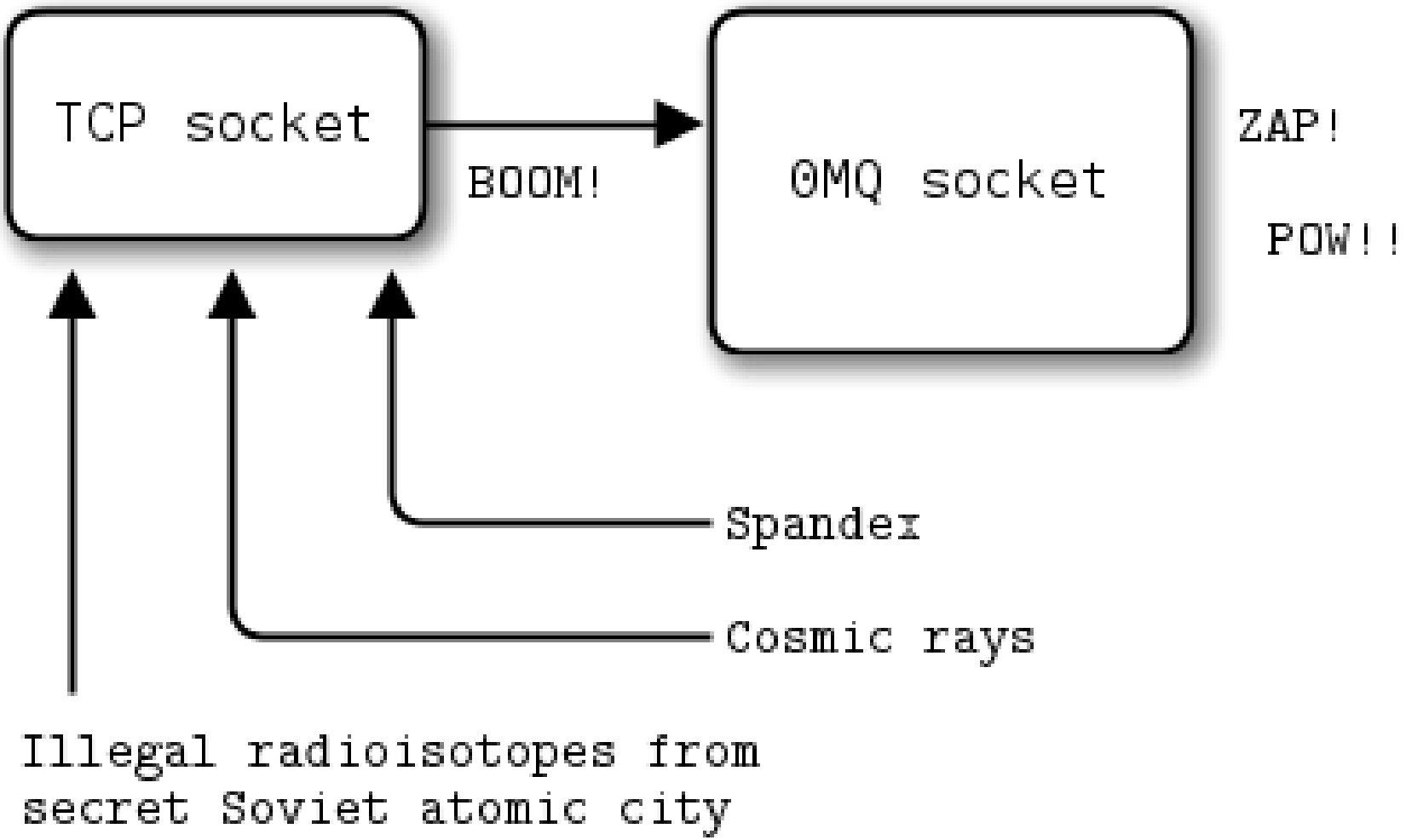
Распределённые объектные технологии: **Технология ZeroMQ**

Д. А. Усталов

УрФУ и ИММ УрО РАН

17 мая 2016 г.

Сетевое программирование



Сетевое программирование — это нетривиально.

Проблемы сетевого программирования

- Воспроизведение типовых шаблонов коммуникаций.
- Обеспечение отказоустойчивости.
- Шифрование трафика.
- Буферизация и асинхронный ввод-вывод.

Существующие решения

- Низкоуровневое программирование: `epoll()`, RIO, etc.
- Сетевые библиотеки: Boost.Asio, libuv, etc.
- Удалённый вызов процедур: ~~CORBA~~, Thrift, etc.
- Очереди сообщений: AMQP, JMS, etc.

