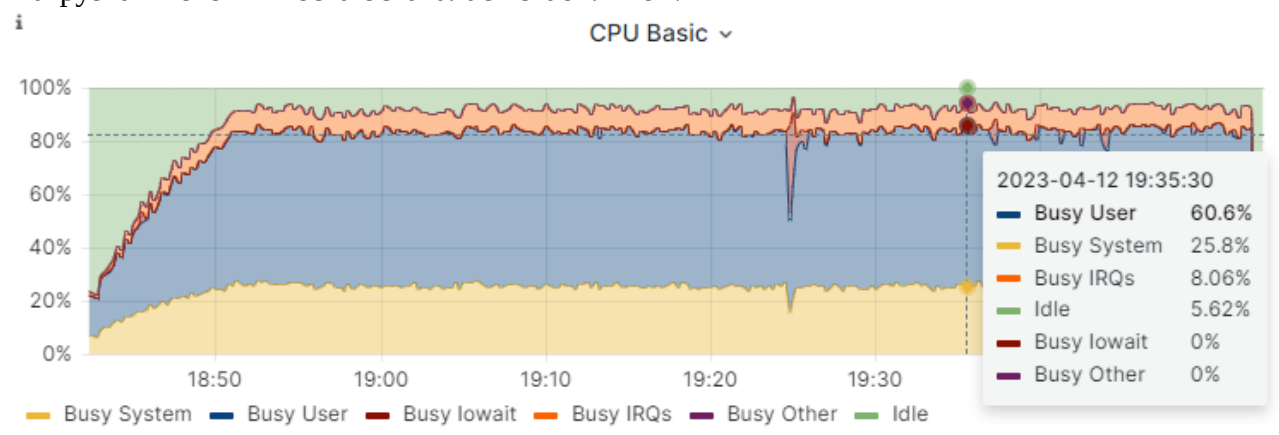


Рис. 17. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 93%

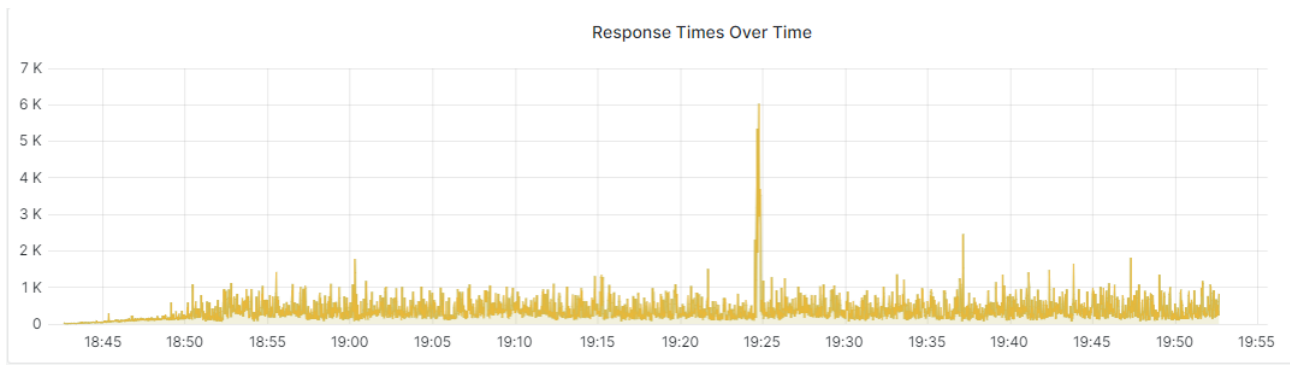


Рис.18. Времена ответа при стабильной нагрузке.

Времена ответа стабильны на всем протяжении теста.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST /api/v1/eventproducer/message	5801449	2	6259	385	325	527	883

Таблица 4. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО
ТЕСТИРОВАНИЯ
микросервиса
ManagementApi

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 27.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Определение соответствия системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

1.4. Сравнение времен ответа odata запросов с фильтрацией и без.

2. Требования:

–Количество запросов и событий от модулей - до 30 в секунду

–Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса ManagementApi составила 75 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответов для запросов odata с фильтрацией меньше чем в запросах без фильтрации и эта разница растет по мере увеличения количества объектов выдаваемых в запросах без фильтров. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 92% составляют:

Запрос	95% line, ms
GET /odata/v1/entity	1655
GET /odata/v1/entity?\$filter=Id eq '{moduleEntity}'	549
GET /odata/v1/module	535
GET /odata/v1/module?\$filter=Code eq '{moduleCode}'	521
GET /odata/v1/moduleevents	953
GET /odata/v1/moduleevents?\$filter=Id eq '{moduleEvents}'	547
POST /api/v1/module/testFunction/{testfunctionId}/trigger	640
PUT /api/v1/module/testFunction/executionlog	522

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис «ManagementApi». В тесте использовался следующие запросы:

Нагружаемые эндпоинты микросервиса:	%%
GET /odata/v1/entity	10
GET /odata/v1/entity?\$filter=Id eq '{moduleEntity}'	10
GET /odata/v1/module	10
GET /odata/v1/module?\$filter=Code eq '{moduleCode}'	10
GET /odata/v1/moduleevents	10
GET /odata/v1/moduleevents?\$filter=Id eq '{moduleEvents}'	10
POST /api/v1/module/testFunction/{testfunctionId}/trigger	20
PUT /api/v1/module/testFunction/executionlog	20

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в секунду.

6. Использование заглушек:

Для работы микросервис «ManagementApi» взаимодействует с микросервисом «DispatcherApi». В ходе тестирования сервис «DispatcherApi» был заглушен с использованием утилиты «Wiremock», а именно, следующий запрос:

```
POST /v1.0/invoke/core_dispatcher/method/api/v1/module/testfunction/{testfunctionId}/log/{logId}
```

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейно нарастающая нагрузка на микросервис согласно следующему профилю:

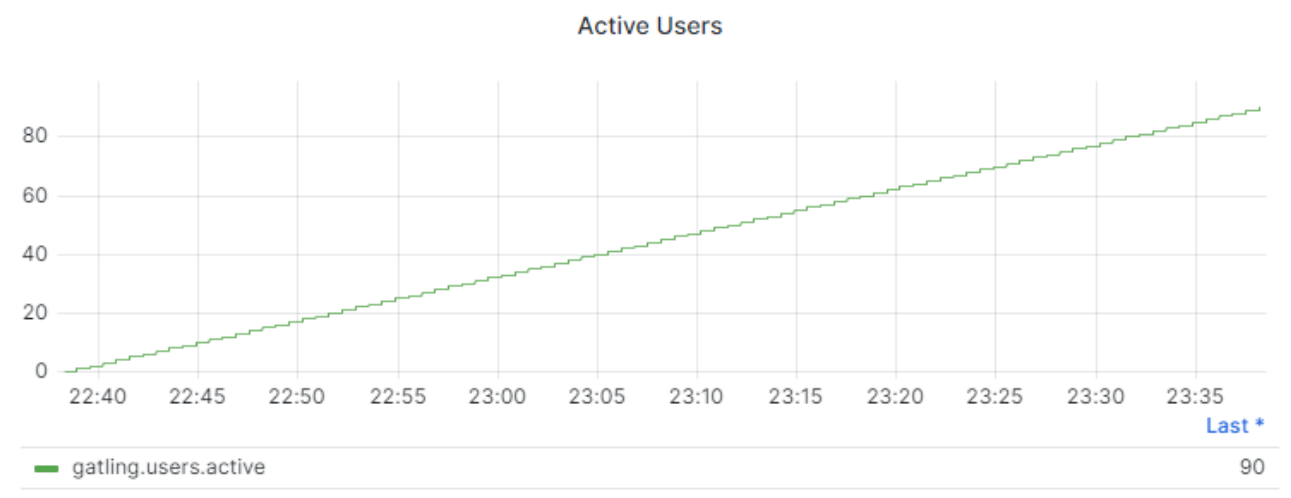


Рис.1 Профиль нагрузки для определения максимальной производительности.

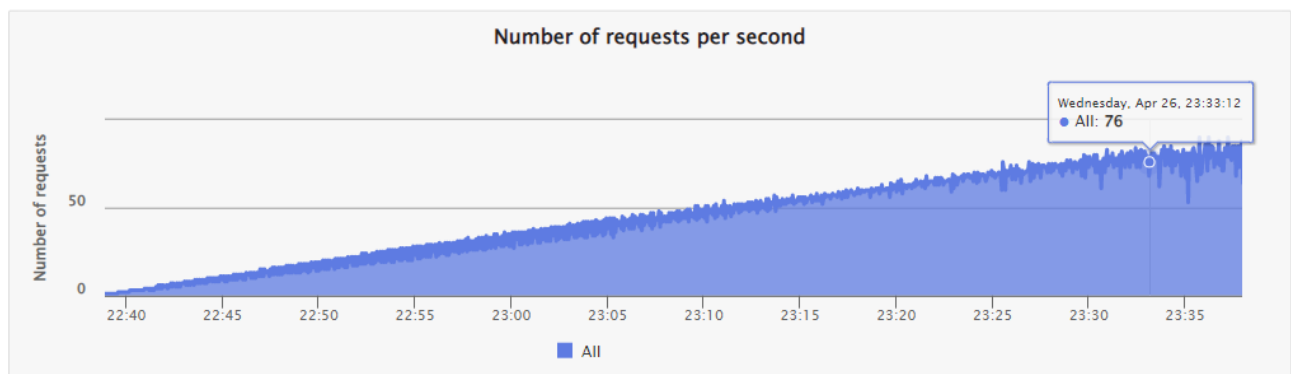


Рис.2. Пропускная способность для растущей нагрузки.

В результате определили максимальную производительность равную 75 запросам в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 достигнув максимального значения перестали расти и ресурсы CPU были практически полностью утилизированы.

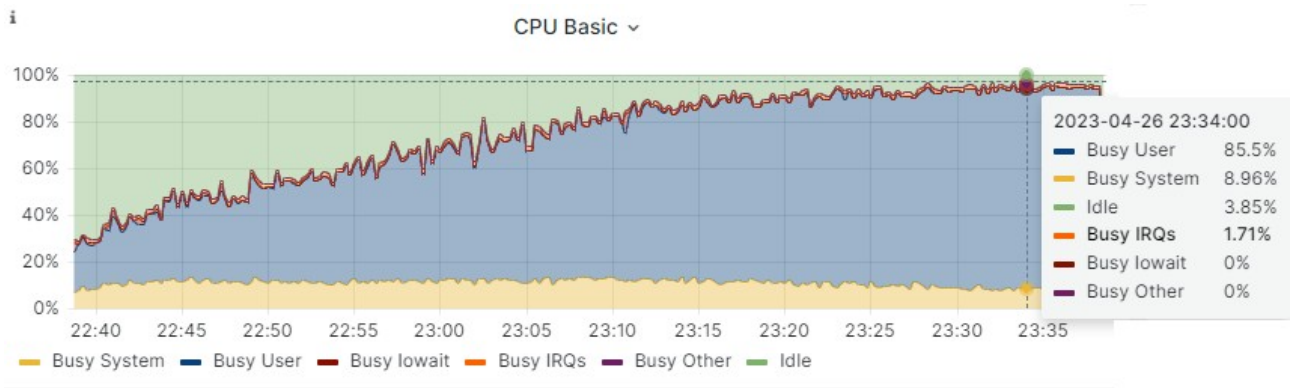


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

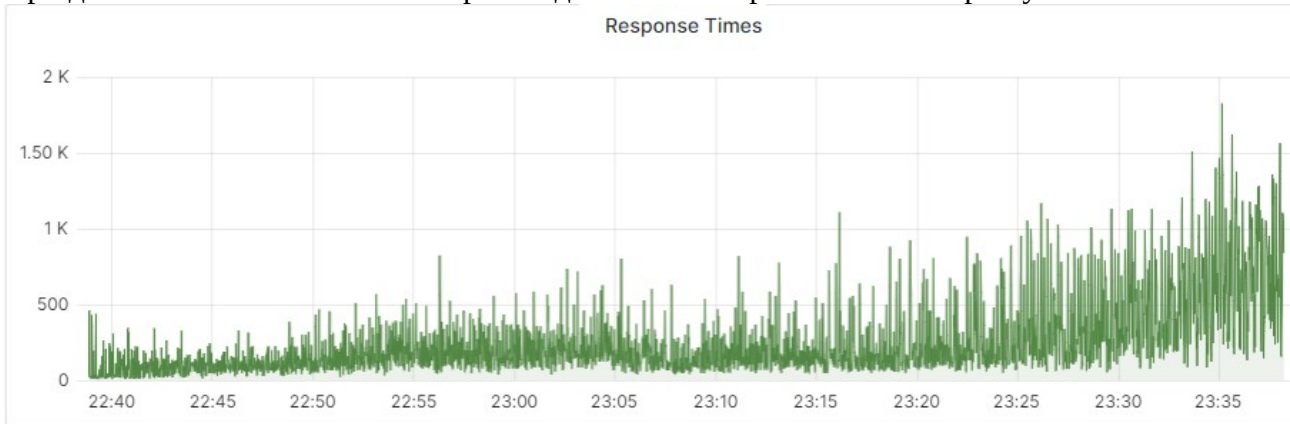


Рис.4. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 62 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

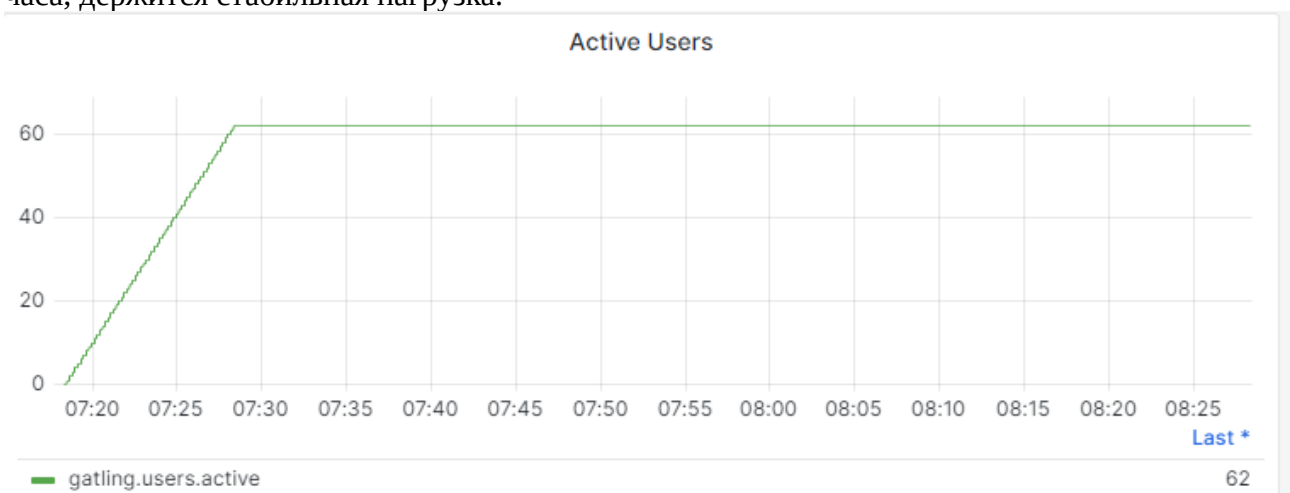


Рис.5. Профиль нагрузки для определения ключевых показателей.

