

Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина)

MARM Performance Testing

Введени

Планировани

Предмет

Факторы

Реализапи

1 0001113011111

окружен

Бензыарки

Систел

Результаты

тестировані

ROS

YARP

Сравне

Заключени

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

Ввеление

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.



Актуальность

MARM Performance Testing

Введение

Планирования тестирования

тестирован

Факторы

Doorwoorwa

Конфигураци

Бешимоп

Анализато

Результаты

тестировани

ROS

YARP

Сравнен

Заключени

 4я промышленная революция и автоматизация процессов;

- Множество различных платформ для разработки робототехнического ПО;
- Практически отсутствие исследований производительности существующих решений.



Сравнение существующих решений

MARM Performance Testing

Ввелени

Планирования тестирования

Предмет тестирования

Факторы производительност

производительності

Конфигурация

Бенчмарки Анализатор

Система

Результаты тестирования

ROS MIRA YARP

Сравнен Итоги

Заключени

НАЗВАНИЕ	ОТКРЫТЫЙ КОД	ДОКУМЕН- ТАЦИЯ	ИЗМЕНЕ- НИЯ	АРХИТЕКТУРА	пк	
ROS	Да	Да	2018	Гибридная	C++, Python	
MIRA	Да	Да	2018	Децентрализованная	C++, Python, JavaScript	
YARP	Да	Да	2018	Децентрализованная	C++, Python, Java, Octave	
MOOS	Да	Да	2018	Централизованная	C++, Java	
OROCOS/Rock	Да	Да	2016	Гибридная	C++, Python, Simulink	
ASEBA	Да	Нет	2018	Распределенная	Собственный язык	
OpenRTM- aist	Да	Нет	2016	Гибридная	C++, Java, Python	
URBI	Да	Нет	2016	Централизованная	C++, Java, urbiscript	



Технические особенности отобранных фреймворков

MARM Performance Testing

Введени

Планирования Тестирования Предмет

тестирования Факторы

производительност

Реализация

Конфигураці окружения

Бенчмарки Анализатор

Система

тестировани:

MIRA

Сравнени

Заключение

	ROS	MIRA	YARP
Централизованные сервисы	Сервисы по- иска, имено- вания, сервис параметров	Нет	Сервис имен
Взаимодействие между узлами	Топики, пара- метры, серви- сы	Топики, RPC	Порты, топики
Протоколы ком- муникации	TCP, UDP, собственный протокол rosserial	IPC, TCP	ACE, TCP, UDP, IPC
Формат сообще- ний	Бинарный	Бинарный, XML, JSON	Бинарный



Факторы производительности

MARM Performance Testing Общий фактор:

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

Введение

Планирования

тестирования Факторы

производительности

.

Реализаци

Конфигураци:

Бенчмаркі

Система

Результаты тестировані

тестирован: ROS

MIRA YARP

Сравнені Итоги

Заключение

ROS:

Размер буфера	1	10	100 1000 10000		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

Введение

Планирования тестирования

тестирования

производительнос

Реализаци Конфигурация

окружения

Анализатор

Результаты

ROS MIRA

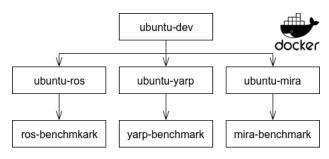
Сравнение

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-се.





Требования к инструменту измерения

MARM Performance Testing

Введени

Планирования тестирования

тестирования

производительнос

Реализаци

Конфигураци

окружения **Бенчмарки**

Система

тестировани тестировани

MIRA

Сравнен

Заключени

■ 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

Введени

Планирования

Презмет

Факторы

производительн

Реализаци

Y----

окружения

Бенчмарки

Систе

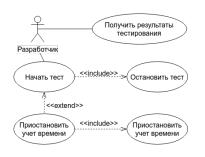
Результаты тестировани

ROS MIRA

Спавне

Сравнен Итоги

Заключени







Benchmark Gripper: диаграмма классов

MARM Performance Testing

Врадания

Планирования

Предмет

Факторы

Реализаци

Конфигурация окружения

Бенчмарки

Анализатор

Систем

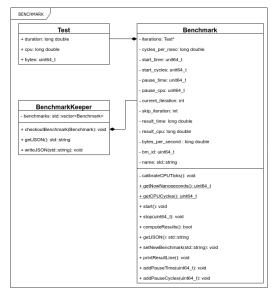
езультаты

ROS

VARP

Сравнені

Заключени





Benchmark Gripper: преимущества и недостатки

MARM Performance Testing

Бенчмарки

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

Введени

Планирования тестирования

тестирования

Факторы производительно

...

