

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное  
образовательное  
учреждение высшего образования  
«Ижевский государственный технический университет  
имени М. Т. Калашникова»  
Факультет «Информатика и вычислительная техника»

Филиппов Алексей Николаевич  
Модели RPC в проектировании олимпиадного сервера  
Диссертация на соискание академической степени  
магистра

09.04.04 – Программная инженерия

09.04.04-1 – Разработка программно-информационных систем

Магистрант

Филиппов А.Н.

Научный руководитель  
к.т.н., профессор

Тарасов В.Г.

Ижевск 2016

# Цель работы

## Цель

Анализ и разработка технологии RPC для олимпиадной системы.

## Требования

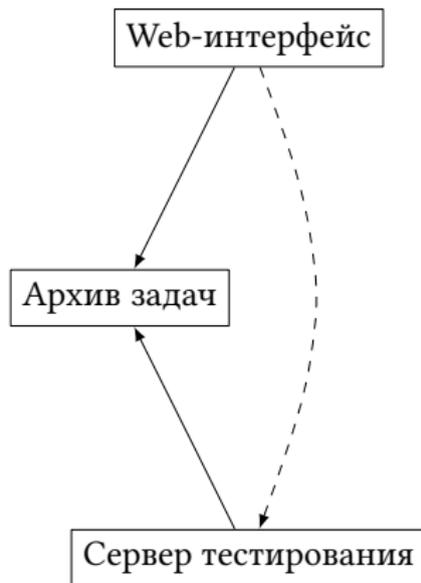
- Асинхронность
- Надёжность передачи данных
- Переносимость
- Производительность
- Персистентность

## Задачи работы

- Исследование моделей использования RPC в олимпиадной системе BACS, а также требований, предъявляемых к RPC в рамках каждой модели;
- исследование существующих технологий RPC и анализ их соответствия полученным моделям;
- разработка протокола RPC для олимпиадной системы;
- реализация протокола для языков программирования Go, Python и C#.

# Исследование моделей использования RPC

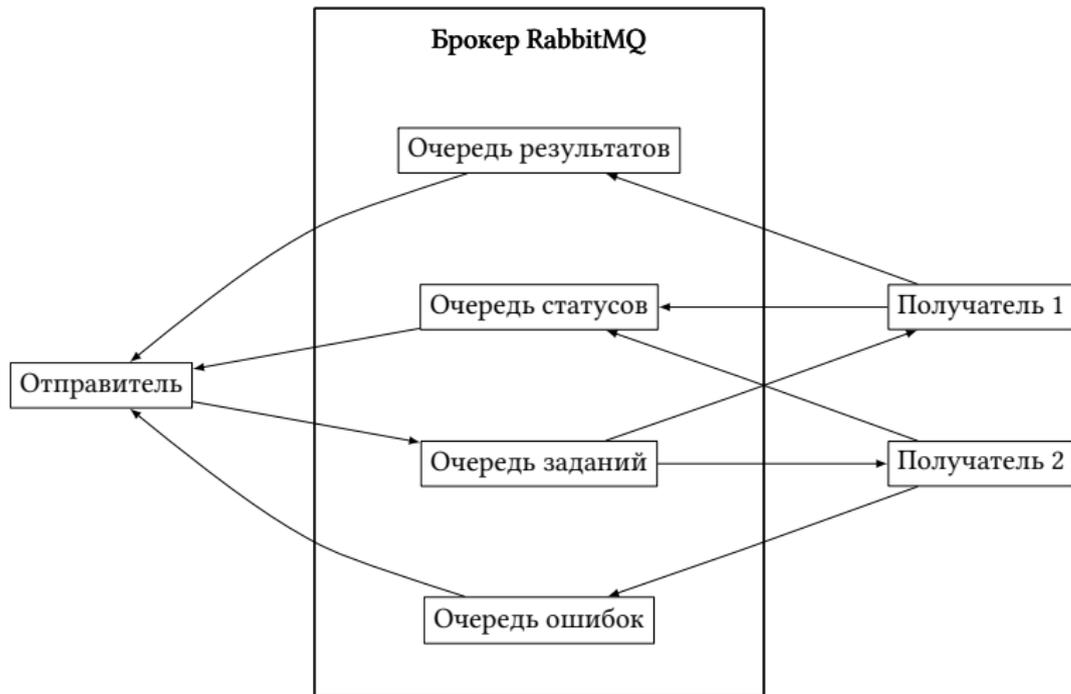
- Асинхронность
- Персистентность
- Переносимость
- Производительность



# Исследование технологий RPC

	Асинхронность	Персистентность	Переносимость	Производительность
XML-RPC	+	-	+	низкая
JSON-RPC	+	-	+	низкая
D-BUS	+	-	-	высокая
Java RMI	+	-	-	высокая
SOAP	+	-	+	низкая
Apache Thrift	+	-	+	высокая
gRPC	+	-	+	высокая