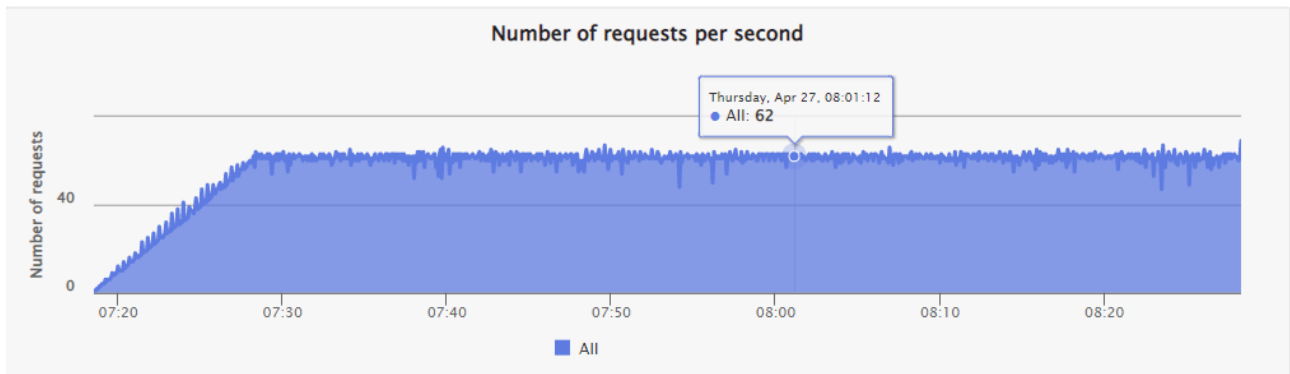


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

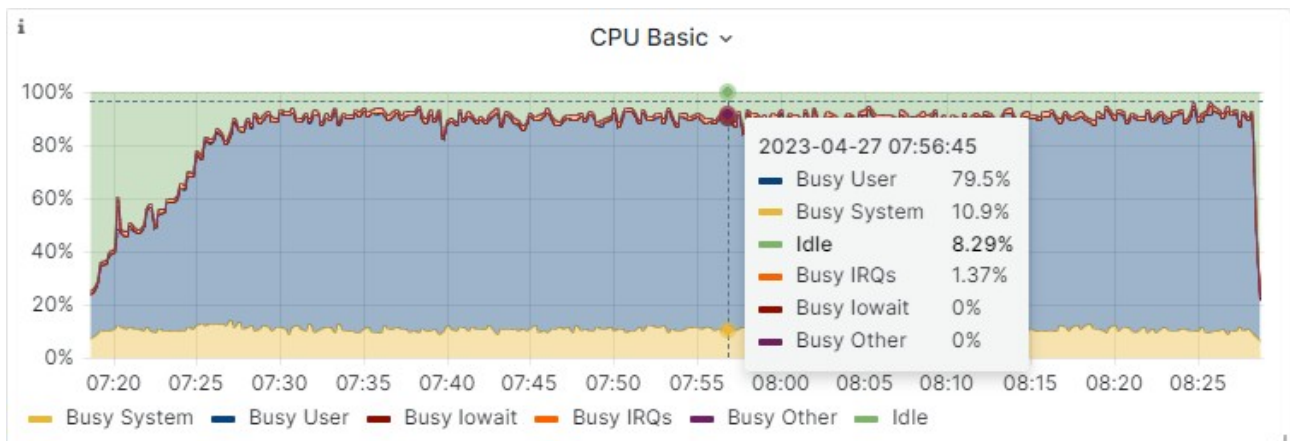


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 92%

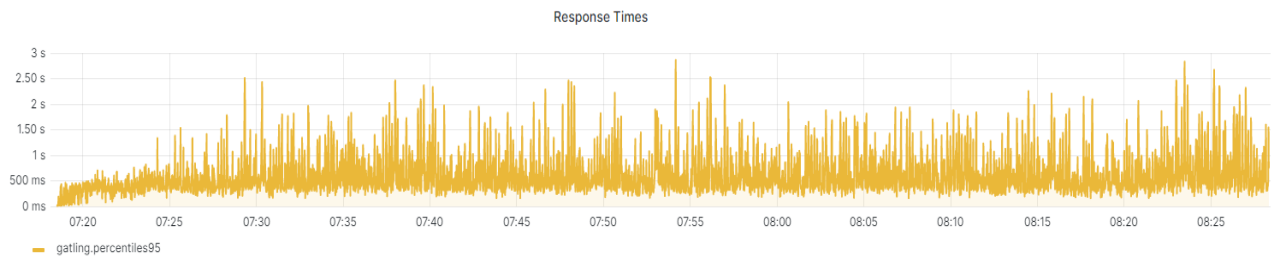


Рис.8. Времена ответа 95% при стабильной нагрузке.

Сравним запросы odata на примере запросов к сущностям модулей GET odata/v1/entity. Поведение остальных запросов аналогичное

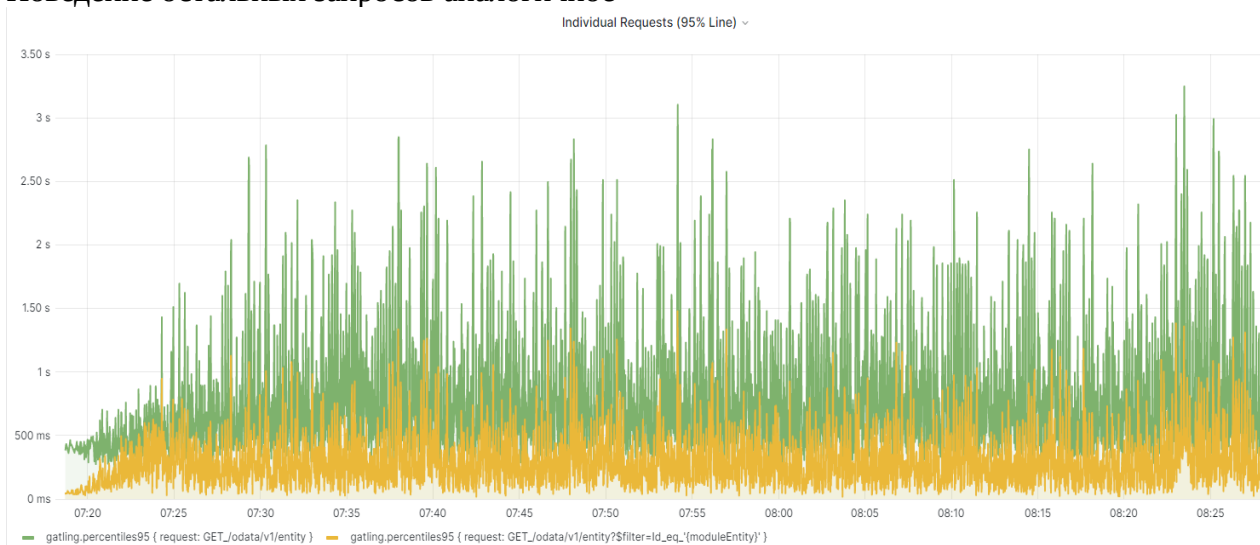


Рис.9. Сравнение времен ответа odata запроса с фильтрацией и без.

Ожидаемо, что времена ответа запроса с фильтрацией меньше и с увеличением количества объектов отдаваемых в запросе без фильтра эта разница будет расти.

В связи с тем что в тестируемой системе содержится небольшое количество модулей, разница времен ответа для запроса GET /odata/v1/module не большая. С количественными показателями сравнения можно ознакомиться ниже в таблице 3

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
GET /odata/v1/entity	23802	151	3235	681	532	838	1655
GET /odata/v1/entity?\$filter=Id eq '{moduleEntity}'	24045	15	1475	188	129	264	549
GET /odata/v1/module	23623	12	1418	181	122	258	535
GET /odata/v1/module?\$filter=Code eq '{moduleCode}'	23802	8	1595	169	109	243	521
GET /odata/v1/moduleevents	24064	73	2224	398	315	504	953
GET /odata/v1/moduleevents?\$filter=Id eq '{moduleEvents}'	23787	10	1592	183	124	256	547
POST /api/v1/module/testFunction/{testfunctionId}/trigger	48294	7	2152	201	130	290	640
PUT /api/v1/module/testFunction/executionlog	47703	2	1686	157	95	233	522

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса MappingRegistry

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 07.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Соответствие системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

- Количество запросов и событий от модулей - до 30 в секунду
- Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса MappingRegistry составила 280 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 90% составляют:

Запрос	95% line, ms
GET /odata/v1/mappingregistry	354

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис “MappingRegistry”. В тесте использовался следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
GET /odata/v1/mappingregistry	100

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 5 запросов за 2 секунды.

6. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейная нагрузка на микросервис согласно следующему профилю:

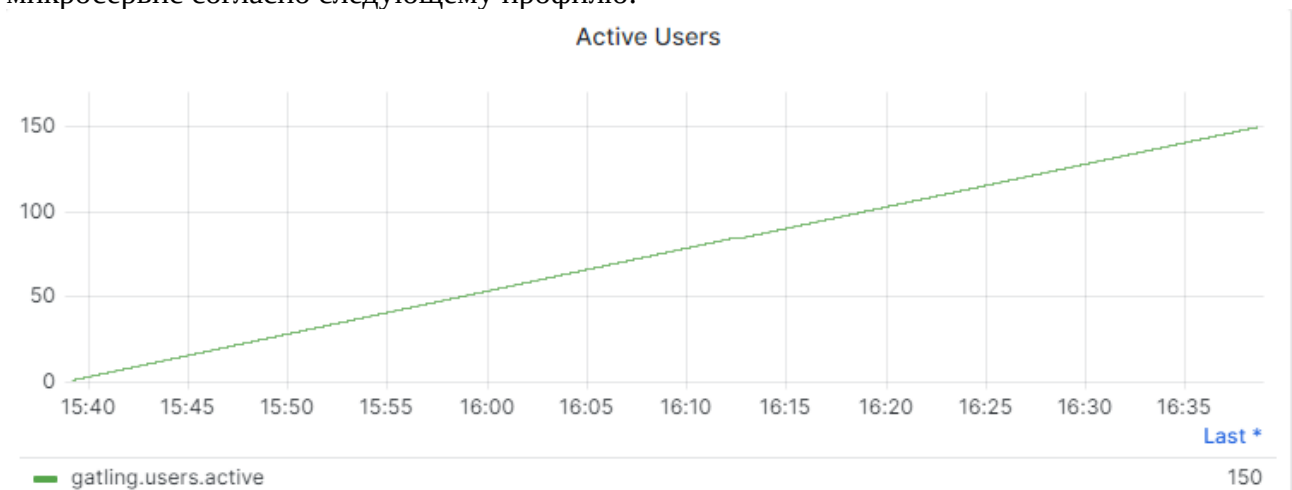


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную 280 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 вышли на насыщение и ресурсы CPU были практически полностью утилизированы.

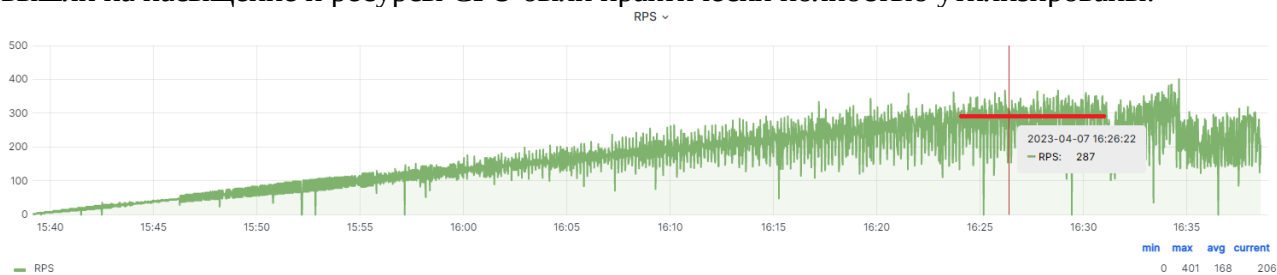


Рис.2. Пропускная способность для растущей нагрузки.

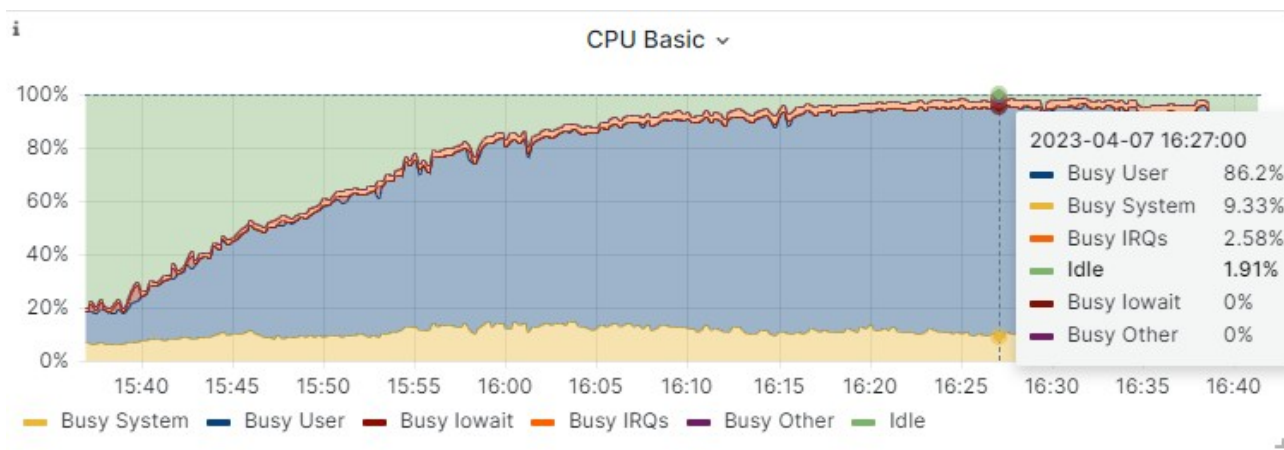


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

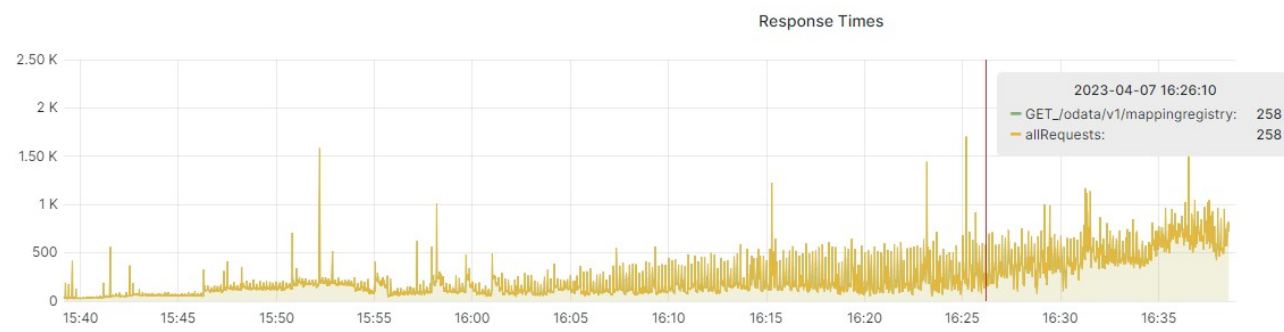


Рис.4. Динамика времен ответа 95% при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 190 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

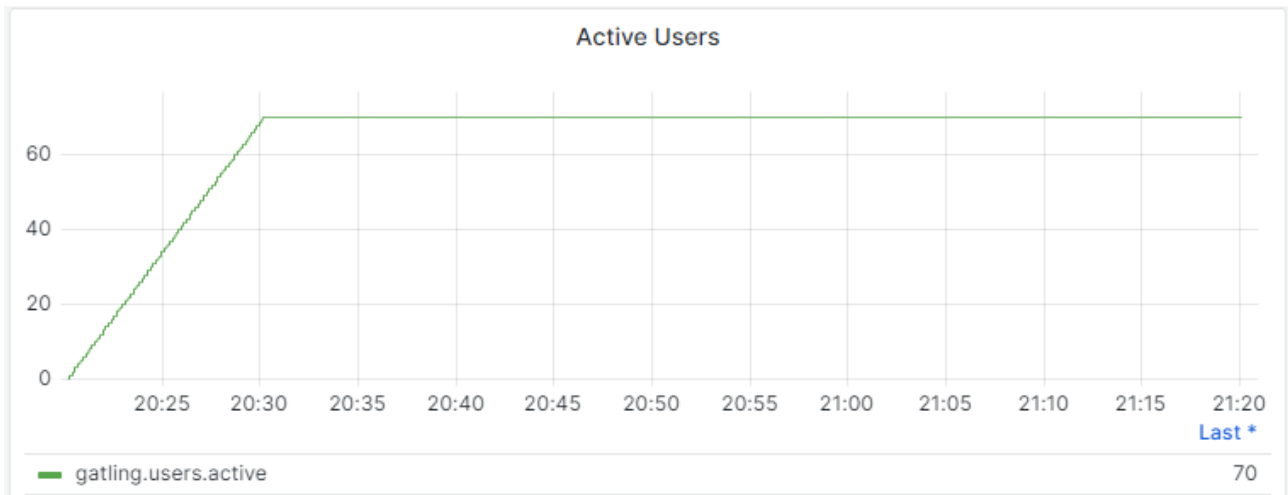


Рис.5 Профиль нагрузки для определения ключевых показателей.