Вопрос

Возможно ли обобщить этот опыт и создать удобную библиотеку для построения сетевых приложений?

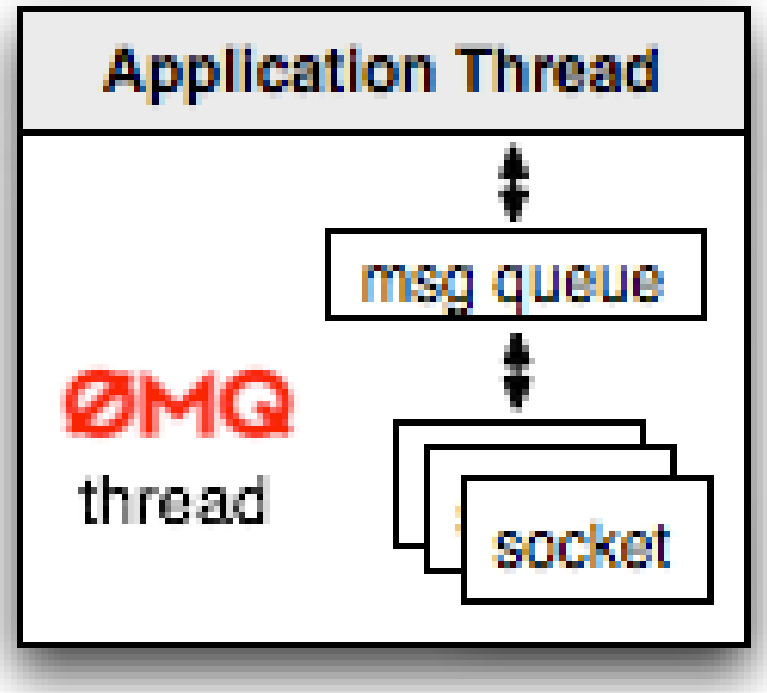
ZeroMQ

ZeroMQ (ØMQ) — высокопроизводительная библиотека обмена сообщениями.

- Не содержит брокера в явном виде, предлагает типовые элементы для построения распределённых систем.
- Программный интерфейс *очень напоминает* BSD-сокеты.
- Широкий спектр поддерживаемых операционных систем.
- “The Ø in ZeroMQ is all about tradeoffs.”

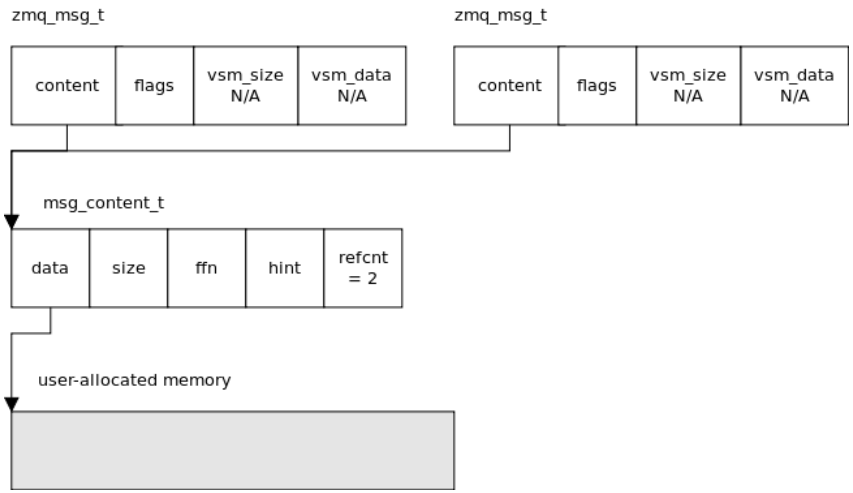
Принцип функционирования I

- При инициализации создаётся отдельная нить для сокетов и очереди сообщений.
- Все сообщения асинхронно передаются и принимаются через эту нить.
- Известны проблемы с GIL в языках сценариев.



Принцип функционирования II

- Возможность выбора транспортного протокола: TCP, UDP, PGM, IPC, UNIX-сокет, etc.
- Свой формат пакетов, инкапсулирующий передаваемые данные.
- Поддерживается шифрование.



