

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
окружения

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 (МАРППО).

Решаемые задачи:

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

- 4я промышленная революция и автоматизация процессов;
- Множество различных платформ для разработки робототехнического ПО;
- Практически отсутствие исследований производительности существующих решений.



Заключение

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻ 4/23

Технические особенности отобранных фреймворков

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

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	Топики, 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

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Общий фактор:

Размер сообщений	1 Кб	4 Кб	16 Кб	64 Кб	256 Кб	1 Мб	4 Мб	16 Мб	64 Мб
------------------	------	------	-------	-------	--------	------	------	-------	-------

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

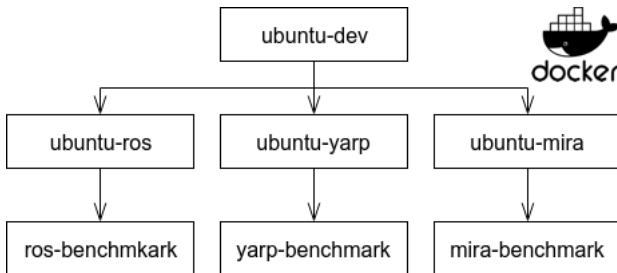
Конфигурация окружения

MARM
Performance
Testing

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.



Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Требования к инструменту измерения

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

- Точность вычисления:
 - Высокоточные часы `std::chrono`
 - Такты процессора `asm rdtsc`
- Сериализация результатов
- 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
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

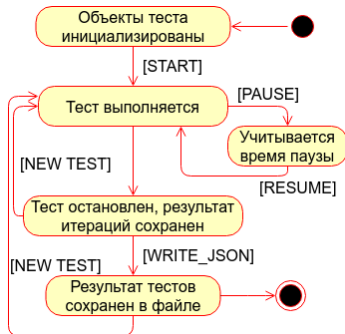
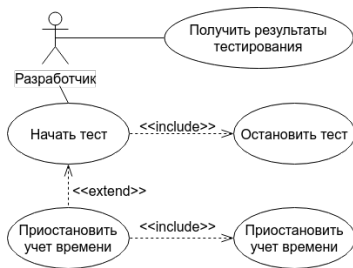
MIRA

YARP

Сравнение

Итоги

Заключение



Benchmark Gripper: диаграмма классов

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

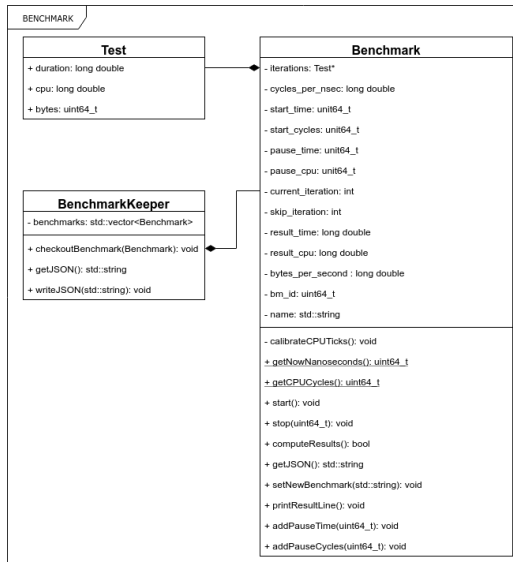
MIRA

YARP

Сравнение

Итоги

Заключение



Benchmark Gripper: преимущества и недостатки

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

YARP

Сравнение

Итоги

Заключение

Преимущества:

- библиотека реализована только на заголовочных файлах, не требуется предварительная компиляция;
- интерфейс использования реализован при помощи макросов, использование которых приближено к методу сэмплирования в профилировщиках;
- тесты выполняются фиксированное количество итераций;
- первые итерации отбрасываются для повышения точности результата;
- сериализация результатов тестирования в JSON файл, совместимый с Google Benchmark.