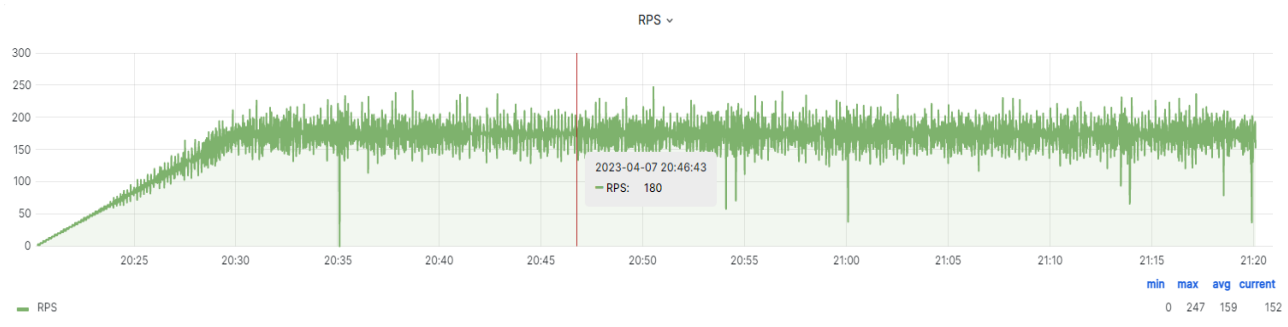


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

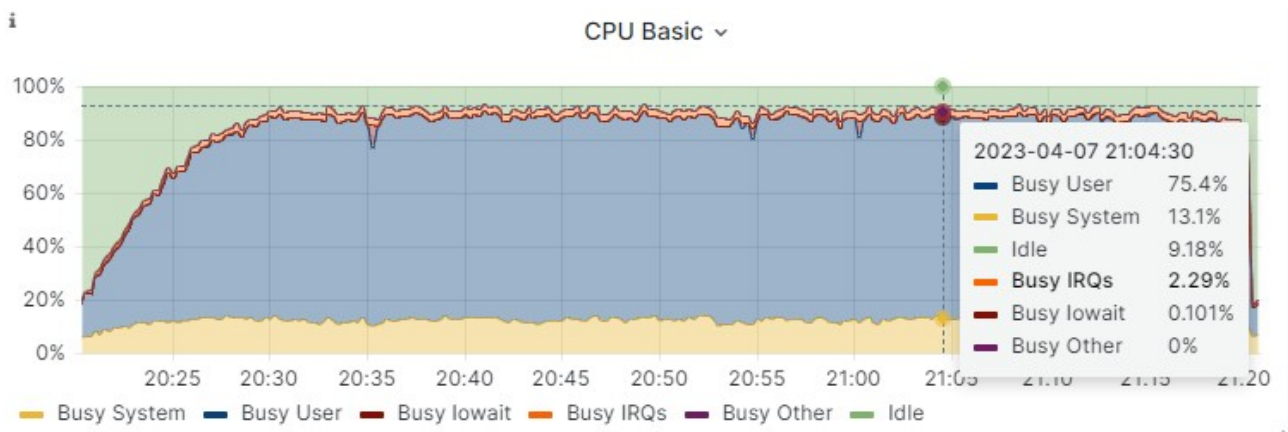


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 90%

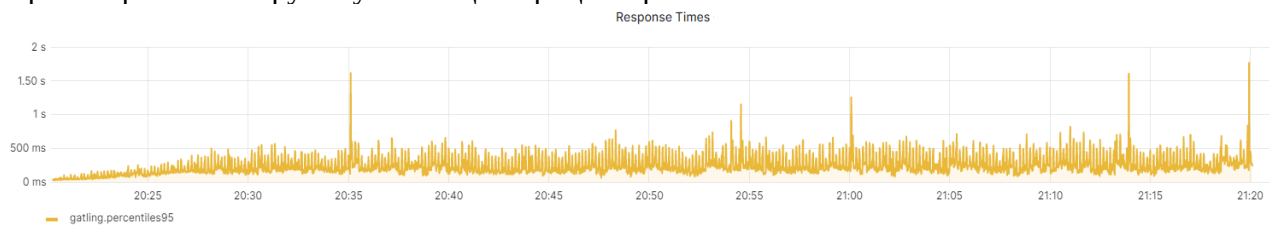


Рис.8. Времена ответа 95% при стабильной нагрузке.

Времена ответа стабильны на всем протяжении теста.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
GET /odata/v1/mappingregistry	572325	12	1793	132	100	175	354

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса MappingService

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 20.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Определение соответствия системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

–Количество запросов и событий от модулей - до 30 в секунду

–Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса MappingService составила 40 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 85% составляют:

Запрос	95% line, ms
POST /api/v1/mappingservice/state	447

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис «MappingService». В тесте использовался следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
POST /api/v1/mappingservice/state	100

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в секунду.

6. Использование заглушек:

Для работы микросервис «MappingService» взаимодействует с микросервисом «ManagementApi». В ходе тестирования сервис «ManagementApi» был заглушен с использованием утилиты «Wiremock», а именно, следующие запросы:

```
GET /odata/v1/Module?$filter=Code eq '{code}'
```

```
GET /odata/v1/ModuleEvents?$filter=(ModuleId eq null or ModuleId eq '{moduleid'}) and
```

```
Is Active
```

```
GET /odata/v1/ModuleEvents?$filter=(EventId in
```

```
('{eventid1}','{eventid2}','{eventid3}','{eventid4}','{eventid5}','{eventid6}','{eventid7}')) and Is
```

```
Active
```

```
GET/odata/v1/Module?$filter=(Id in
```

```
('{id}','{id2}','{id3}','{id4}','{id5}','{id6}','{id7}','{id8}','{id9}')) and true
```

```
GET /odata/v1/ModuleEvents?$filter=(Id in ('{id}')) and Is Active
```

```
GET /odata/v1/Module?$filter=( Id in ('{id}')) and true
```

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейно нарастающая нагрузка на микросервис согласно следующему профилю:

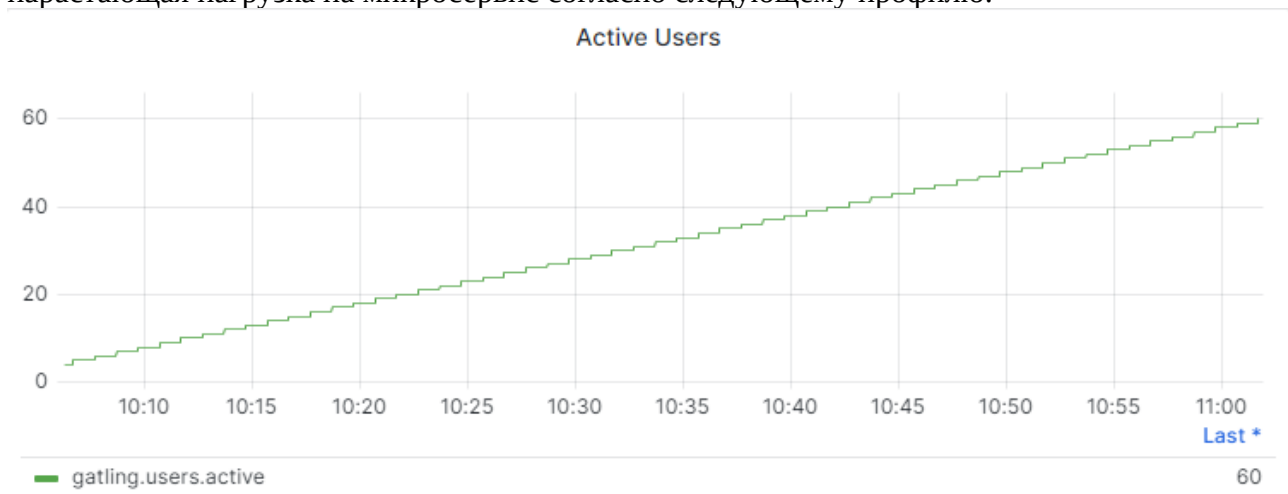


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную 40 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 достигнув максимального значения перестали расти и ресурсы CPU были практически полностью утилизированы.

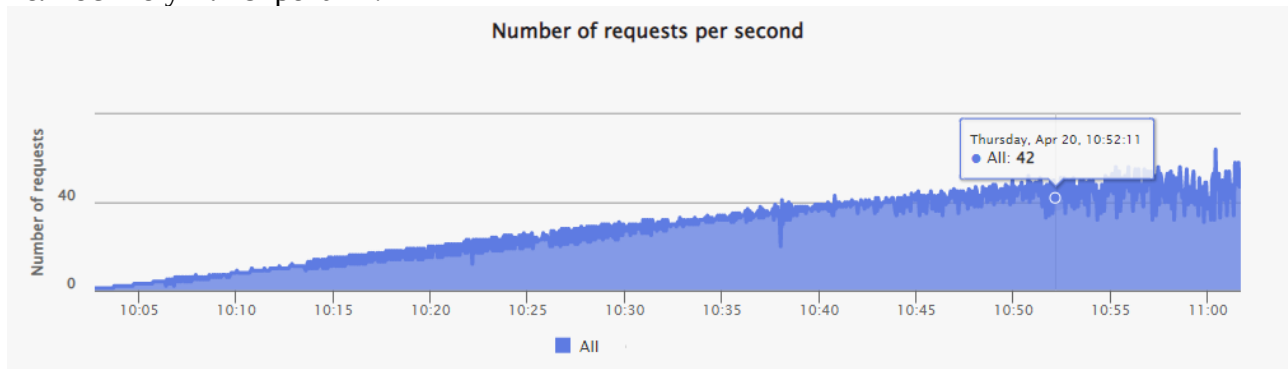


Рис.2. Пропускная способность для растущей нагрузки.

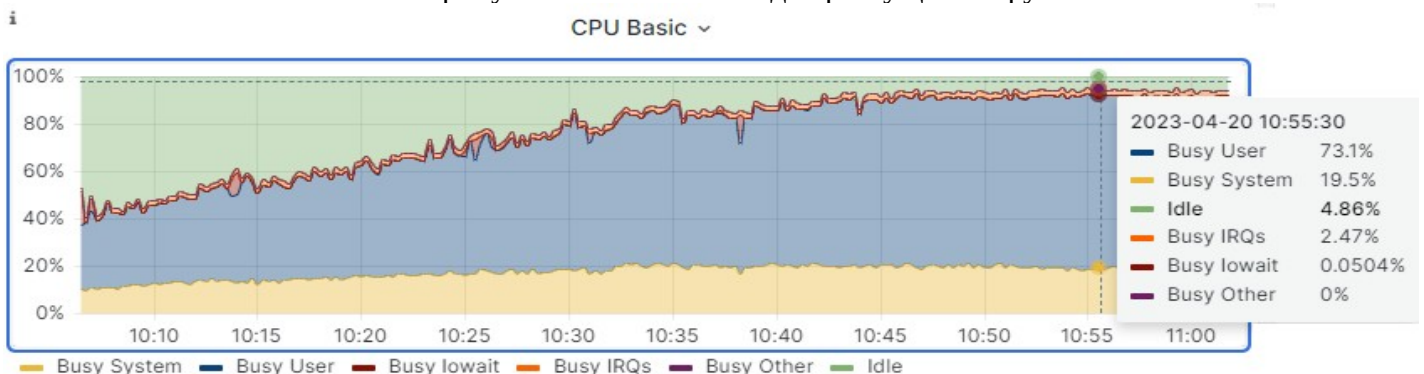


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

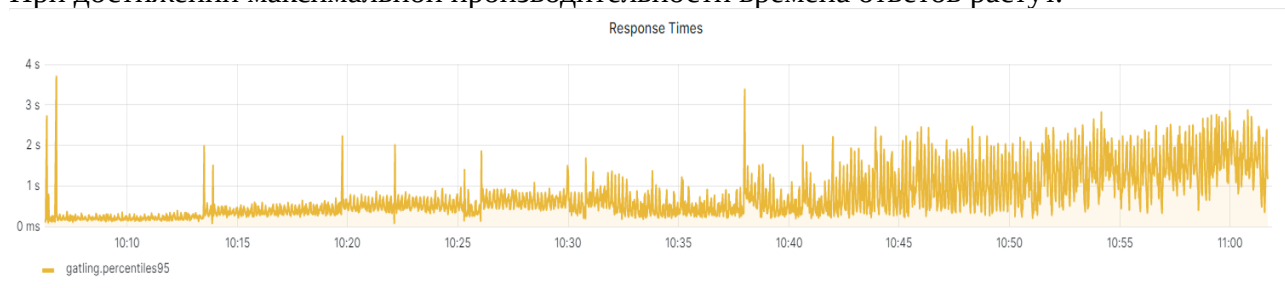


Рис.4. Динамика времен ответа 95% при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 32 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

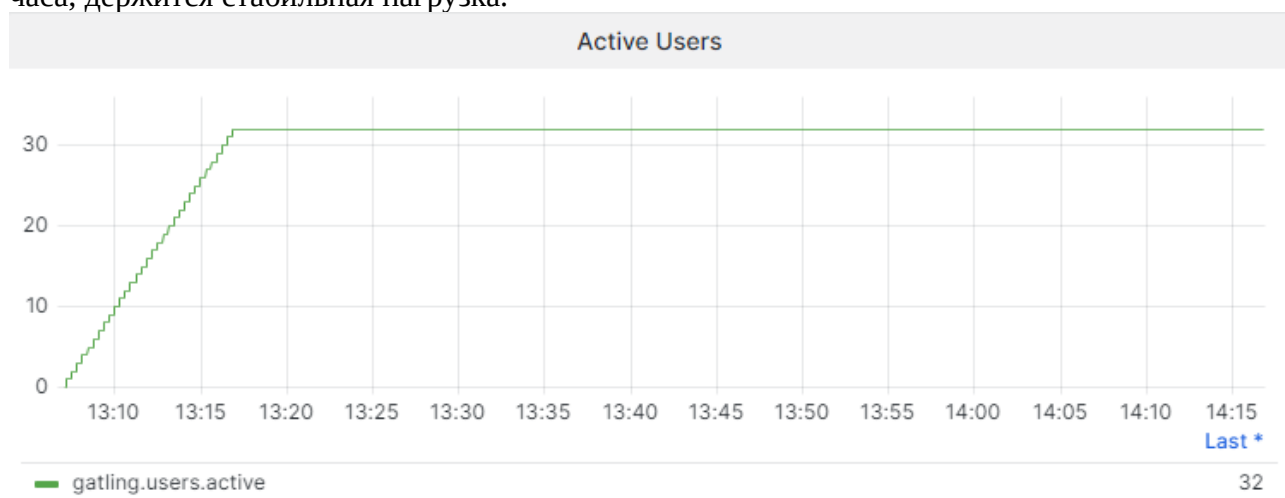


Рис.5. Профиль нагрузки для определения ключевых показателей.

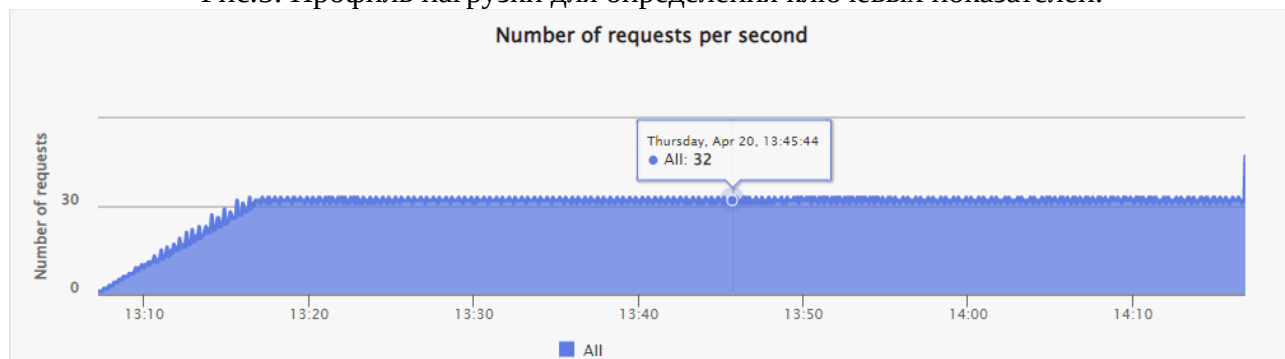


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

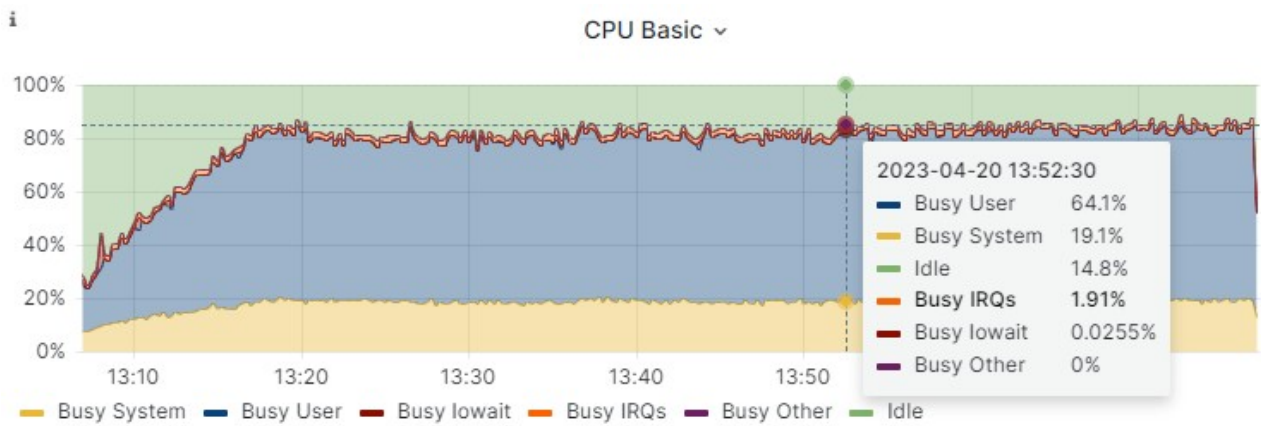


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 85%

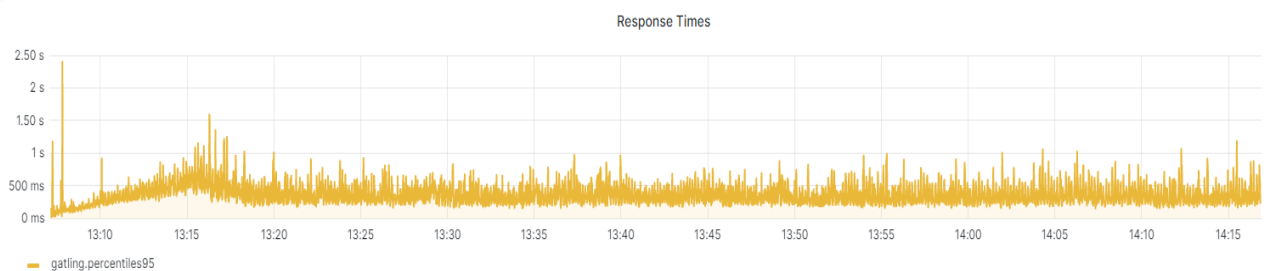


Рис.8. Времена ответа 95% при стабильной нагрузке.