Сокеты в ZeroMQ

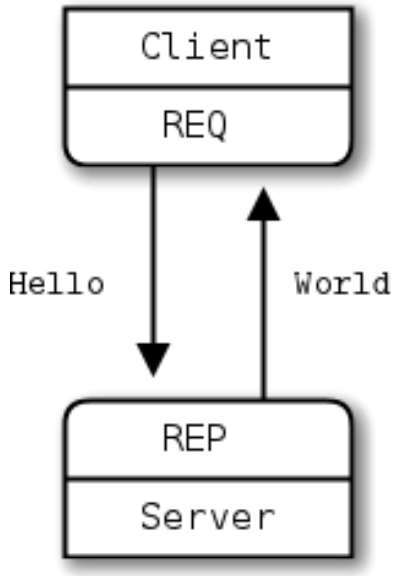
Сокеты в ZeroMQ различаются по типам:

- циклический перебор: REQ, PUSH, DEALER;
- многовещание: PUB;
- равноправные очереди: REP, SUB, PULL, DEALER;
- явная адресация: ROUTER;
- одноадресная передача: PEER.

http://www.slideshare.net/pieterh/overview-of-zeromq

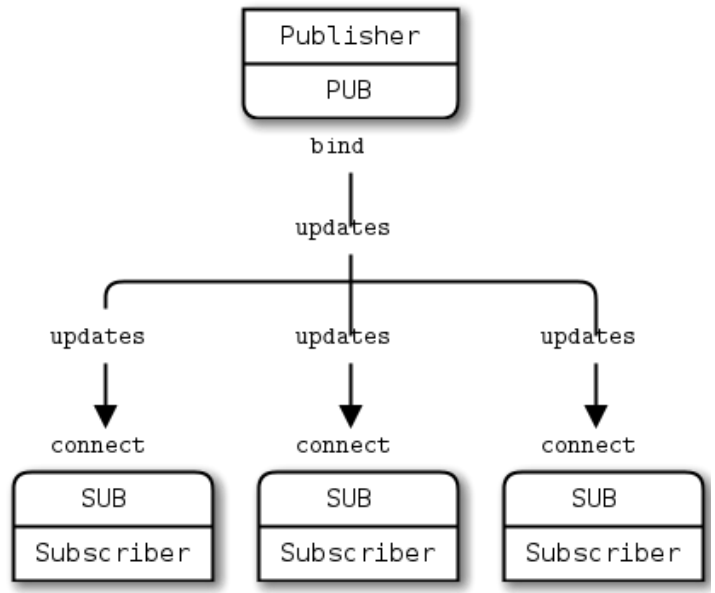
Сокеты в ZeroMQ: запрос-ответ

- Сервер ожидает подключения.
- Клиент подключается и передаёт запрос.
- Сервер выполняет запрос и возвращает результат.
- Клиент отключается.



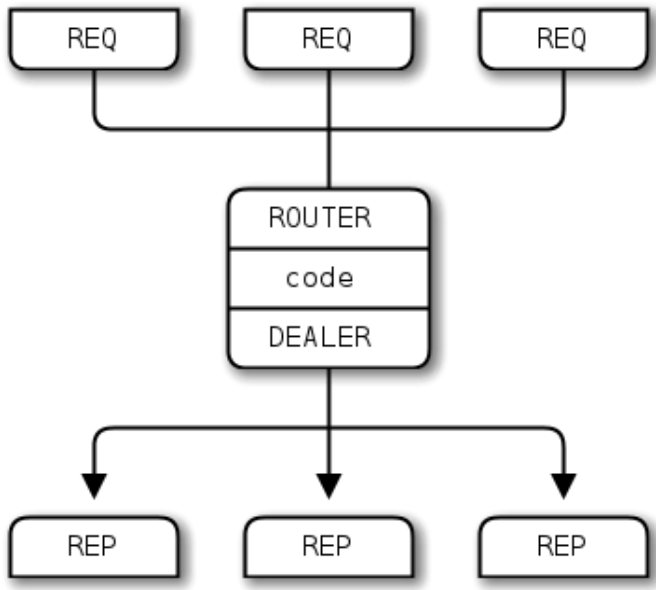
Сокеты в ZeroMQ: публикация-подписка

- Узел публикации ожидает подключения.
- Подписчик подключается и подписывается на сообщения.
- Узел публикации генерирует сообщение, которое рассылается подписчикам.
- Напоминает пример использования AMQP из прошлого занятия.



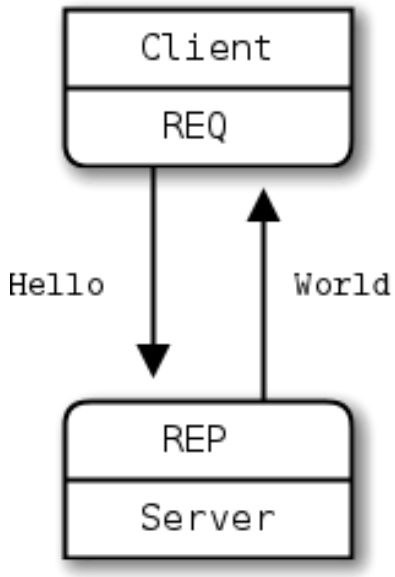
Сокеты в ZeroMQ: запрос-ответ с брокером

- Реализация шаблона «запрос-ответ» с использованием промежуточного узла (брокера).
- Брокер может осуществлять обработку данных, балансировку нагрузки, и т. д.



