

Тестирование производительности middleware фреймворков для разработки программного обеспечения мобильных роботов

Студент группы 2304
Руководитель

Ефремов Михаил Александрович
к.т.н. Кринкин Кирилл Владимирович

Цели и задачи

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования
Факторы
производительности

Реализация

Конфигурация
Бенчмарки
Анализатор
Система

Результаты
тестирования

ROS
MIRA
YARP
Сравнение
Итоги

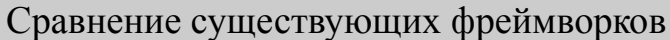
Заключение

Цель данной работы: получение и анализ результатов тестирования производительности для наиболее доступного и используемого многоагентного робототехнического промежуточного ПО.

Решаемые задачи:

- Рассмотрение и отбор фреймворков для тестирования
- Анализ методов коммуникации и формирование план тестирования
- Выбор инструментов для реализации тестирования и реализация тестовых сценариев
- Получение, обработка и анализ данных тестирования производительности, составление отчета

- Рост исследований и разработок в области автономных систем
- Множество различных платформ для разработки робототехнического ПО
- Малое количество исследований производительности существующих решений



Заключение

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻ 4/24

Сравнение реализаций отобранных для тестирования фреймворков

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
Бенчмарки
Анализатор
Система

Результаты
тестирования

ROS
MIRA
YARP

Сравнение
Итоги

Заключение

	ROS	MIRA	YARP
Централизованные сервисы	Сервисы поиска, именования, сервис параметров	Нет	Сервис имен
Взаимодействие между узлами	Топики, параметры, сервисы	Каналы, RPC	Порты, топики
Протоколы коммуникации	TCP, UDP, собственный протокол roserial	IPC, TCP	ACE, TCP, UDP, IPC
Формат сообщений	Бинарный	Бинарный, XML, JSON	Бинарный

Факторы производительности

Общий фактор:

Размер сообщений	1Kб	4Kб	16Kб	64Kб	256Kб
	1Mб	4Mб	16Mб	64Mб	

ROS:

Размер буфера	1	10	100	1000	10000
Количество подписчиков	1	2	4	8	
Способ взаимодействия	Издатель-подписчик		Сервис-клиент		

MIRA:

Локализация модуля	Один процесс	Различные процессы
Способ взаимодействия	Каналы	RPC

YARP:

Тип порта	С буфером	Без буфера	RPC	
Протокол	TCP	UDP*	FastTCP	ShMem

Конфигурация среды тестирования производительности

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

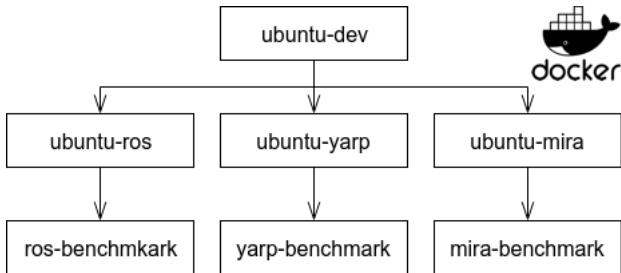
Итоги

Заключение

Тесты проводились на следующей аппаратной конфигурации:

- 8 процессоров Intel Xeon E5-2580 2.4 ГГц;
- 64 Гб оперативной памяти.

Операционная система: Ubuntu 16.04. Для тестирования был установлен Docker версии 18.04.0-ce.



Требования к инструменту измерения производительности

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Измеряемые характеристики:

- Задержка передачи сообщений между зулами (нс)
- Полоса пропускания канала коммуникации (б/с)

Требования к инструменту:

- Точность вычисления:
 - Высокоточные часы `std::chrono`
 - Такты процессора `asm rdtsc`
- Сериализация результатов в файл
- Возможность приостанавливать учет времени
- Итеративность выполнения тестов

Решения:

- Google Benchmark framework
- Собственная библиотека Benchmark gripper (причина: MIRA)