Выброс в начале теста связан с прогревом системы и быстрым разгоном. Далее времена ответа снижаются. В остальном времена ответа стабильны при стабильной нагрузке.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
POST /api/v1/mappingservice/state	124499	11	2399	176	140	234	447

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса PortalApi

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 24.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Определение соответствия системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

–Количество запросов и событий от модулей - до 30 в секунду

–Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса PortalApi составила 65 запросам в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 90% составляют:

Запрос	95% line, ms
GET /api/v1/files/form/{formId}/manifest	562

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

5. Модель нагрузки:

Тестировался микросервис «PortalApi». В тесте использовался следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
GET /api/v1/files/form/{formId}/manifest	100

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в секунду.

6. Использование заглушек:

Для работы микросервис «PortalApi» взаимодействует с микросервисом «ManagementApi» и «UserControlApi». В ходе тестирования сервис «ManagementApi» и «UserControlApi» были заглушены с использованием утилиты «Wiremock», а именно, следующие запросы:

GET /v1.0/invoke/core_managment/method/odata/v1/\$metadata

GET /v1.0/invoke/core_user_control/method/odata/v1/\$metadata

GET /v1.0/invoke/core_license/method/api/v1/license/status

GET /v1.0/invoke/core_user_control/method/api/v1/user/settings

GET /v1.0/invoke/core_managment/method/odata/v1/ModuleForms?\$filter=Id%20eq%20%27{formId}%27%20and%20true

GET /v1.0/invoke/core_managment/method/odata/v1/ModuleCode?\$filter=Id%20eq%20%27{moduleId}%27%20and%20true

GET /v1.0/invoke/core_user_control/method/odata/v1/PermissionForm?\$filter=Form%2FCode%20eq%20%27{formCode}%27%20and%20Module%2FCode%20eq%20%27{moduleId}%27

GET /v1.0/invoke/core_managment/method/odata/v1/ModuleFormHistories?\$filter=Id%20eq%20%27{fileFormHistoryId}%27%20and%20true

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейно нарастающая нагрузка на микросервис согласно следующему профилю:

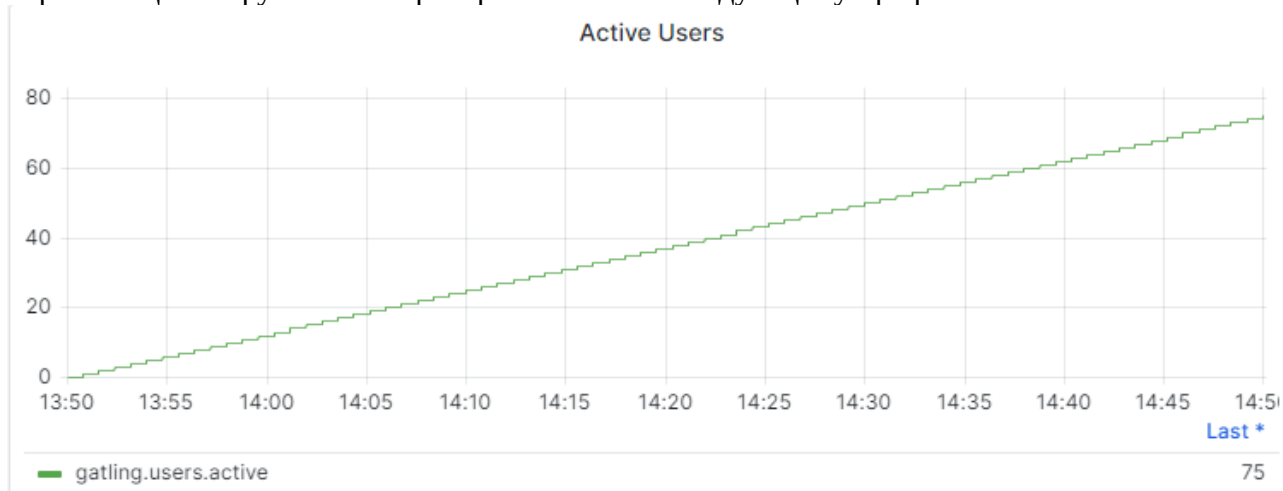


Рис.1 Профиль нагрузки для определения максимальной производительности.

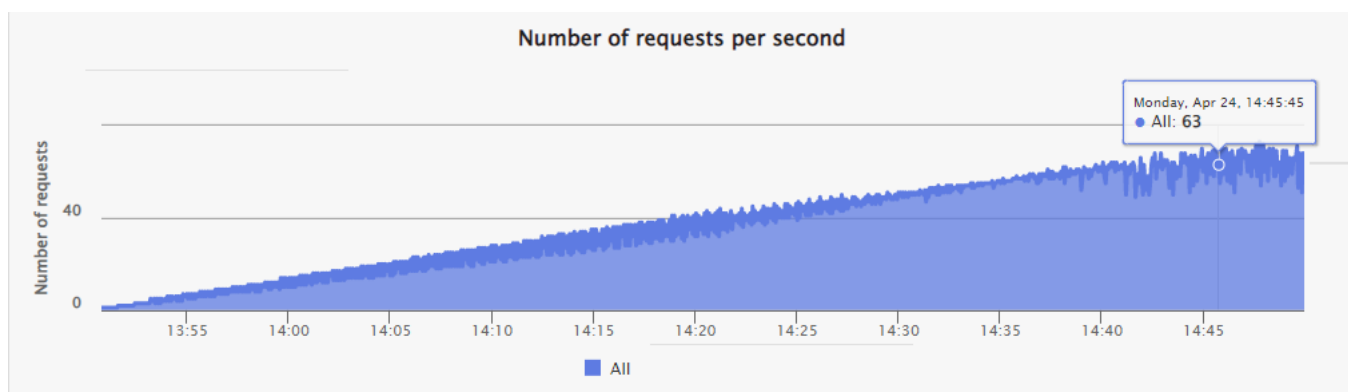


Рис.2. Пропускная способность для растущей нагрузки.

В результате определили максимальную производительность равную 65 запросам в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 достигнув максимального значения перестали расти и ресурсы CPU были практически полностью утилизированы.

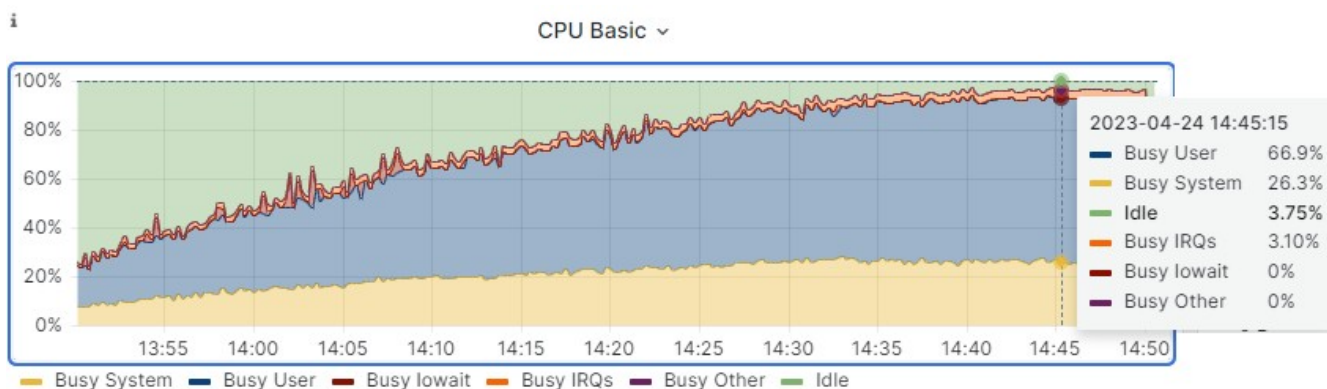


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

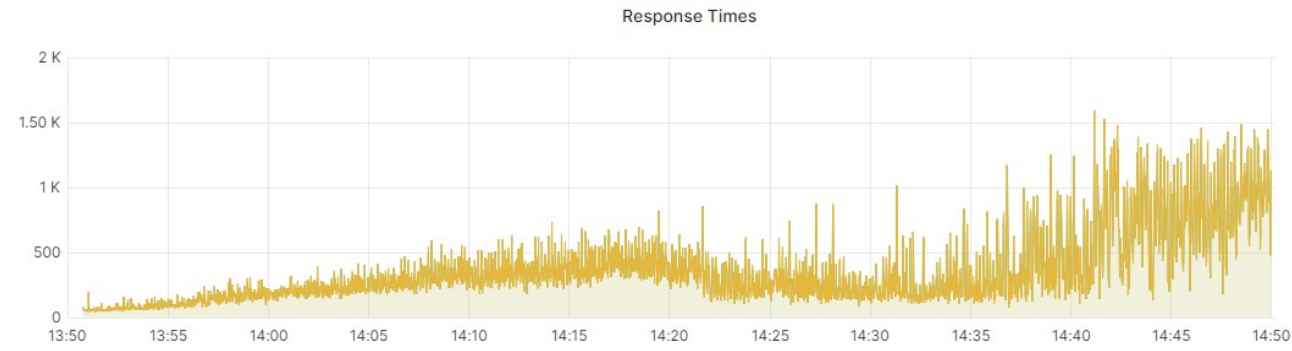


Рис.4. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 45 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

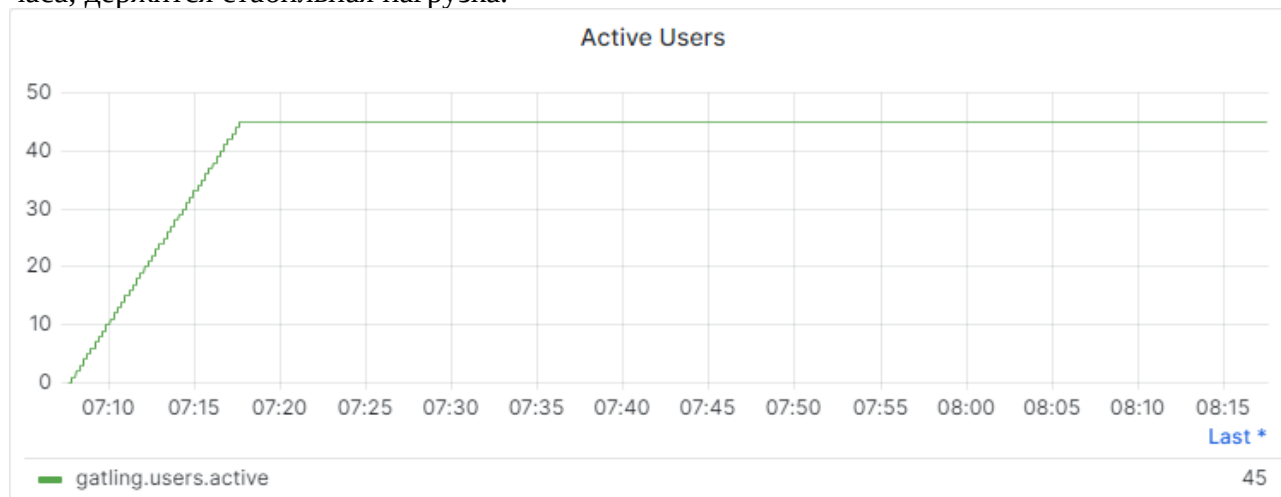


Рис.5. Профиль нагрузки для определения ключевых показателей.

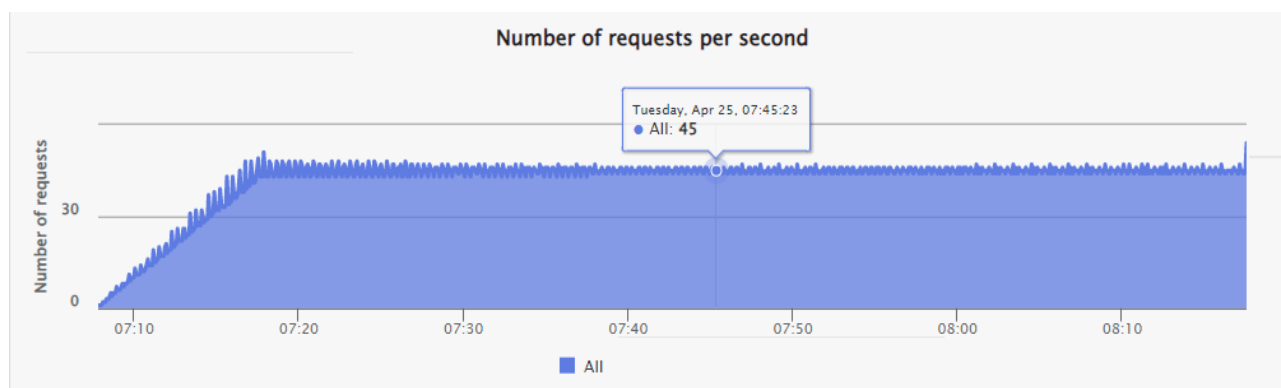


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.,

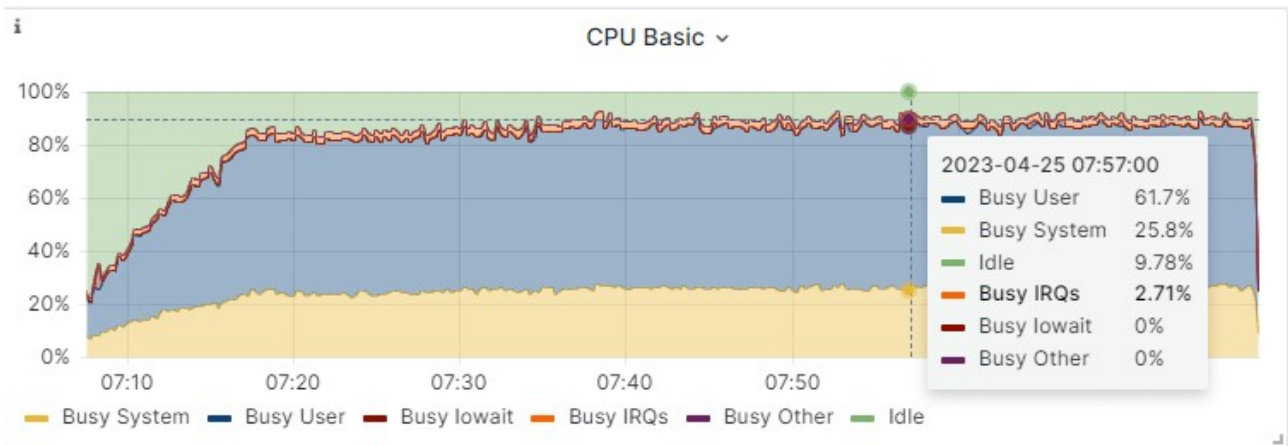


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 90%

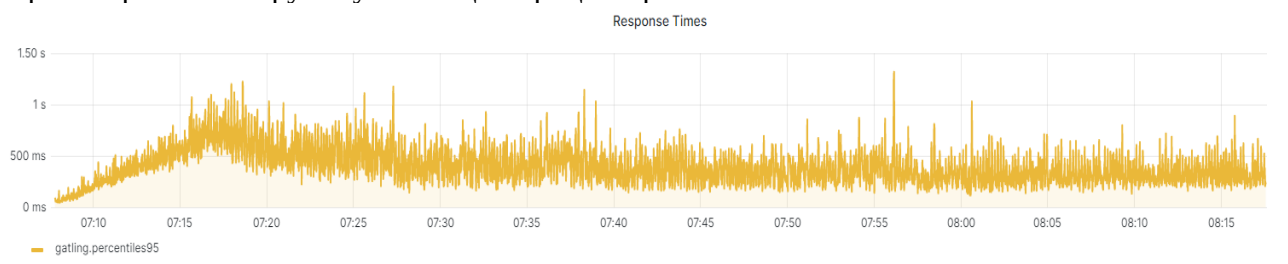


Рис.8. Времена ответа 95% при стабильной нагрузке.

Выброс в начале теста связан с прогревом системы и быстрым разгоном. Далее времена ответа снижаются. В остальном времена ответа стабильны при стабильной нагрузке.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
GET /api/v1/files/form/{formId}/manifest	175201	22	1571	240	199	322	562

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса SchemaRegistry

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 05.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Соответствие системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

- Количество запросов и событий от модулей - до 30 в секунду.
- Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 16Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса SchemaRegistry составила 40 запросов в секунду, что удовлетворяет требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Ошибок нет. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора порядка 90% составляют:

Запрос	95% line, ms
GET /odata/v1/schemaregistry	1640

5. Модель нагрузки:

Тестировался микросервис “SchemaRegistry”. В тесте использовались следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
GET /odata/v1/schemaregistry	100

В рамках одного потока нагрузка подавалась с интенсивностью 1 запрос в 2 секунды.

6. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась ступенчатая нагрузка на микросервис согласно следующему профилю:

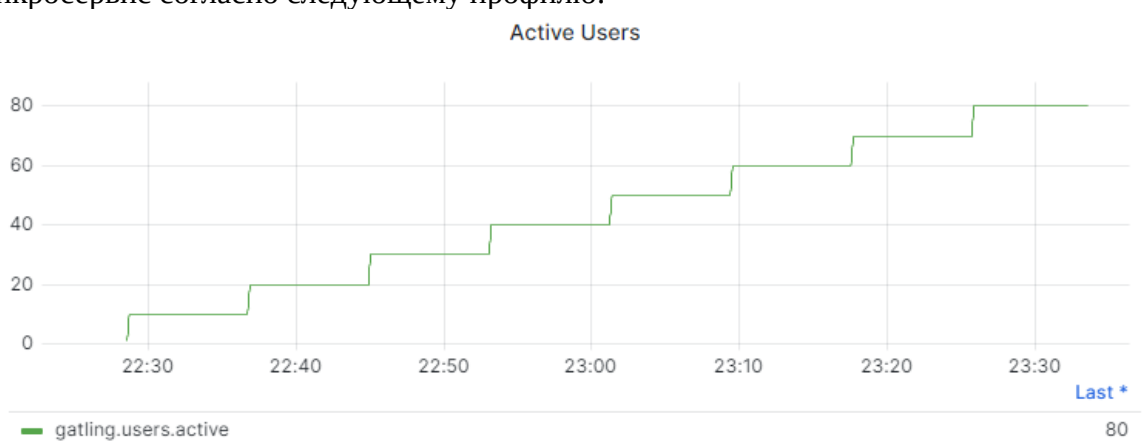


Рис.1 Профиль нагрузки для определения максимальной производительности.

В результате определили максимальную производительность равную 40 запросам в секунду. Такой вывод был сделан на основании того, что ресурсы CPU были практически полностью утилизированы.

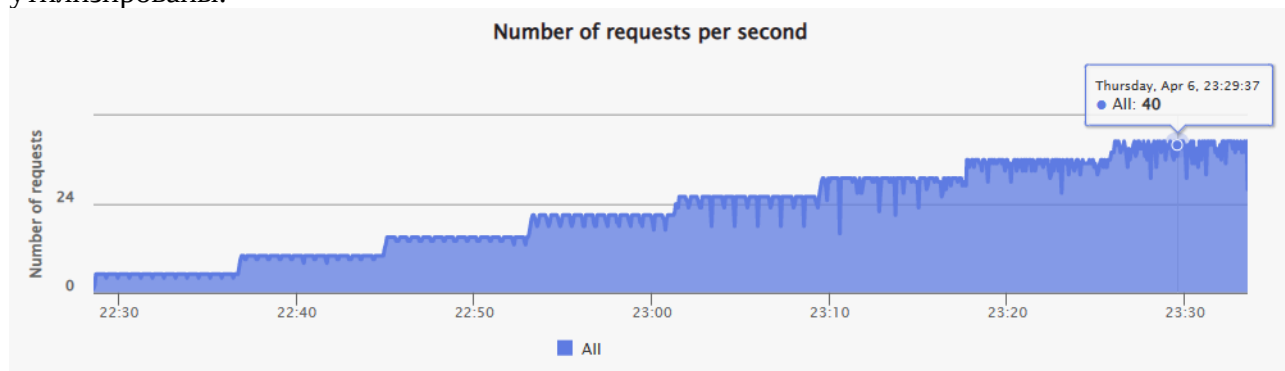


Рис.2. Пропускная способность для растущей нагрузки.

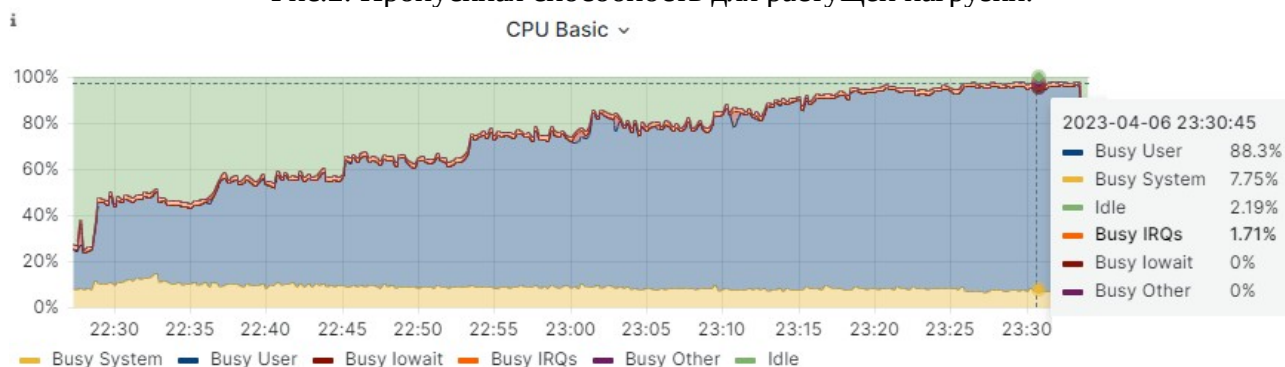


Рис.3. Утилизация CPU на хостовой виртуальной машине.

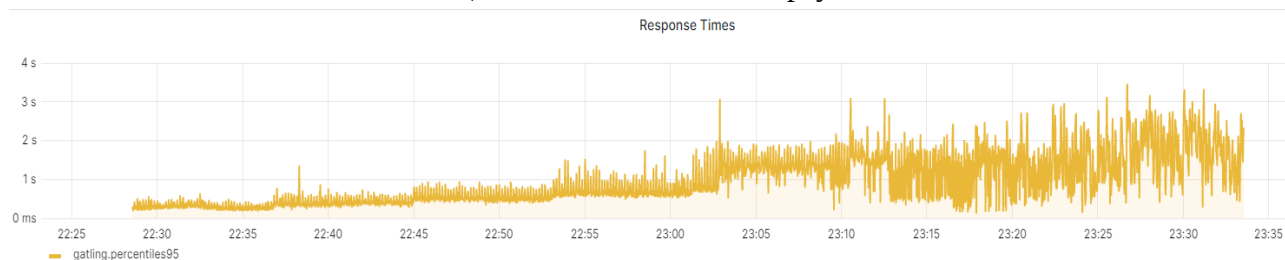


Рис.4. Динамика времен ответа 95% при растущей нагрузке.

При достижении максимальной производительности времена ответов растут.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки была выбрана ступень в 60 потоков, что соответствует значению в 30 запросов в секунду из требований. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

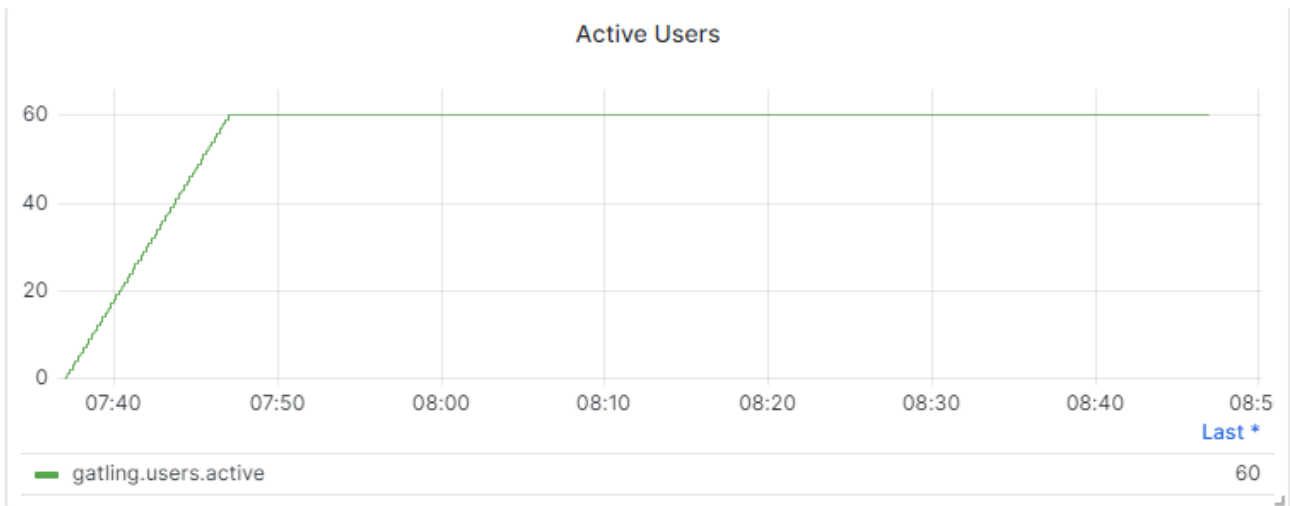


Рис.6 Профиль нагрузки для определения ключевых показателей.

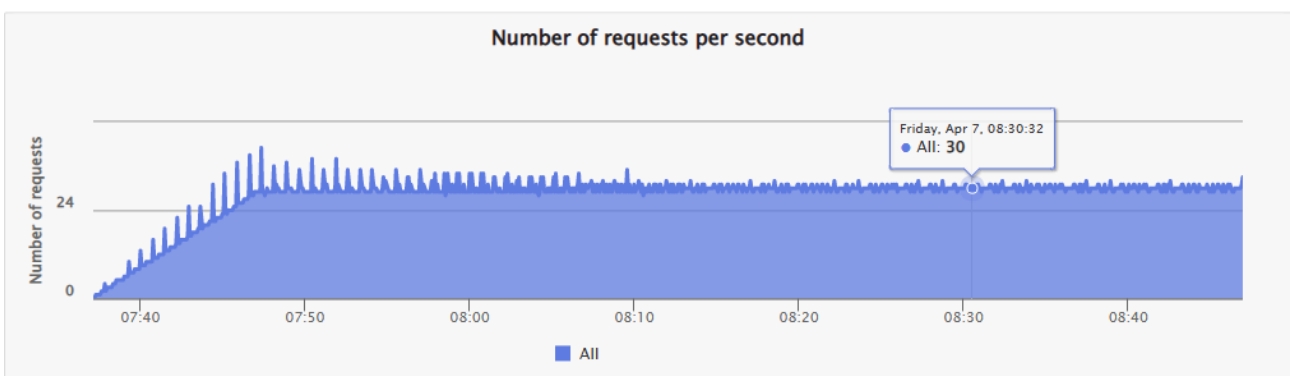


Рис.7. Пропускная способность с стабильной нагрузкой.

Нагрузка в течении теста оставалась стабильной.

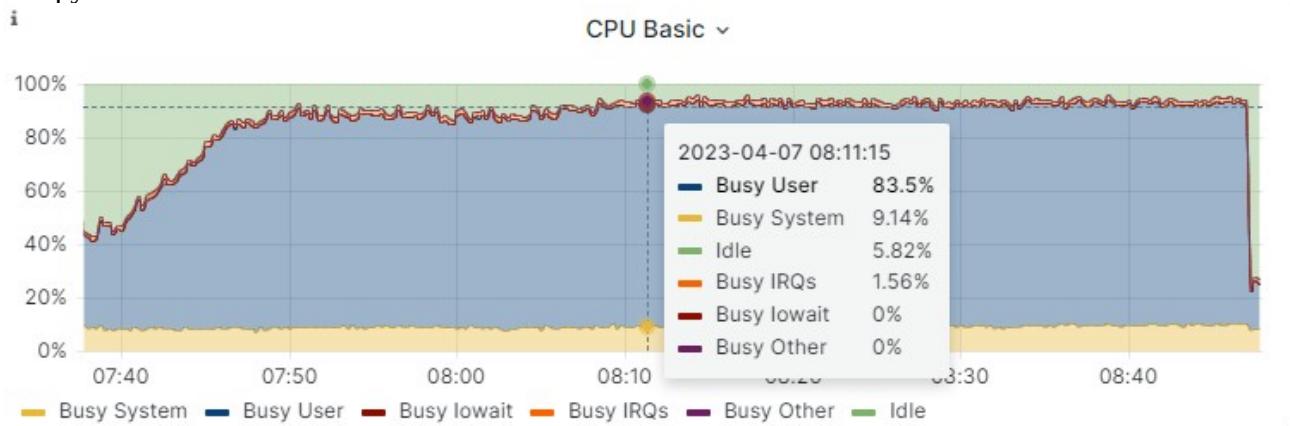


Рис 8. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила 90%

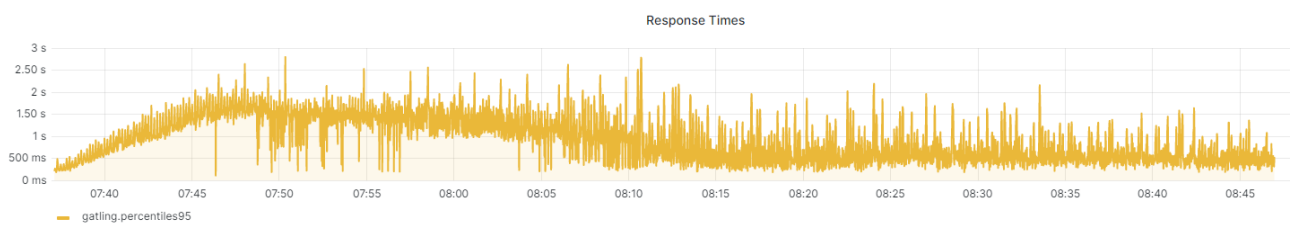


Рис.9. Времена ответа 95% при стабильной нагрузке.

Периода разгона сопровождается ростом времен ответа. После которого наблюдается плавное снижение значений. Это связано с стабилизацией системы после быстрого разгона нагрузки. В момент стабилизации происходит очистка памяти в контейнере и наблюдается небольшой рост утилизации CPU (см. Рис.8). Вероятно это связано с работой GC. Для подтверждения требуется проведение дополнительных исследований.

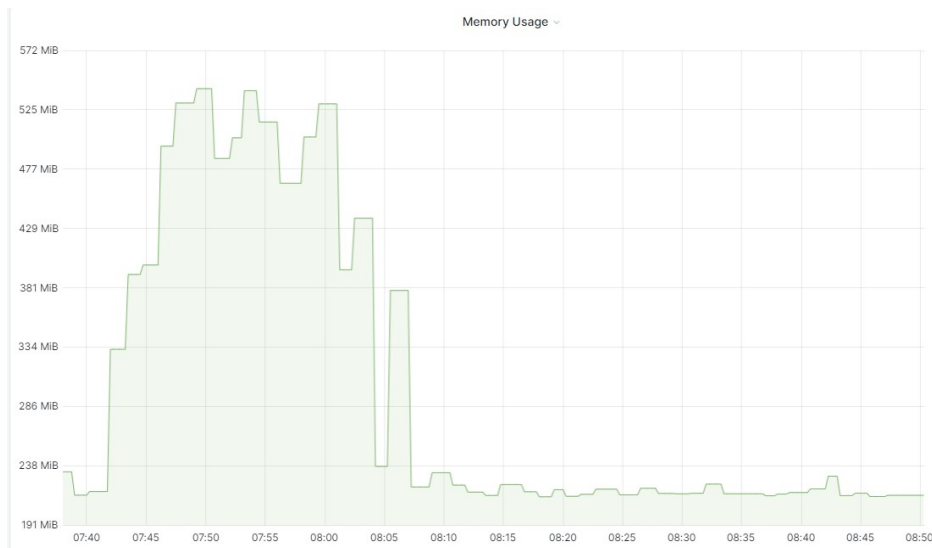


Рис.10. Динамика утилизации памяти в контейнере

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
GET /odata/v1/schemaregistry	116825	75	3203	767	640	1187	1640

Таблица 1. Времена ответа при стабильной нагрузке.

ОТЧЕТ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ микросервиса UserControl

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 25.04.2023

1. Цели тестирования:

Основными целями нагрузочного тестирования являются:

1.1. Определение максимальной производительности.

1.2. Определение соответствия системы предъявляемым требованиям.

1.3. Выявление в аппаратно-программной конфигурации продуктовой среды узких мест, определяющих полученные результаты.

2. Требования:

–Количество запросов и событий от модулей - до 30 в секунду

–Для пользователей допускаются задержки в интерфейсе до 5 секунд.

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

Максимальная производительность микросервиса UserControl составила 26 запросов в секунду, что не соответствует требованиям в 30 запросов в секунду. Отсутствуют ограничения не связанные с аппаратными ресурсами. Производительность ограничена утилизацией CPU. Времена ответа (95 перцентиль) при стабильной нагрузке с утилизацией процессора 92% составляют:

Запрос	95% line, ms
PUT /api/v1/permission/role/{role}	259
POST /api/v1/permission/state/module/rebuild	2150
POST /api/v1/permission/state/form/rebuild	252

Таблица.1. Времена ответа для 95 перцентилья при стабильной нагрузке

Рекомендации: проанализировать возможность оптимизации медленного запроса, проанализировать причину высокой утилизации cpu при помощи профилирования.