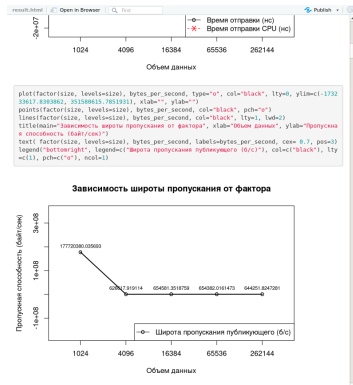
Недостатки:

- скудный по-сравнению с Google benchmark функционал;
- возможно длительное выполнение из-за фиксированного количества итераций.

Анализатор результатов - Merger.py

MARM
Performance
Testing

Анализатор принимает на вход множество json-файлов с результатами тестов и выдает markdown файл, который при помощи транслятора knitr модно преобразовать в html файл для удобного анализа результатов тестирования производительности



Система тестирования производительности

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

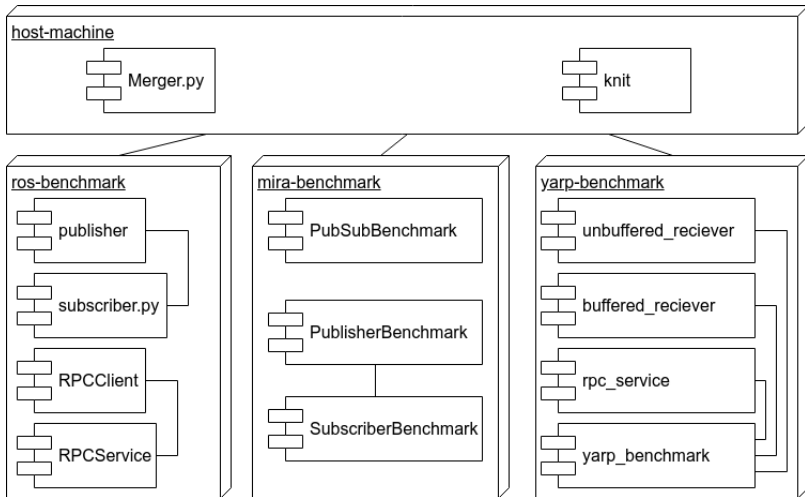
MIRA

YARP

Сравнение

Итоги

Заключение



ROS - влияние различных факторов

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

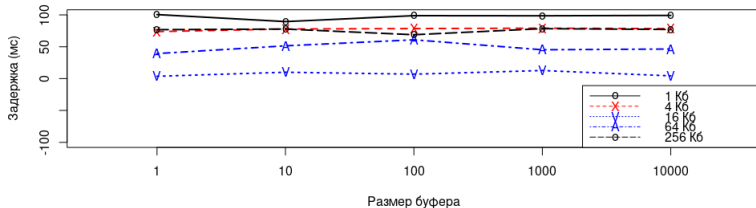
YARP

Сравнение

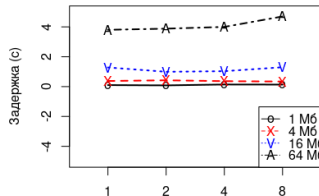
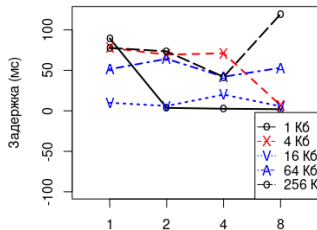
Итоги

Заключение

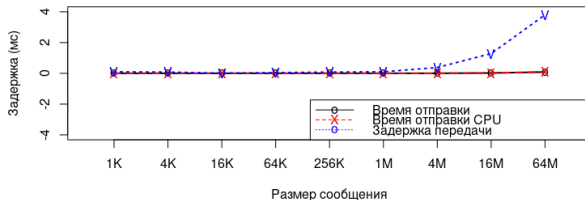
Размера буфера:



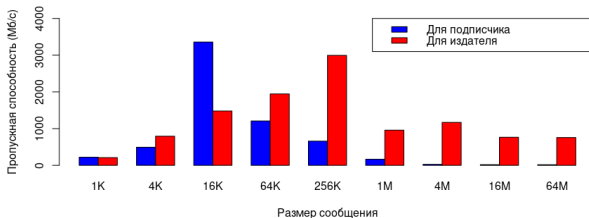
Количество подписчиков:



Задержка при передаче сообщений:



Пропускная способность:



ROS - Клиент-сервис

MARM
Performance
Testing

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

MIRA

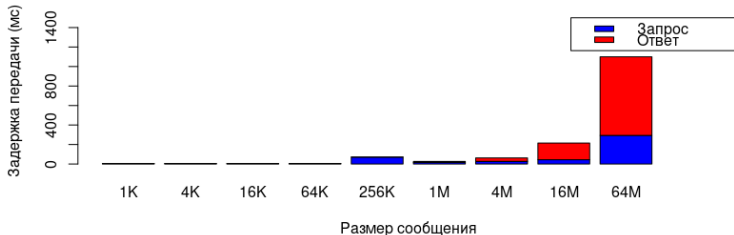
YARP

Сравнение

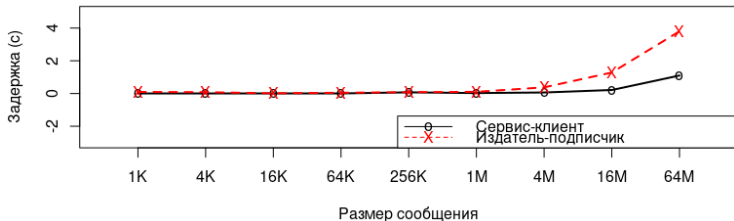
Итоги

Заключение

Задержка передачи запроса и ответа:



Сравнение с подходом "издатель-подписчик":



MIRA - зависимость от расположения модулей

MARM Performance Testing

Введение

Планирование
тестирования

Предмет
тестирования
Факторы
производительности

Реализация

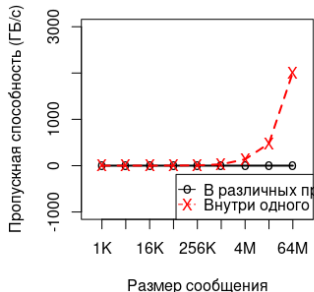
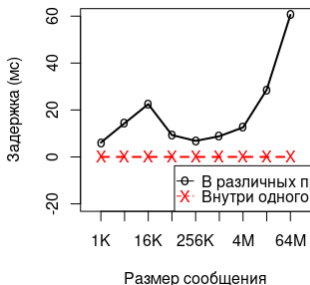
Конфигурация
окружения
Бенчмарки
Анализатор
Система

Результаты
тестирования

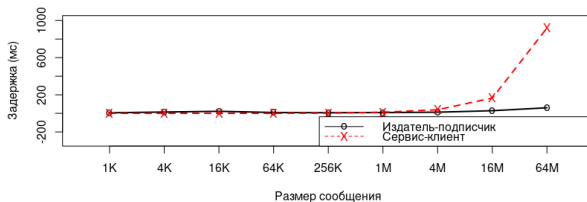
ROS
MIRA

YARP
Сравнение
Итоги

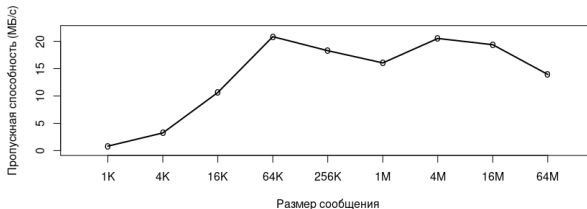
Заключение



Сравнение задержки передачи данных с каналами:



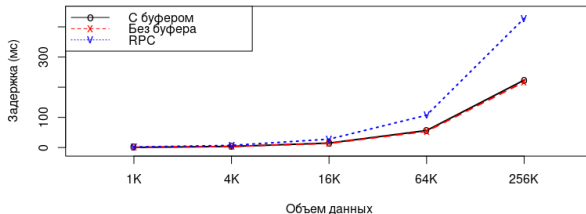
Пропускная способность:



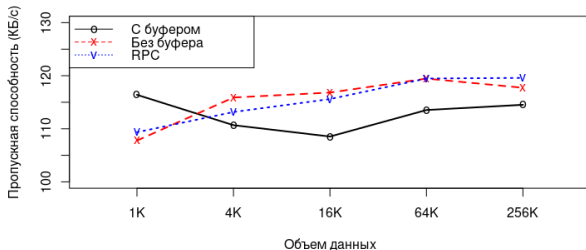
YARP - сравнение различных типов портов

MARM
Performance
Testing

Сравнение задержки передачи данных:



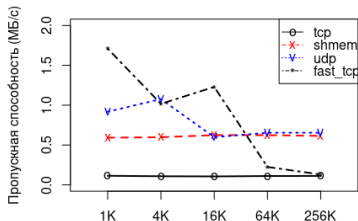
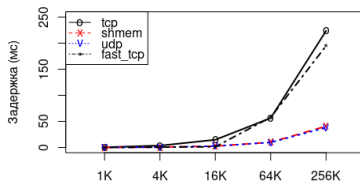
Сравнение пропускной способности:



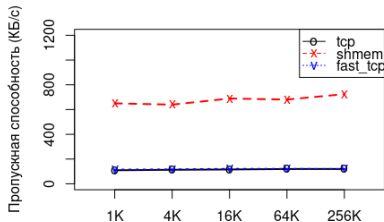
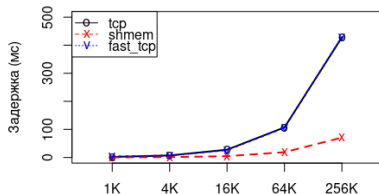
YARP - сравнение различных протоколов

MARM
Performance
Testing

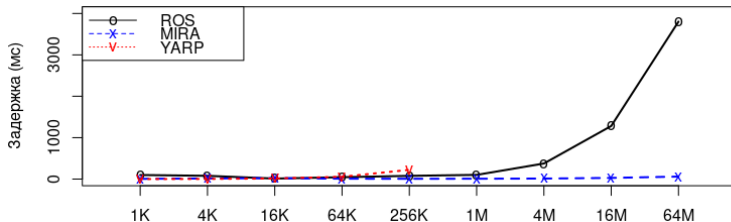
Для буферизованного порта:



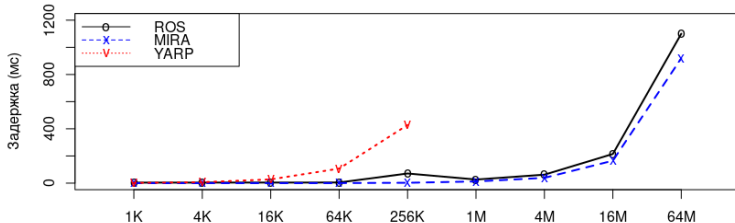
Для RPC-порта:



Передача данных по TCP для различных процессов:



Производительность реализаций RPC:



Итоги тестирования производительности

MARM
Performance
Testing

Таким образом, результаты тестирования выявили следующие факты:

- на производительность ROS не влияет ни количество подписчиков, ни размер буфера сообщений (но это сильно сказывается на объеме потребляемой оперативной памяти);
- производительность ROS резко падает при передаче больших объемов данных;
- YARP имеет наихудшие показатели производительности;
- при разработке прикладного ПО для автономных роботов на платформе YARP рекомендуется использовать разделяемую память на основе фреймворка ACE;
- MIRA имеет наилучшие показатели производительности при наименьшем потреблении системных ресурсов.

Введение

Планирование
тестирования

Предмет
тестирования

Факторы
производительности

Реализация

Конфигурация
окружения

Бенчмарки

Анализатор

Система

Результаты
тестирования

ROS

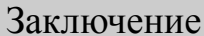
MIRA

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