## Разработка протокола



# Реализация протокола

## Языки

- Go: [github.com/streadway/amqp](https://github.com/streadway/amqp)
- Python: `python-pika`
- C#: `RabbitMQ.Client`

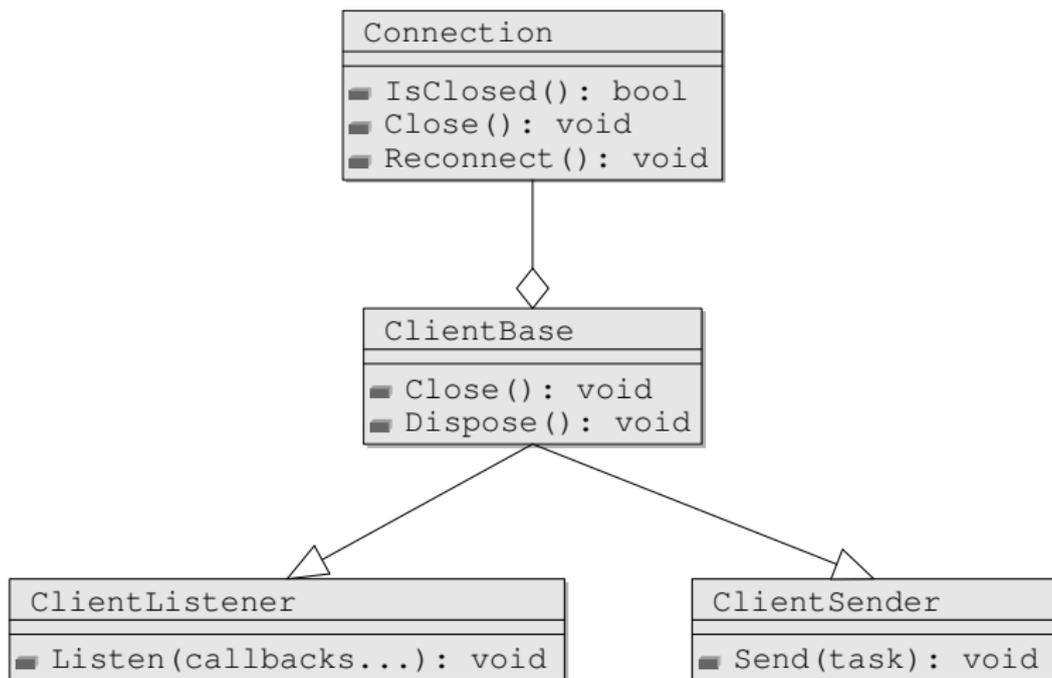
## Отправитель

- Подключение и мониторинг соединения с брокером
- Уведомления пользователя о результатах и промежуточных состояниях

## Получатель

- Асинхронная отправка промежуточных состояний
- Последовательная обработка заданий

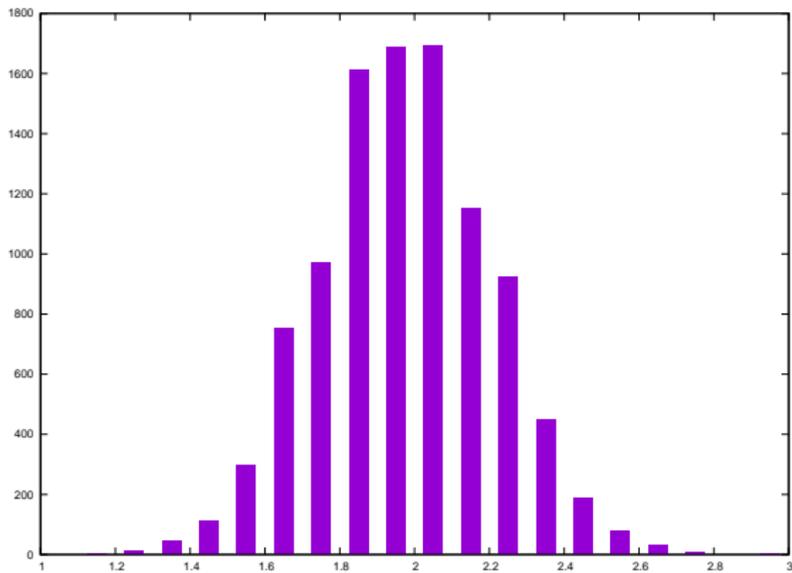
## Устройство библиотек



# Результаты

$$\mu = \frac{\sum_{i=1}^N t_i}{N} = 1.975 \text{ секунд}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (t_i - \mu)^2}{N}} = 0.229 \text{ секунд}$$



## Заключение

1. Обзор моделей и требований;
2. анализ существующих технологий, необходимость новой технологии;
3. разработка протокола RPC на основе очередей сообщений;
4. реализация протокола на языках Go, Python и C#.

Спасибо за внимание!

# Содержание

Цель работы

Задачи работы

Исследование моделей использования RPC

Исследование технологий RPC

Разработка протокола

Реализация протокола