5. Модель нагрузки:

Тестировался микросервис «UserControl». В тесте использовался следующий запрос:

Нагружаемые эндпоинты микросервиса:	%%
PUT /api/v1/permission/role/{role}	40
POST /api/v1/permission/state/module/rebuild	30
POST /api/v1/permission/state/form/rebuild	30

Таблица.2. Нагружаемые эндпоинты и распределение запросов

В рамках одного потока нагрузка подавалась с интенсивностью 6 запросов в 3 секунды.

6. Использование заглушек:

Для работы микросервис «UserControl» взаимодействует с микросервисом «ManagementApi». В ходе тестирования сервис «ManagementApi» был заглушен с использованием утилиты «Wiremock», а именно, следующие запросы:

GET /v1.0/invoke/core_managment/method/odata/v1/\$metadata

GET /v1.0/invoke/core_managment/method/odata/v1/ModuleForms?\$filter=Code eq '{formCode}' and ModuleId eq '{moduleId}'

```

GET /v1.0/invoke/core_management/method/odata/v1/ModuleForms?$filter=Id eq null and
true
GET /v1.0/invoke/core_management/method/api/v1/module/form/{formId}/components
GET/v1.0/invoke/core_management/method/api/v1/module/form/formId/useForms
GET /v1.0/invoke/core_management/method/odata/v1/ModuleCode?$filter=Id eq
'{moduleId}' and true
GET /v1.0/invoke/core_management/method/api/v1/module/{moduleId}/permissions
GET /v1.0/invoke/core_management/method/odata/v1/ModuleCode?$filter=Id eq
'{moduleId}' and true,
GET /v1.0/invoke/core_management/method/odata/v1/ModuleEvents?$filter=ModuleId eq
'{moduleId}'
GET /v1.0/invoke/core_management/method/odata/v1/ModuleForms?$filter=ModuleId eq
'{moduleId}'

```

7. Ход нагрузки:

Тест на определение максимальной производительности.

Для определения максимальной производительности подавалась линейно нарастающая нагрузка на микросервис согласно следующему профилю:

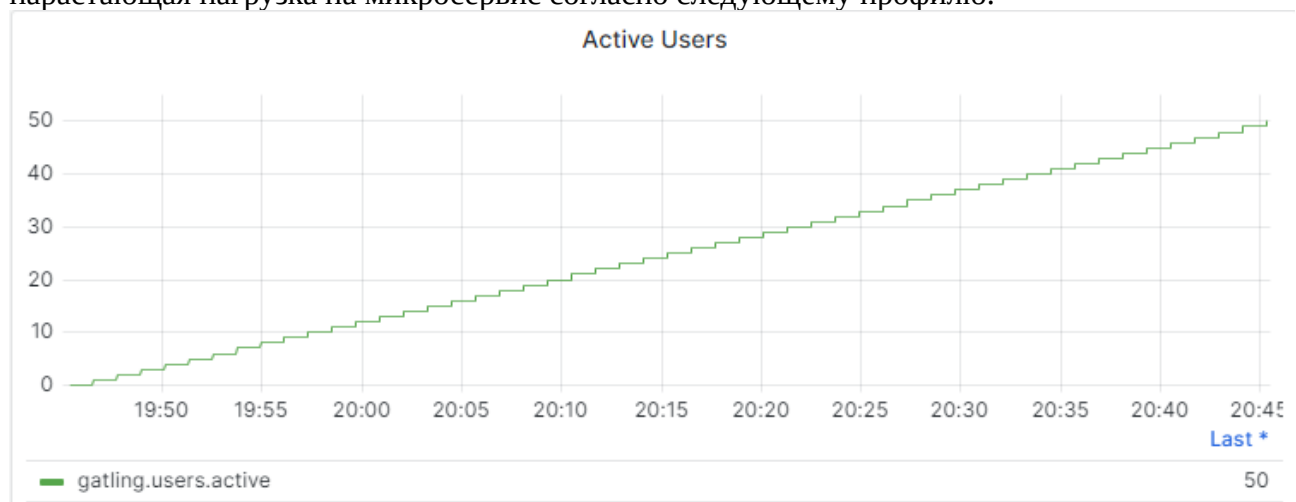


Рис.1 Профиль нагрузки для определения максимальной производительности.

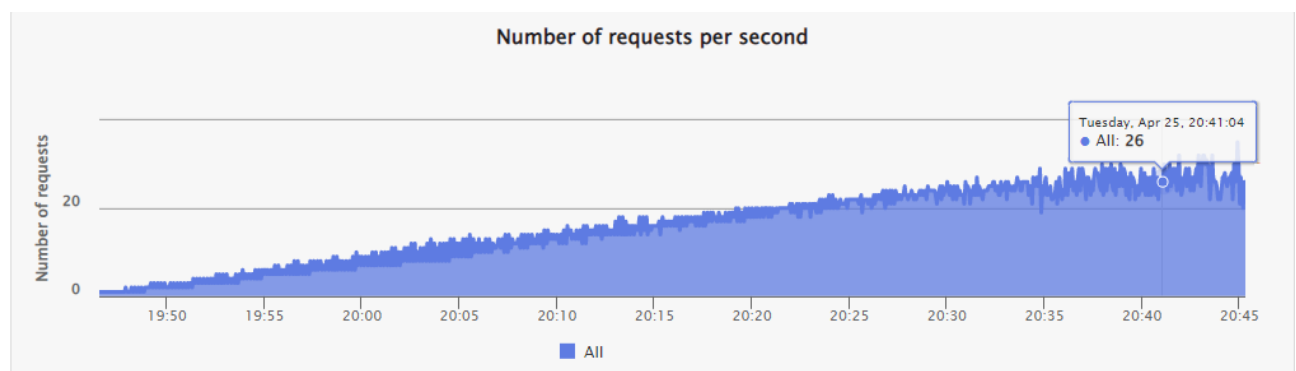


Рис.2. Пропускная способность для растущей нагрузки.

В результате определили максимальную производительность равную 26 запросов в секунду. Такой вывод был сделан на основании того, что значения пропускной способности на Рис 2 достигнув максимального значения перестали расти и ресурсы CPU были практически полностью утилизированы.

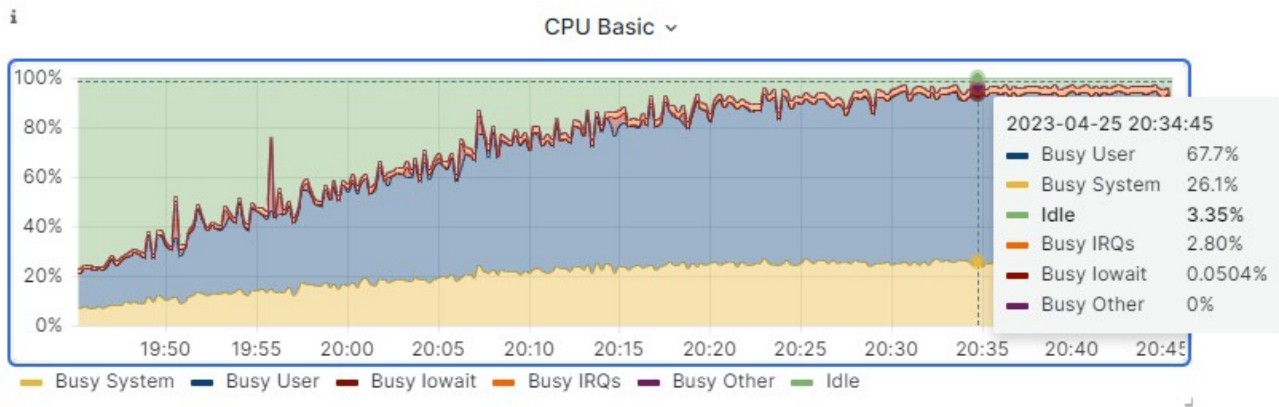


Рис.3. Утилизация CPU на хостовой виртуальной машине.

При достижении максимальной производительности времена ответов растут.

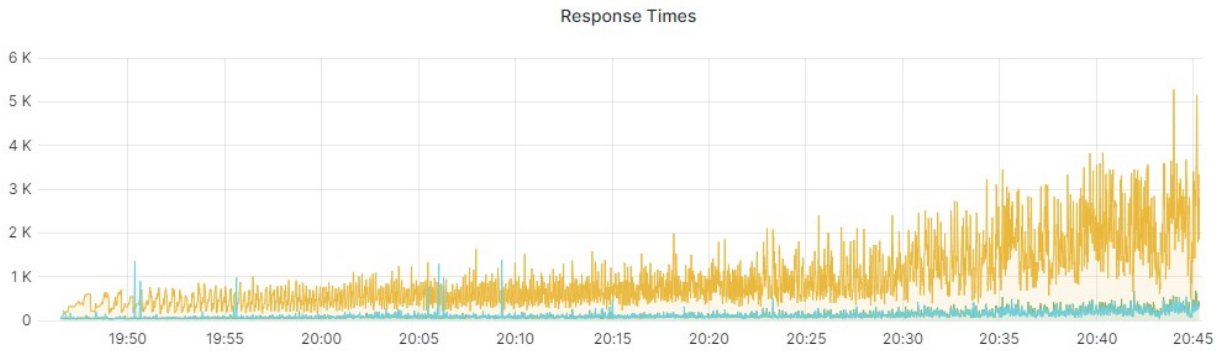


Рис.4. Динамика времен ответа при растущей нагрузке.

Тест на определение ключевых показателей.

Для подачи стабильной нагрузки был выбран уровень нагрузки из «зоны комфорта» соответствующий 22 запросам в секунду. За 10 минут происходит разгон и затем, в течение часа, держится стабильная нагрузка.

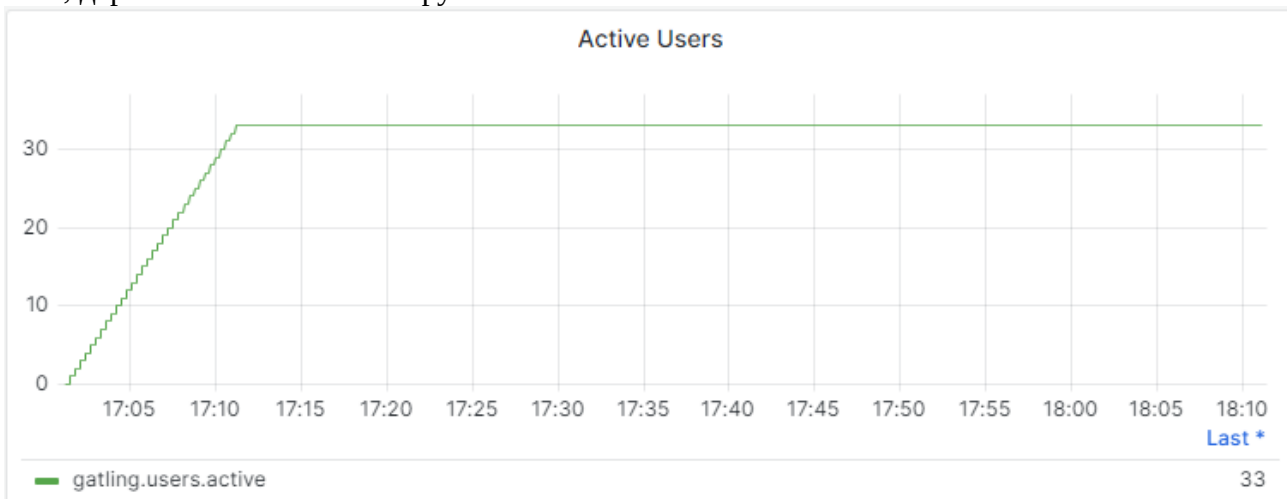


Рис.5. Профиль нагрузки для определения ключевых показателей.

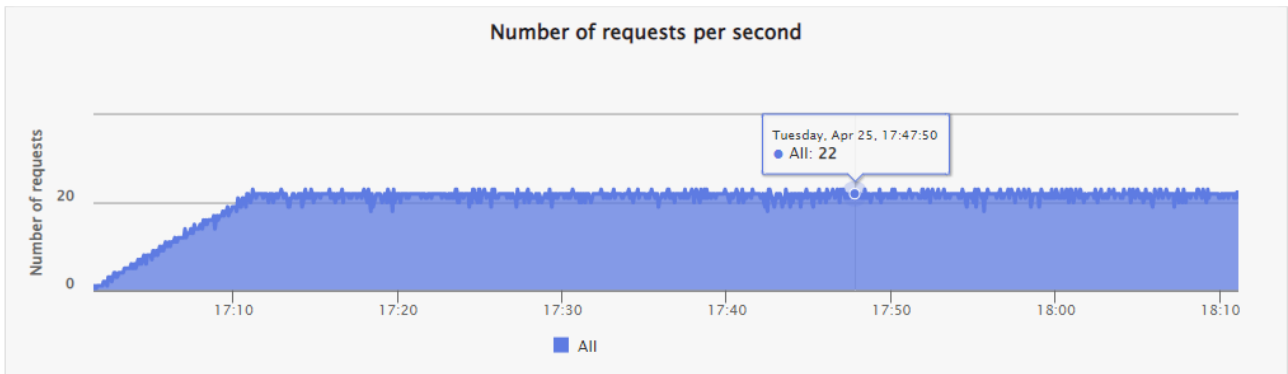


Рис.6. Пропускная способность с стабильной нагрузкой.

Нагрузка в течение теста оставалась стабильной.

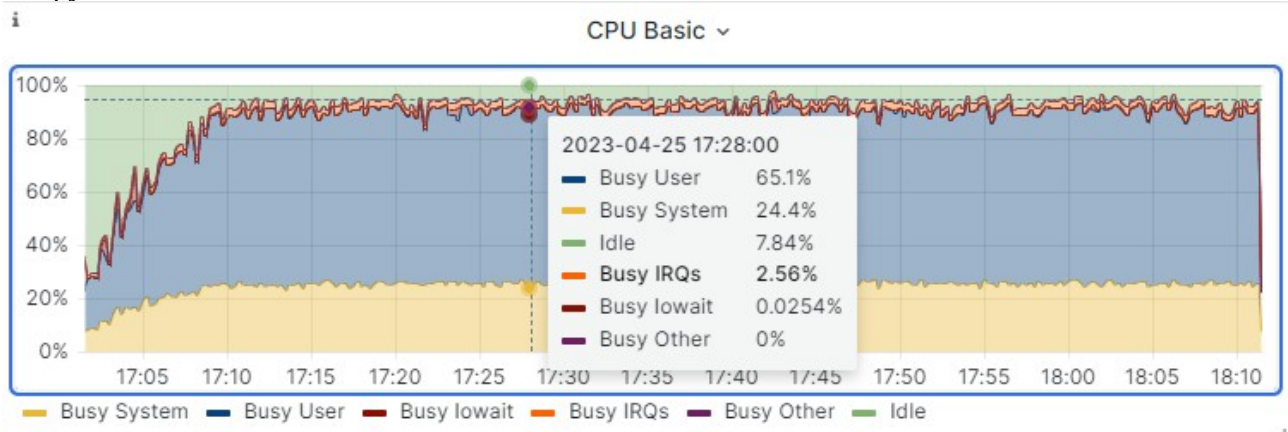


Рис 7. Утилизация процессора при стабильной нагрузке.

При выбранной нагрузке утилизация процессора хостовой машины составила около 92%

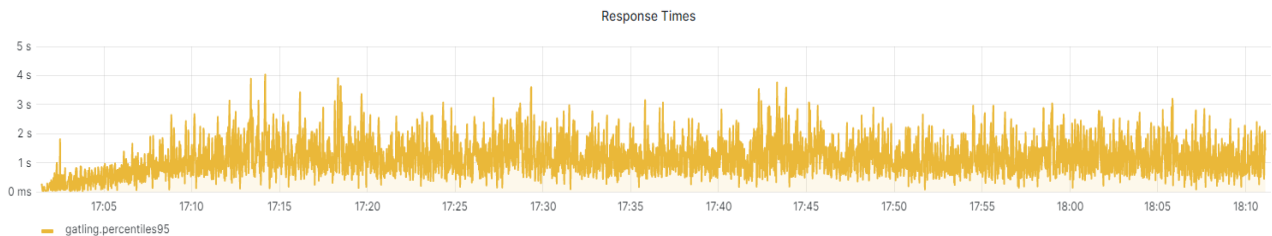


Рис.8. Времена ответа 95% при стабильной нагрузке.

Запрос	Кол-во	Времена ответа, ms					
		Min	Max	Avg	50% line	75% line	95% line
PUT /api/v1/permission/role/{role}	25204	15	3267	120	94	143	259
POST /api/v1/permission/state/module/rebuild	25157	143	2150	1001	853	1291	2150
POST /api/v1/permission/state/form/rebuild	33787	20	1697	113	93	143	252

Таблица 3. Времена ответа при стабильной нагрузке.