Benchmark Gripper: сценарии использования и диаграмма состояний

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

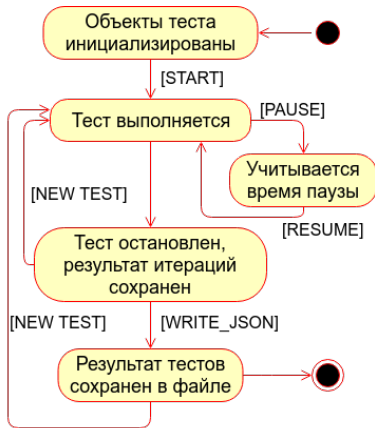
Реализация

Конфигурация
Бенчмарки
Анализатор
Система

Результаты
тестирования

ROS
MIRA
YARP
Сравнение
Итоги

Заключение



Benchmark Gripper: диаграмма классов

MARM
 Performance
 Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

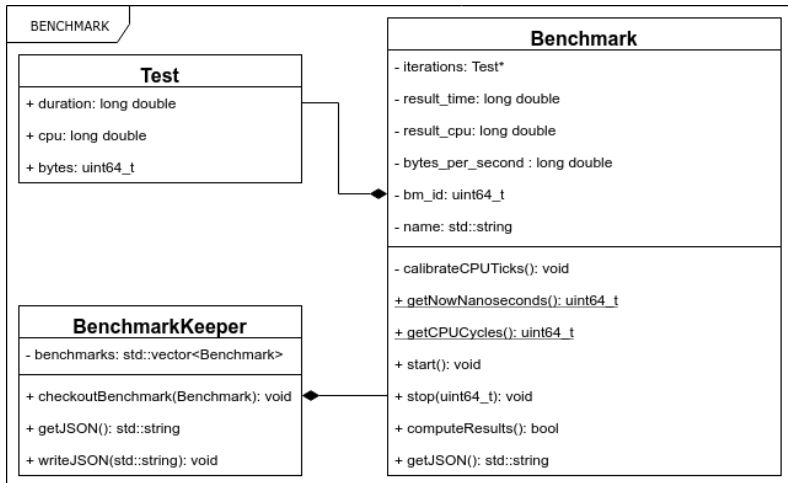
MIRA

YARP

Сравнение

Итоги

Заключение



Benchmark Gripper: преимущества и недостатки

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Преимущества:

- не требуется предварительная компиляция
- интерфейс основан на макросах
- высокая точность
- сериализация результатов совместимая с Google Benchmark

Недостатки:

- скудный по-сравнению с Google benchmark функционал
- длительное исполнение тестов

Анализатор результатов - Merger.py

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

■ Принимает на вход:

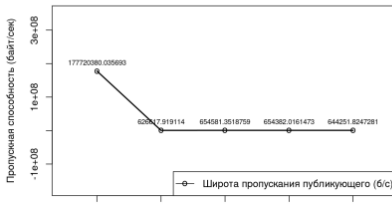
- множество результатов в формате json
- файлы конфигурации

■ Статистически обрабатывает результаты

■ Результат - файл rmarkdown

```
result.html Open in Browser Find Publish
ра", xlab="Объем данных", ylab="Пропускная способность (байт/сек)"),
text( factor(size, levels=size), bytes_per_second,
labels=bytes_per_second, cex= 0.7, pos=3)
legend("bottomright", legend=c("Широта пропускания
публикующего (б/с)"), col=c("black"), lty=c(1), pch
=c("o"), ncol=1)
```

Зависимость широты пропускания от фактора



Система тестирования производительности

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

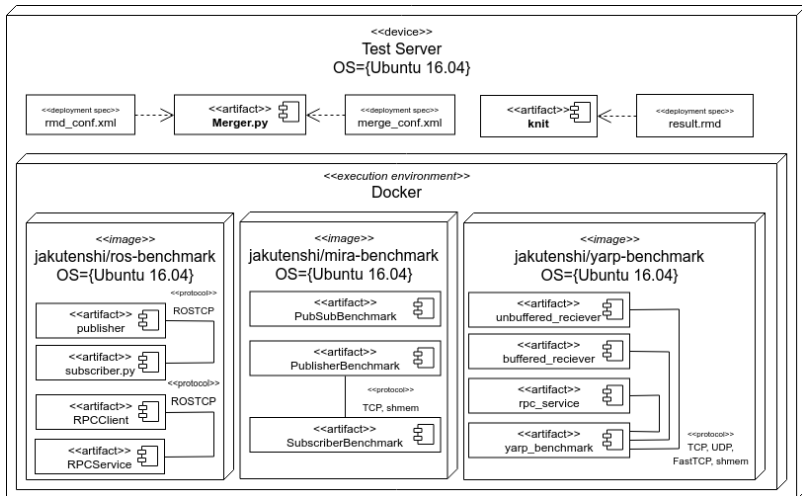
MIRA

YARP

Сравнение

Итоги

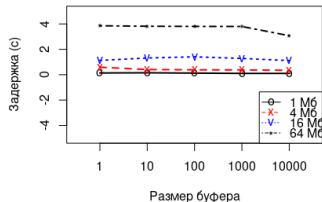
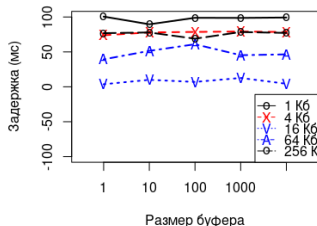
Закключение