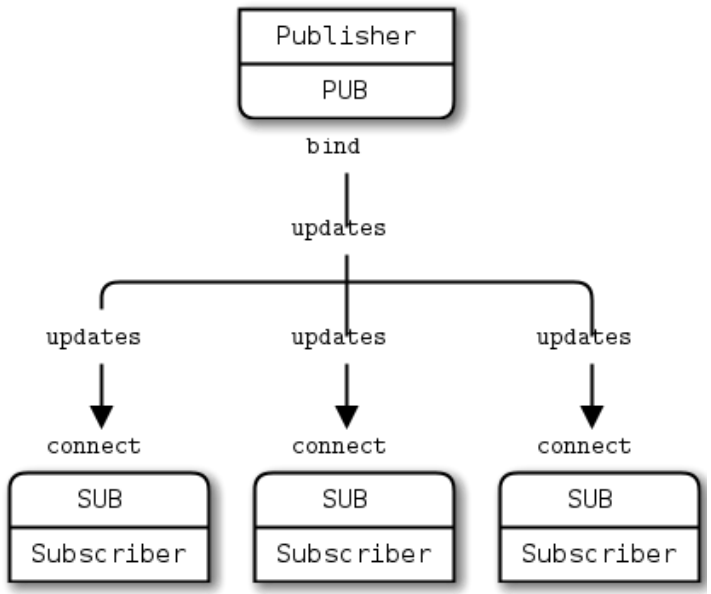
Демонстрация работы: запрос-ответ

- hwclient.py
- hwserver.py



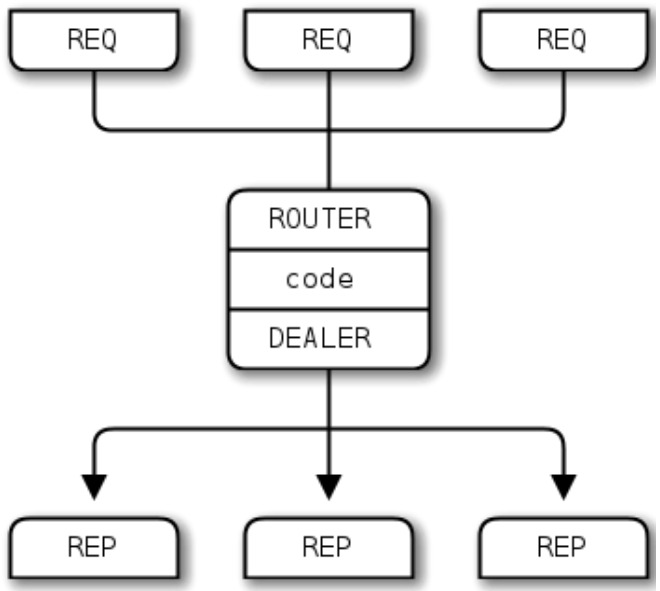
Демонстрация работы: публикация-подписка

- wuclient.py
- wuserver.py



Демонстрация работы: запрос-ответ с брокером

- rrclient.py
- rrserver.py
- rrbroker.py



Обсуждение

- ZeroMQ позволяет строить сложные распределённые системы из типовых элементов.
- Рассмтрено только небольшое подмножество всех возможных типов сокетов.
- **Важно:** сокеты ZeroMQ похожи на BSD-сокеты, но наделены *совершенно* иной семантикой.
- Истории успеха: Loggly и Samsung Chord (прекращён).
- Истории неуспеха: Apache Storm и MIMIMI GAMES.

Домашнее задание

Разработать веб-сервер с распределёнными рабочими процессами.

- Процесс *master* слушает порт 8080 по протоколу HTTP при помощи обычного TCP-сокета.
- Процесс *worker* подключается к *master* при помощи ZeroMQ и ждёт его команд.
- При поступлении HTTP-запроса, *master* извлекает из URI путь и передаёт его свободному *worker*.
- Свободный *worker* возвращает *master* содержимое запрошенного файла.
- Процесс *master* формирует HTTP-ответ.

Аналогичным образом устроен Mongrel2.

Спасибо за внимание!

Вопросы?

Дмитрий Усталов

https://linkedin.com/in/ustalov

http://kvkt.urfuclub.ru/courses/dot/

https://telegram.me/doturfu

dmitry.ustalov@urfu.ru