ОТЧЕТ

о тестировании добавления манифеста модуля

КОНСОМ. ЯДРО «ИНКА»
Подготовил: Калистратов Д.В.
версия документа: 1.0
Дата 24.05.2023

1. Цели тестирования:

- 1.1. Определение времени выполнения операции
- 1.2. Соответствие системы предъявляемым требованиям.
- 1.3. Сравнение времени выполнения операции до и после оптимизации производительности

2. Требования:

- Время регистрации функционального модуля не более 30 мин

3. Аппаратные характеристики тестируемого стенда:

Сервер приложений: Intel(R) Xeon(R) CPU E5-2660 v4@ 2.00GHz, 4 Core CPU, 20Gb RAM. Лимиты для микросервисов отсутствуют.

4. Выводы:

После проведенных оптимизаций время выполнения операции и утилизация аппаратных ресурсов значительно сократилась. Данные приведены в таблице Таб.1. На модуле «FM.NSS» требование к времени регистрации функциональных модулей (30 минут) выполняется.

Ключевые параметры сравнения	FM.NSS		Bigmodule
	добавление	обновление	добавление
Длительность операции до оптимизации	35 минут	3,5 минуты	Операция не выполнена
Утилизация CPU до оптимизации	100%	80%	100%
Длительность операции после оптимизации	1 минута	1 минута	8ч 52 мин
Утилизация CPU после оптимизации	80%	20%	60%

Таб.1. Сравнительная таблица по ключевым параметрам.

Время регистрации функционального модуля сильно зависит от количества объектов в этом модуле. При увеличении количества сущностей, функций, форм и т. д. длительность операции увеличивается. Следует это учитывать при оценке длительности операции для других функциональных модулей.

Рекомендации:

- запрос на добавления манифеста модуля сделать асинхронным,
- при длительной обработке очереди в kafka необходимо увеличить количество консьюмеров, с пропорциональным увеличением количества партиций в соответствующем топике.

5. Проведенные оптимизации:

Изменение в API:

POST /api/v1/relationship/entity/rebuild - перестроить связи сущностей. Добавлена модель данных (перестроение для модуля или сущности модуля). Убрана возможность перестроения всех сущностей в ядре.

6. Модель тестирования

В ходе тестирования с помощью запроса POST /api/v1/module/manifest добавляли 2 разных модуля:

- модуль «FM.NSS», предоставленный «Консом»

- модуль сгенерированный и содержащий в себе объекты по максимальному количеству указанному в техническом задании к нагрузочному тестированию. («Bigmodule») Модуль содержит 1000 сущностей, 300 параметров одной сущностей, 800 функций, 100 входных параметров функций, 300 выходных параметров функций, 100 тестовых функций, 500 бизнес процессов, 3000 форм, 5000 элементов форм.

Также тестировалась операция обновления модуля «FM.NSS» путем добавления в него одной формы и сущности.

В процессе тестирования снимались аппаратные метрики, на основании которых делался вывод о времени выполнения операции.

7. Ход теста

Добавление манифеста «FM.NSS».

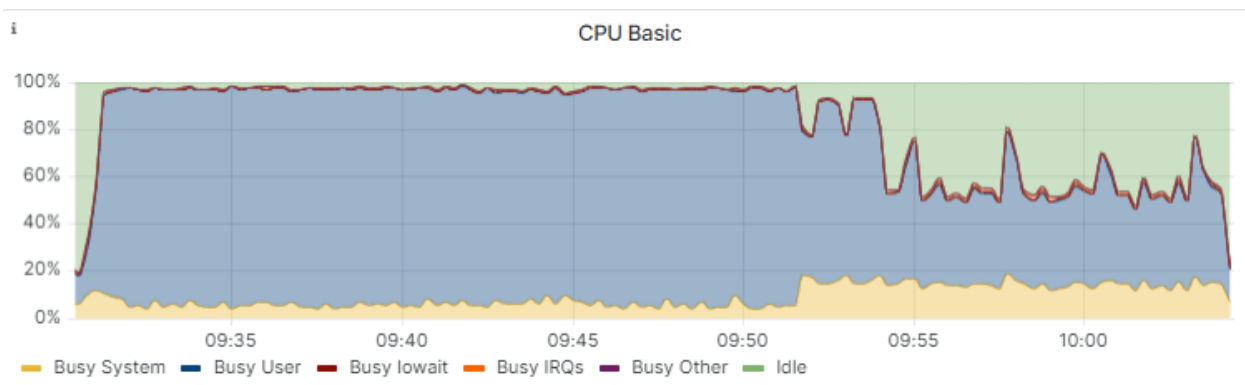


Рис.1. Утилизация CPU при добавлении «FM.NSS» до оптимизаций.

После старта запроса, наблюдается 100% утилизация процессора, которая через некоторое время снижается и возвращается к изначальным показаниям после окончания выполнения операции.

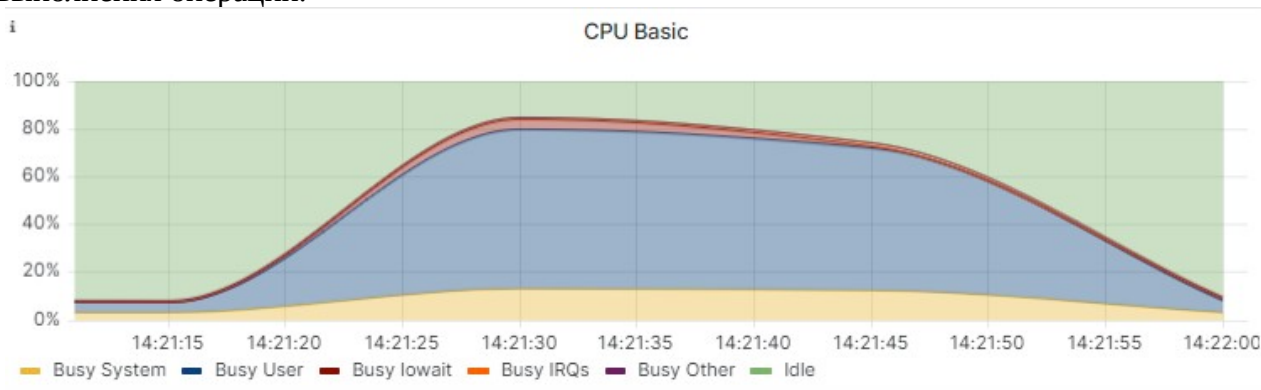


Рис.2. Утилизация CPU при добавлении «FM.NSS» после оптимизаций.

Утилизация процессора ниже чем при добавлении аналогичного модуля до произведенных оптимизаций и ее значение в пике 80%.

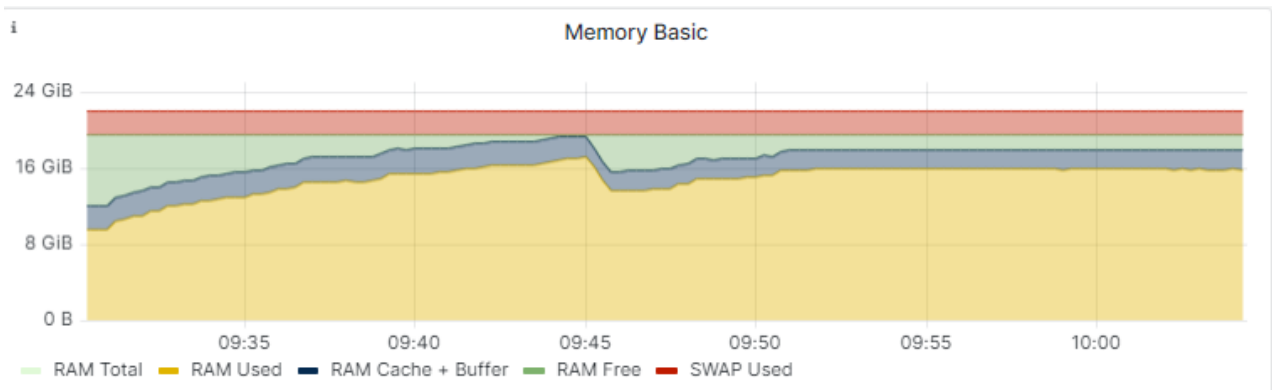


Рис.3. Утилизация RAM при добавлении «FM.NSS» до оптимизаций.

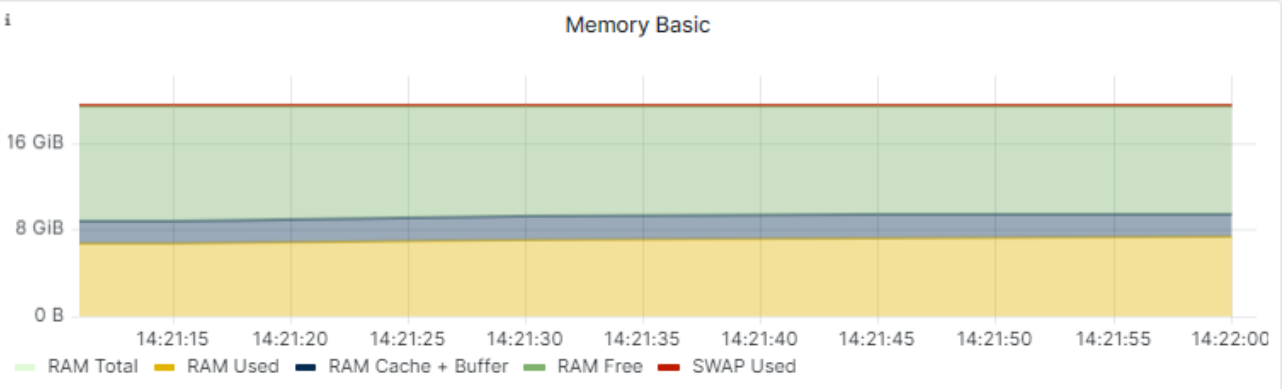


Рис.4. Утилизация RAM при добавлении «FM.NSS» после оптимизаций.

Во втором случае память практически не утилизируется.

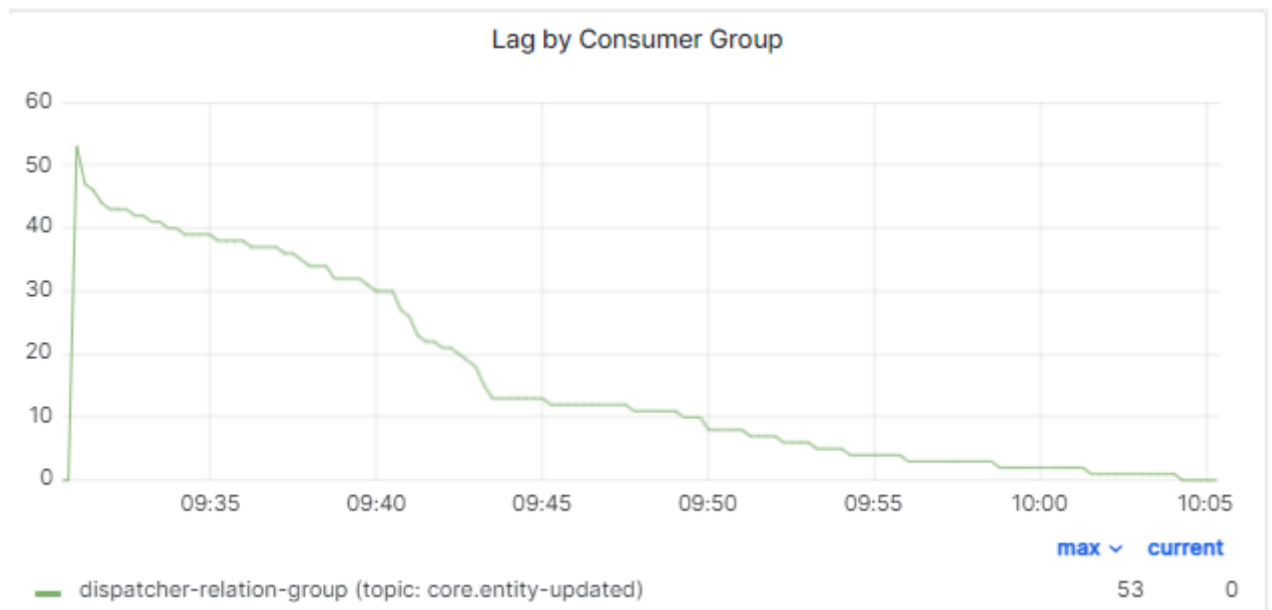


Рис.5. Динамика обработки очереди в kafka при добавлении «FM.NSS» до оптимизаций.

После старта операции в kafka появляется очередь в топике core.entity-updated, которая обрабатывается консьюмером до тех пор пока она не исчерпается.

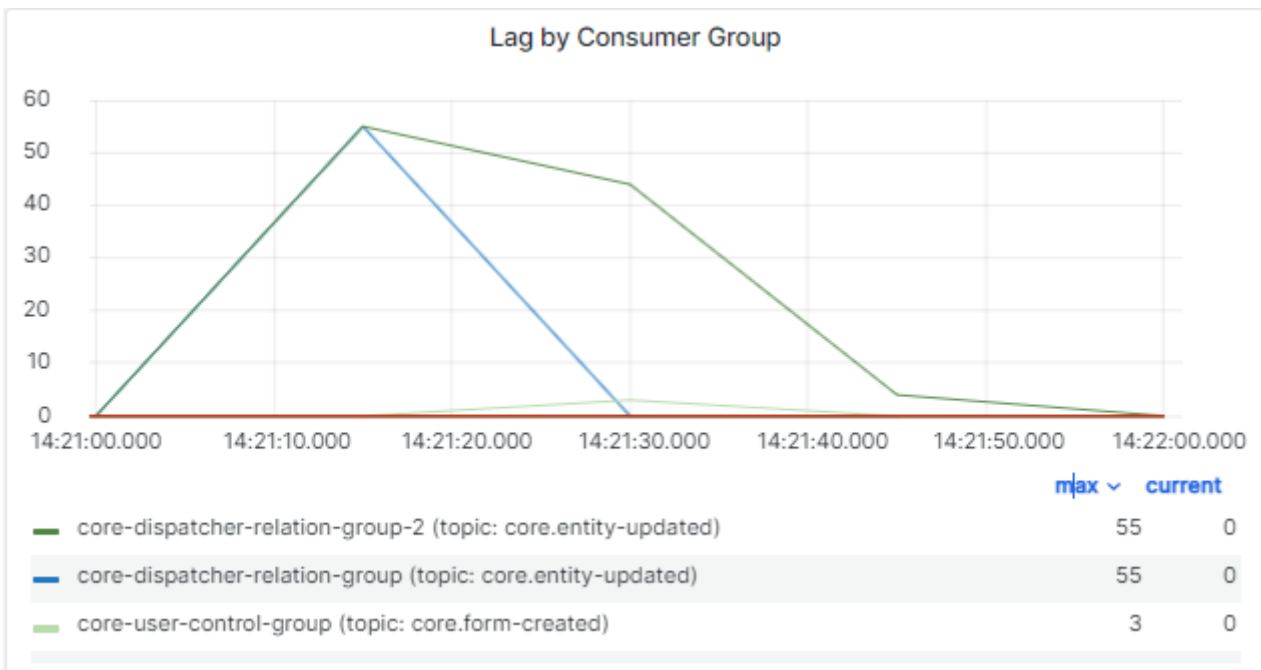


Рис.6. Динамика обработки очереди в kafka при добавлении «FM.NSS» после оптимизаций.

Согласно приведенным графикам Рис 1 и Рис 5. операция по добавлению модуля «FM.NSS» до оптимизаций длилась 35 минут.

Согласно приведенным графикам Рис.2. и Рис.6. операция по добавлению модуля «FM.NSS» после оптимизаций длилась 1 минуту.

Обновление манифеста «FM.NSS».

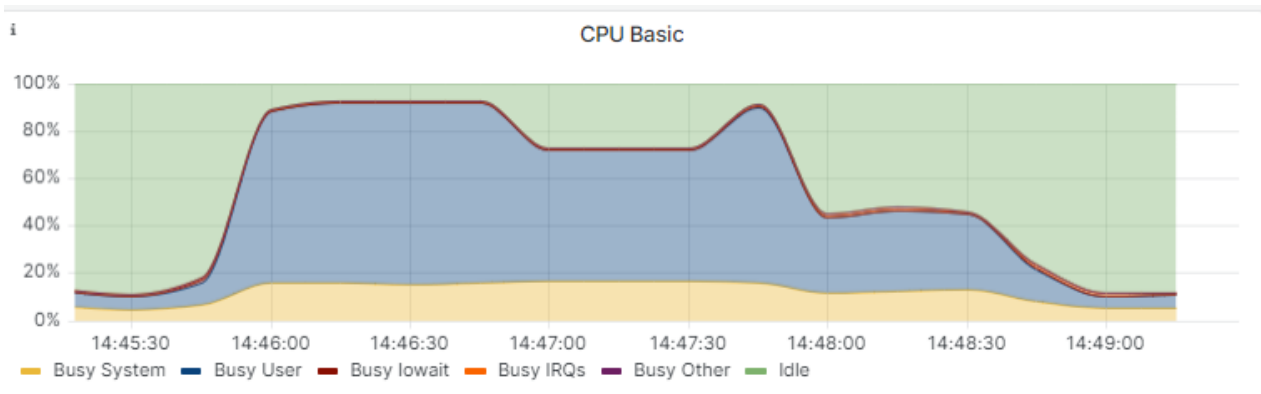


Рис.7. Утилизация CPU при обновлении «FM.NSS» до оптимизаций.