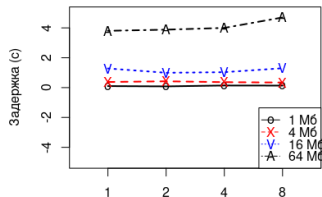
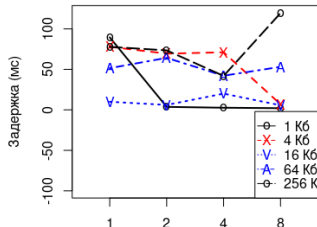
ROS - влияние различных факторов

MARM
Performance
Testing

Размера буфера:



Количество подписчиков:



ROS - паттерн «Издатель-подписчик»

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

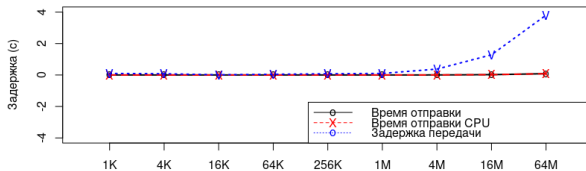
YARP

Сравнение

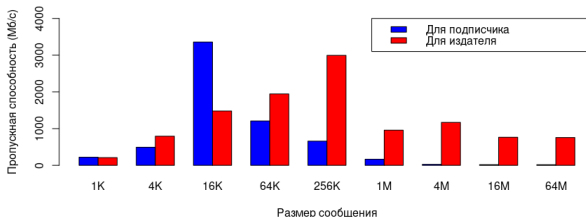
Итоги

Закключение

Задержка при передачи сообщений:

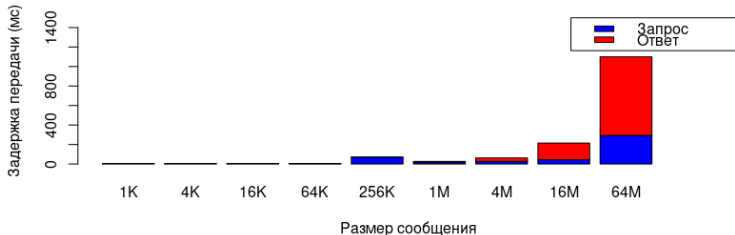


Пропускная способность:

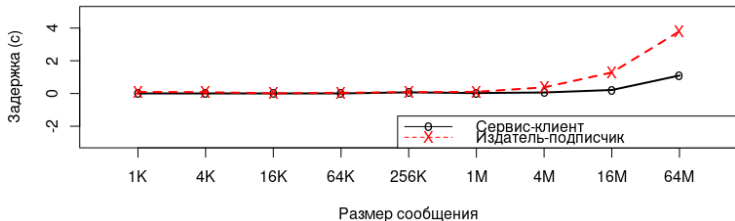


ROS - паттерн «Клиент-сервис»

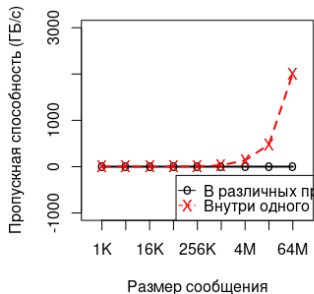
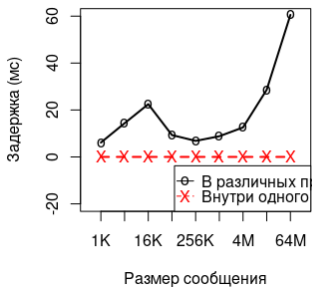
Задержка передачи запроса и ответа:



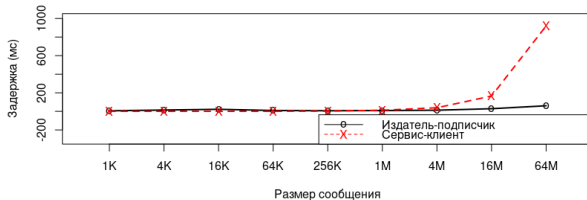
Сравнение с подходом "издатель-подписчик":



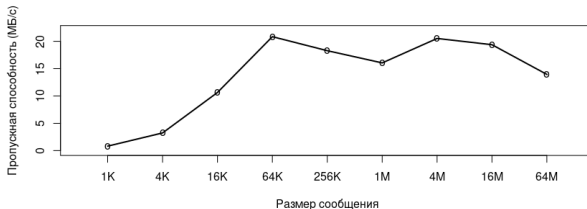
MIRA - зависимость от расположения модулей



Сравнение задержки передачи данных с каналами:



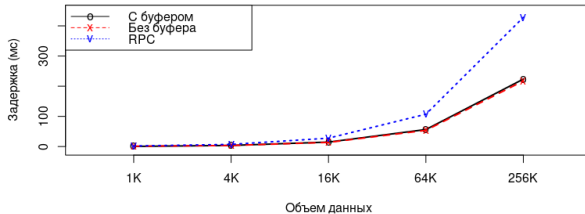
Пропускная способность:



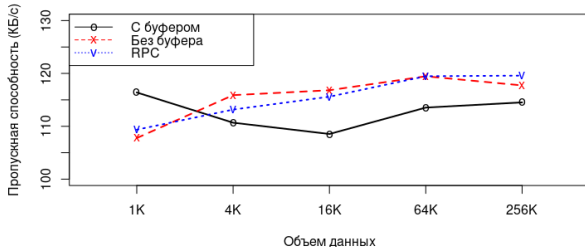
YARP - сравнение различных типов портов

MARM
Performance
Testing

Сравнение задержки передачи данных:



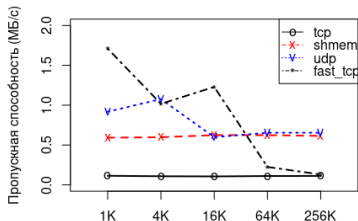
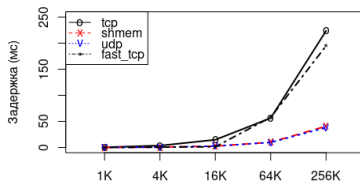
Сравнение пропускной способности:



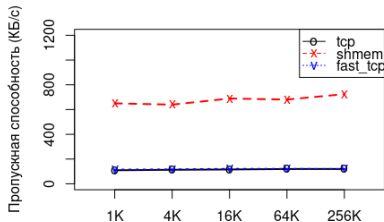
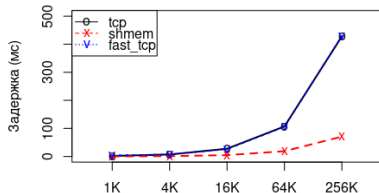
YARP - сравнение различных протоколов

MARM
Performance
Testing

Для буферизованного порта:



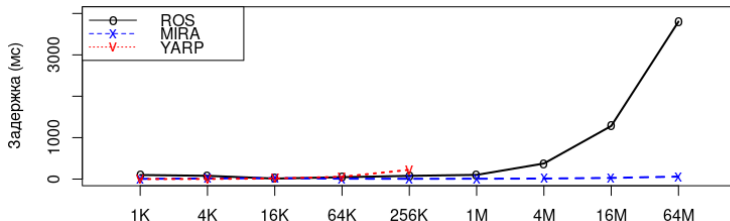
Для RPC-порта:



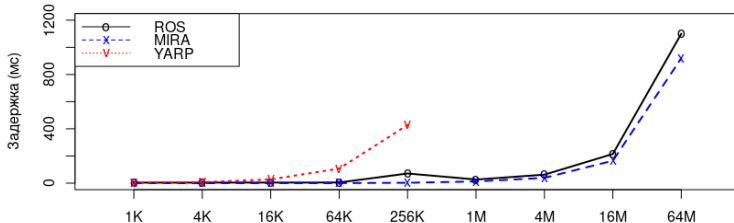
Сравнение производительности фреймворков

MARM
Performance
Testing

Передача данных по TCP для различных процессов:



Производительность реализаций RPC:



Итоги тестирования производительности

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Таким образом, результаты тестирования выявили следующие факты:

- на производительность ROS не влияет ни количество подписчиков, ни размер буфера
- производительность ROS резко падает при передачи больших объемов данных
- YARP имеет наихудшие показатели производительности
- в YARP рекомендуется использовать разделяемую память на основе ACE
- MIRA имеет наилучшие показатели производительности

Заключение

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Таким образом, в ходе работы:

- Было проведено исследование такой предметной области, как промежуточное ПО для разработки прикладных систем автономных роботов
- Был составлен список факторов, которые гипотетически могли влиять на производительность
- Был составлен и реализован план тестирования в ходе которого:
 - Была реализована собственная библиотека для разработки бенчмарк-тестов
 - Были определены и реализованы тестовые случаи
 - Был реализован анализатор результатов тестирования производительности
- Был проведен анализ результатов тестирования и оформлен отчет по исследованию

Исходный код и материалы: https://github.com/JAkutenshi/master_degree_repo

Статья: Ефремов М.А. Проблематика производительности многоагентных фреймворков для разработки ПО автономных роботов / сборник докладов студентов, аспирантов и молодых ученых 71-й научно-технической конф. профессорско-преподавательского состава университета., СПб, 29 янв.-- 9 фев. 2018 г. / СПбГЭТУ «ЛЭТИ», СПб, 2018, с. 99-102.