Конфигурац

Банныя

Анализатор

Результаты

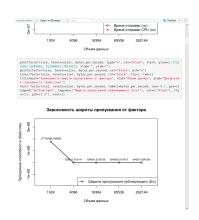
ROS

YARP

Сравнен Итоги

Заключени

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





Система тестирования производительности

MARM Performance Testing

Ввелени

Планирования

Предмет

Факторы

геализаци

октомения

Reuman

Анализато

Система

тестировани

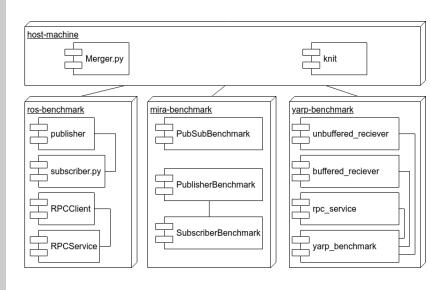
ROS

YARP

Сравне

7-----

Заключение



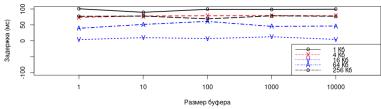


ROS - влияние различных факторов

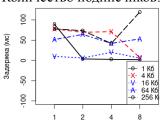
MARM Performance Testing

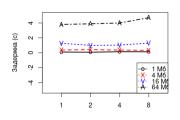
ROS

Размера буфера:



Количество подписчиков:







ROS - Издатель-подписчик

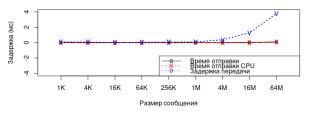
MARM Performance Testing

Ввеление

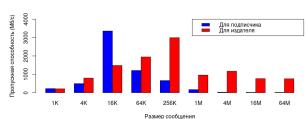
Результаты

ROS

Задержка при передачи сообщений:



Пропускная способность:





ROS - Клиент-сервис

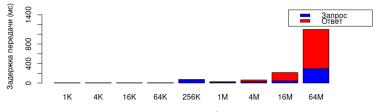
MARM Performance Testing

Ввеление

Результаты

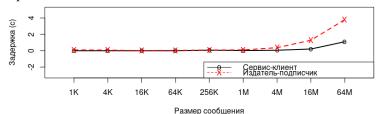
ROS

Задержка передачи запроса и ответа:



Размер сообщения

Сравнение с подходом "издатель-подписчик":





MIRA - зависимость от расположения модулей

MARM Performance Testing

Введение

Планирования

Предмет

Факторы

Реализапп

гсализаци

Конфигурац

Keumann

Анализато

Результаты

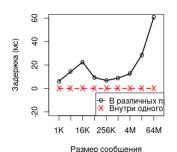
тести

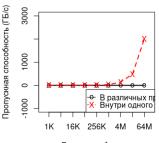
MIRA

Сравно

Сравнен Итоги

Заключени







MIRA - RPC

MARM Performance Testing

Ввеление

Планирования

Предмет

фина

производительнос

Реализаци

1 0001113011111

окружен

Cucres

Результаты

тестирован

ROS

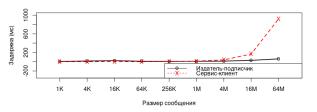
YARP

Сравне

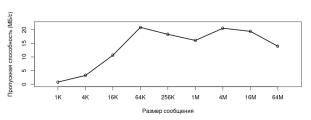
FIIOIN

Заключение

Сравнение задержки передачи данных с каналами:



Пропускная способность:





YARP - сравнение различных типов портов

MARM Performance Testing

Ввеление

Планирования

Предмет

Факторы

Реализапи

гсализаци

окружен

Система

Результаты тестирования

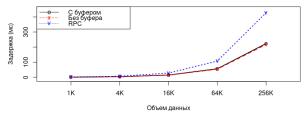
MIRA

YARP

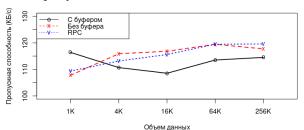
Сравнени Итоги

Заключение

Сравнение задержки передачи данных:



Сравнение пропускной способности:





YARP - сравнение различных протоколов

MARM Performance Testing

Ввелени

Планировани тестирования

Предмет

Факторы производительно

Реализаци

гсализаци

Конфигурац

окружен

4

C-----

Результаты тестирования

ROS

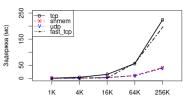
YARP

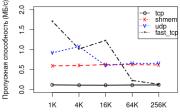
Сравнени

201111011011011

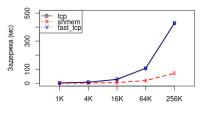
Заключение

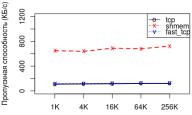
Для буферизованного порта:





Для RPC-порта:





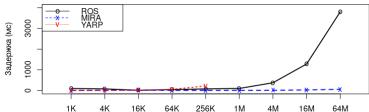


Сравнение

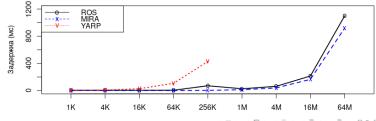
MARM Performance Testing

Сравнение

Передача данных по ТСР для различных процессов:



Производительность реализаций RPC:





Итоги тестирования производительности

MARM Performance Testing

Введени

Планирования тестирования

тестирования Факторы

Факторы производительное

Реализаці

Конфигураци

Бенчмарки Анализато

Анализатор Система

Результаты тестировани

ROS

Сравнен: **Итоги**

Заключение

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 передачи 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 АСЕ;
- MIRA имеет наилучшие показатели производительности при наименьшем потреблении системных ресурсов.



Заключение

MARM Performance Testing

Введени

Планирования тестирования

тестирования

Факторы произволительное

Реализаци

т силизици

отпостания

Бенчмарки

Анализато

Результаты

тестировани

MIRA

Сравнен 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