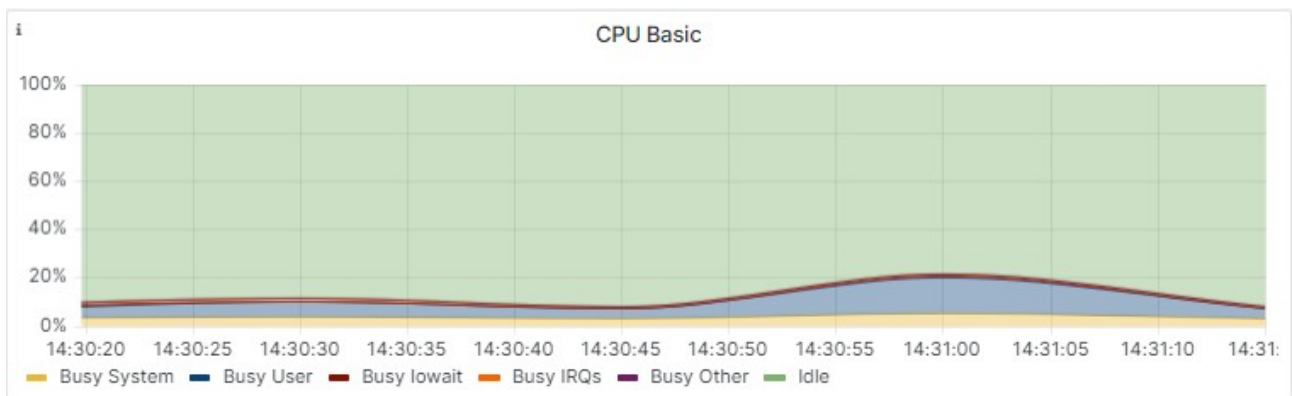


Рис.8. Утилизация CPU при обновлении «FM.NSS» после оптимизаций.

Утилизации CPU практически нет.

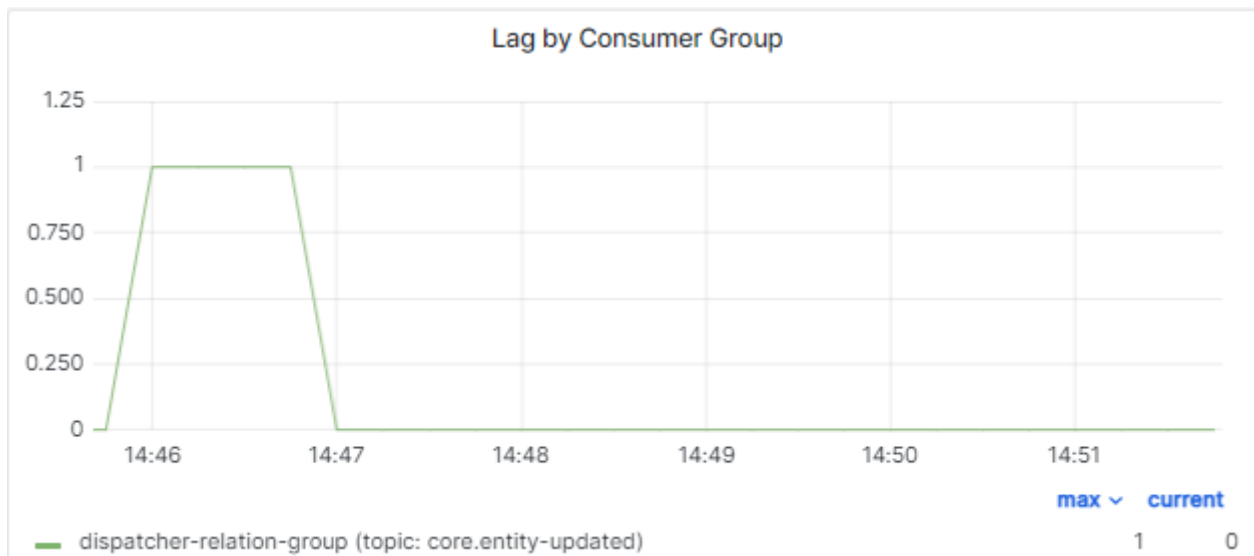


Рис.9. Динамика обработки очереди в kafka при обновлении «FM.NSS».

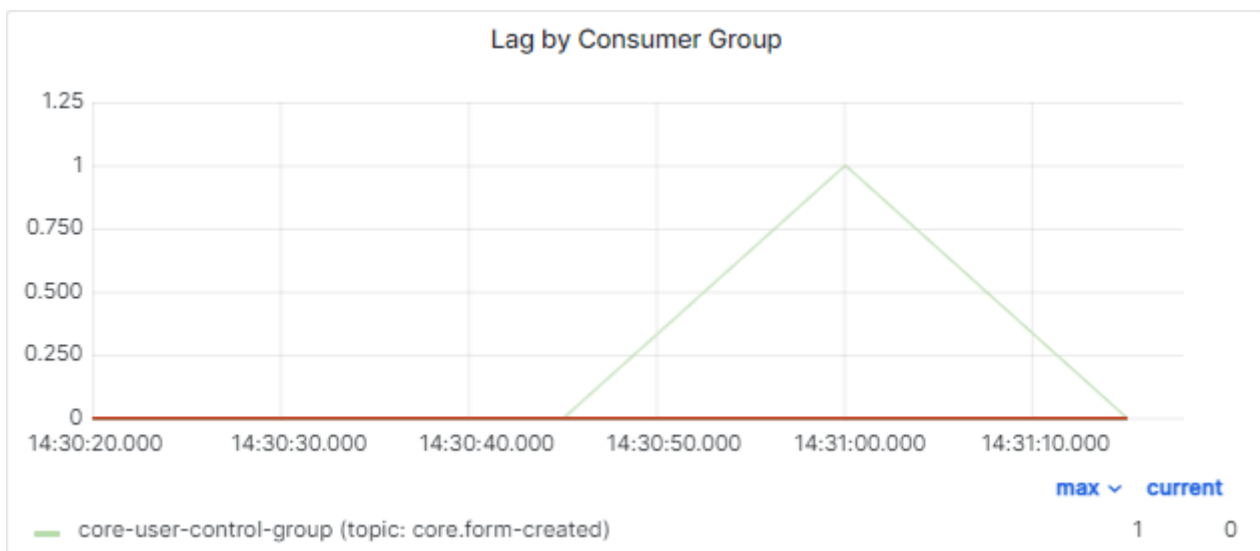


Рис.10. Динамика обработки очереди в kafka при обновлении «FM.NSS» после оптимизаций.

Согласно приведенным графикам Рис.7. и Рис.9. операция по обновлению модуля «FM.NSS» до оптимизаций длилась 3,5 минуты.

Согласно приведенным графикам Рис.8. и Рис.10. операция по обновлению модуля «FM.NSS» длилась около 1 минуты.

Добавление манифеста «Bigmodule».

Запрос на добавление модуля в обоих случаях завершился ошибкой 504 Gateway Time-out. При этом модуль весом 246 мегабайт загружался и начинал обрабатываться.

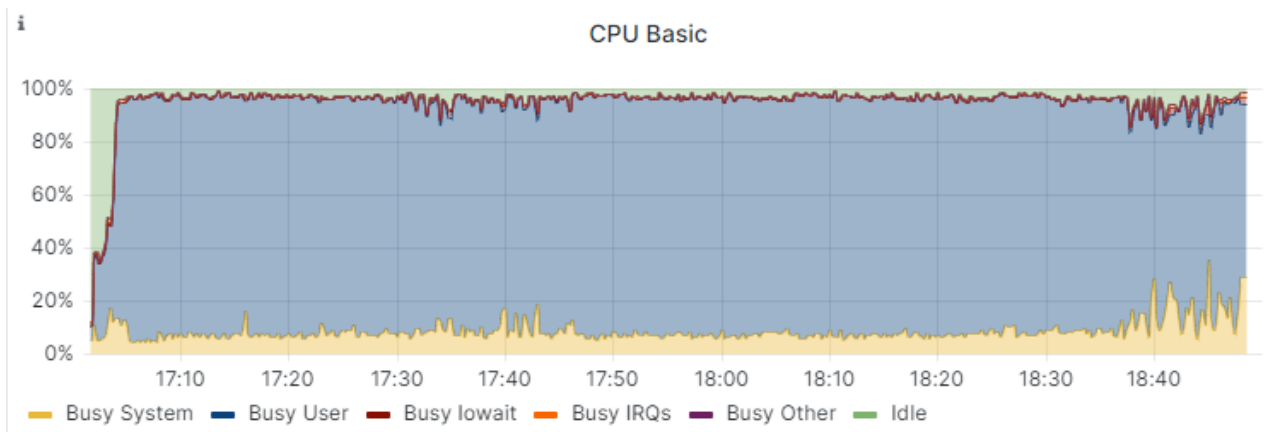


Рис.11. Утилизация CPU при добавлении «Bigmodule» до оптимизаций.

На всем протяжении теста утилизация процесса практически 100%

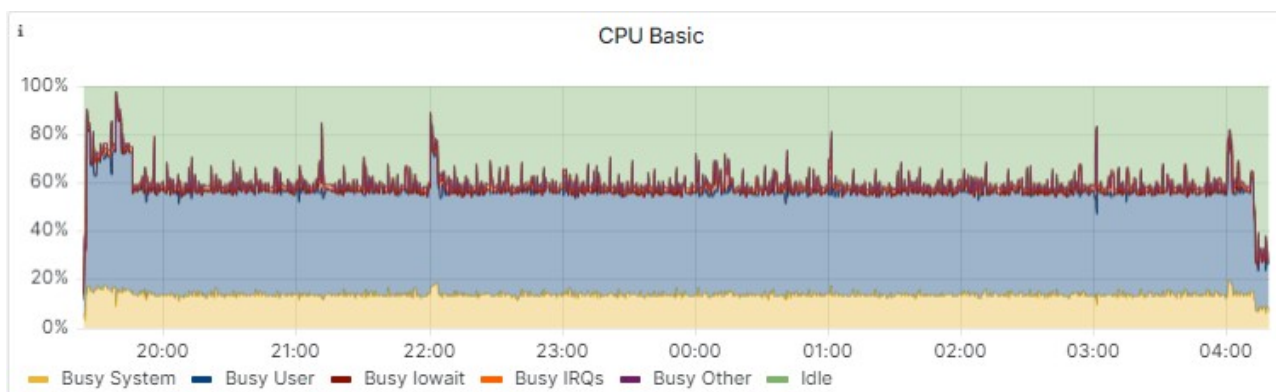


Рис.12. Утилизация CPU при добавлении «Bigmodule» после оптимизаций.

После старта теста утилизация процессора повышается до 80%, затем снижается до 60% и держится стабильной.

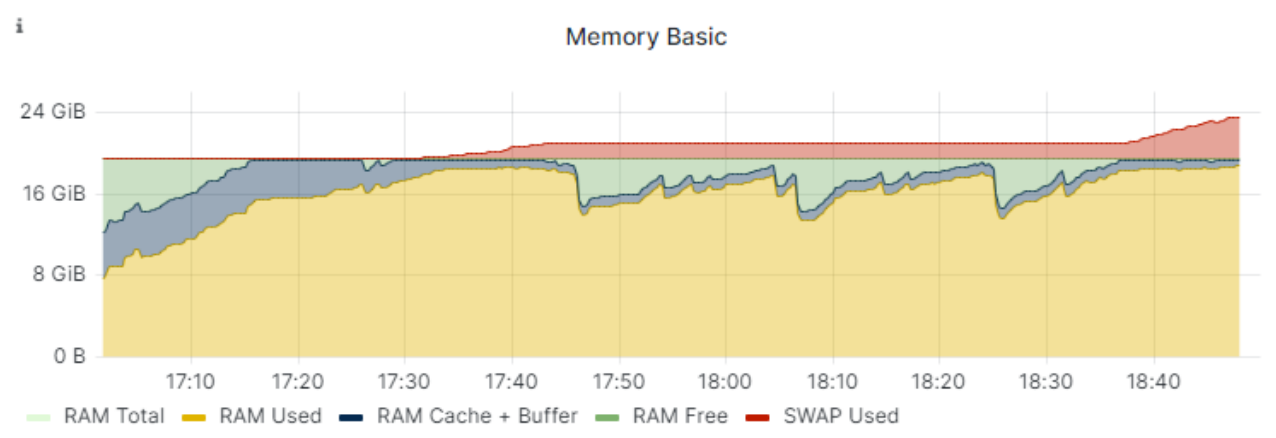


Рис.13. Утилизация RAM при добавлении «Bigmodule» до оптимизаций.

Наблюдается рост утилизации памяти, затем периодическое ее очищение, вероятно вследствие срабатывания GC, рост потребления SWAP памяти, утилизации дисковой подсистемы и отказ стенда. Операция по добавлению модуля «Bigmodule» до оптимизаций до конца не выполнялась.

i

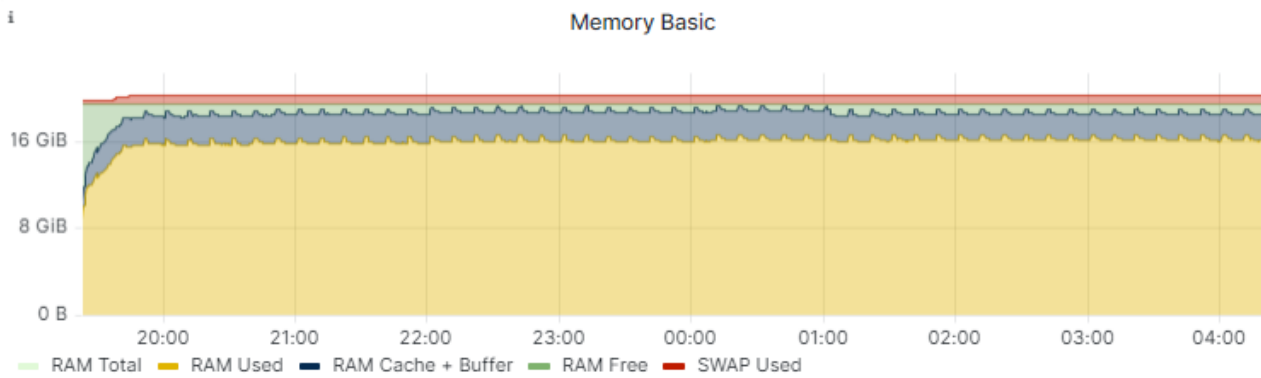


Рис.14. Утилизация RAM при добавлении «Bigmodule» после оптимизаций.

Память утилизируется, но периодические сбрасывания GC, не приводят к активной утилизации SWAP памяти.

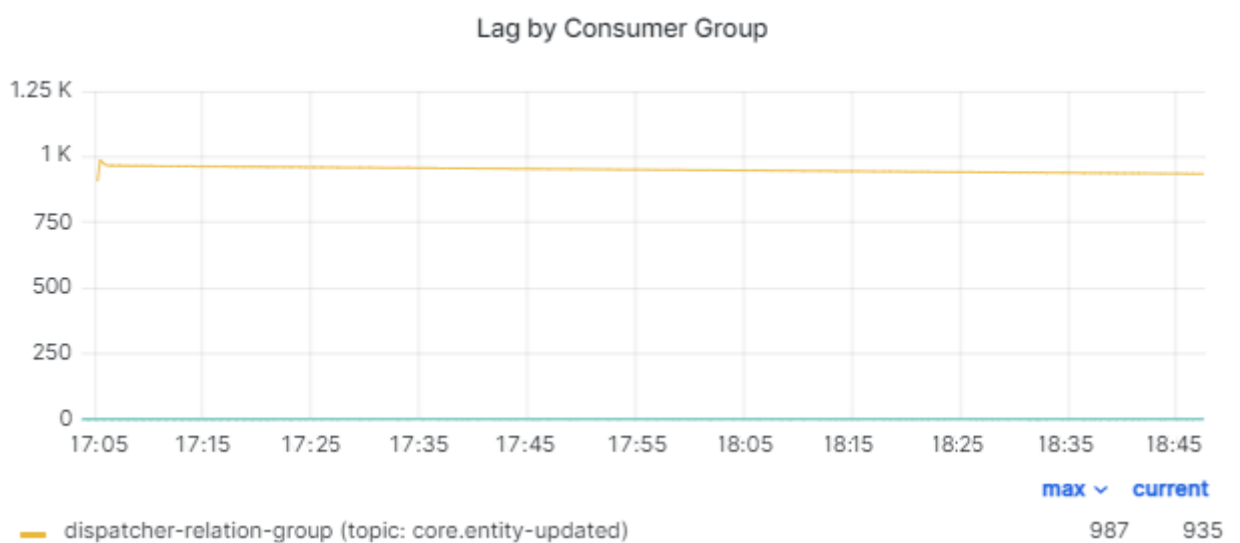


Рис.15. Динамика обработки очереди в kafka при добавлении «Bigmodule» до оптимизаций.



Рис.16. Динамика обработки очереди в kafka при добавлении «Bigmodule» после оптимизаций.

Согласно приведенным графикам Рис.11. и Рис.16. операция по обновлению модуля «Bigmodule» длилась 8 часов 52 минуты.