

# РАЗРАБОТКА АЛГОРИТМА КОММУТАЦИИ МЕЖДУ УЗЛАМИ СЕТИ «ИНТЕРНЕТ-ВЕЩЕЙ»

А. Скопченко

Томский политехнический университет

skop@tpu.ru

## Введение

На сегодняшний день существует большое количество задач, которые требуют высокопроизводительных вычислений, и с каждым днем их становится больше [1-3].

Современная концепция «интернет вещей» однозначно базируется на сети физических предметов, взаимодействующих с внешней средой и друг с другом. Важным компонентом в таких взаимодействиях являются технологии их коммуникации.

Данное исследование направлено на разработку способа коммутации (обмен данными) между компонентами сети «интернет вещей», не имеющих статических идентификаторов.

Для экспериментально-программной отладки разрабатываемого алгоритма коммутации, в качестве базовой инфраструктуры сети «интернет вещей» была выбрана инфраструктура информационно-вычислительного кластера на базе персональных компьютеров. Разрабатываемое программное обеспечение позволяет решить задачи обмена данными между компонентами кластера в ограниченных условиях (когда нет прямого доступа к узлам кластера).

## Исследование способов коммутации

Формирование инфраструктуры модельного кластера для проектирования алгоритма коммутации – достаточно трудоемкая задача. Одной из проблем, связанных с разработкой такого кластера, является выбор того, как именно будет работать кластер, а конкретнее, как будет реализована взаимодействие между его узлами. Так ли необходим центральный узел (узел управления), должен ли каждый узел выступать как узел управления и как компонент сети «интернет вещей» одновременно, должны ли узлы видеть «напрямую» другие узлы кластера или это необязательно. Необходимо ли, чтобы узлы имели возможность посылать запросы к узлу управления или они выступают только в роли ведомого (выполняют команду и возвращают результат).

В данном случае можно использовать два варианта коммутации, в зависимости от требований разработчика, но при этом имеющие схожий принцип работы. Первый вариант – это использование сервера коммутации для работы. Основное требование к серверу коммутации (в дальнейшем сервер-посредник) – это его видимость для всех компонентов кластера. Сервер-посредник, хранит таблицу коммутации с узлами, количество подключенных узлов, полную информацию об узлах (емкость жесткого диска, количество ядер процессора, объем доступной оперативной памяти и др.). В

связи с этим, другим узлам кластера не обязательно «видеть» друг друга и узел управления для выполнения своей работы. Сервер-посредник периодически опрашивает подключенные узлы – «живы» ли они. Это позволяет ему редактировать таблицу коммутации и держать её актуальной на текущий момент времени.

Если в процессе работы возникает ситуация, когда узел получил команду и внезапно «исчез», сервер-посредник возвращает ответ серверу, что данный узел недоступен и его последняя команда транслируется на другой свободный узел.

Второй вариант – это использование одной локальной сети, когда все участники кластера «видят» друг друга. В таком случае сервер-посредник хранит не только таблицу коммутации и полную информацию об узлах. В этом случае его можно использовать и как хранилище для некоторых данных и т.п., что позволит разгрузить узел управления и освободить его ресурсы для другой работы.

## Разработка алгоритма коммутации

На основе анализа различных программных решений по организации распределенных вычислительных систем был выбран оптимальный алгоритм коммутации между узлами сети «интернет вещей». Инфраструктура данного алгоритма представлена на рисунке. В данную инфраструктуру входит множество устройств (персональные компьютеры, мобильные телефоны, маршрутизаторы и др.) на базе различных операционных систем (Windows, Linux, OSX и др.). Взаимодействие между узлами, осуществляется через сервер-посредник, он сообщает узлу управления и узлу, которому требуется подключение, ip-адрес и порт, по которому производить передачу данных. Необходимо отметить, что взаимодействие в вычислительном кластере, реализовано на основе сокетного интерфейса.

Основные этапы работы вычислительного кластера при коммутации представлены ниже:

1. Запуск узла управления и сбор информации.
2. Передача информации серверу-посреднику.
3. Ожидание подключения узлов.
4. Запуск узла.
5. Сбор информации о системе.
6. Передача xml-описания серверу-посреднику.
7. Получение каналов связи с узлом управления от сервера посредника.
8. Узел входит в состояние ожидания команд.

9. Сервер-посредник передает узлу управления информацию, о подключенных узлах.
10. Принятие данных об узлах узлом управления.
11. Узел управления заполняет информацию об узле.
12. Узел управления переходит в ожидание команд для узлов и получение запросов от узлов.
13. При подключение нового узла выполняются шаги 4-12.

### Заключение

Проведенные модельные исследования по разработке алгоритма коммутации между узлами сети «интернет-вещей» показали, что разработанный алгоритм коммутации между узлами сети, обладает определенными преимуществами:

- Простота использования.
- Универсальность, а именно: любой сторонний разработчик может дополнить его необходимыми программными модулями.

### Список использованных источников

Проектирование инфраструктуры и реализация распределенного информационно-вычислительного кластера на базе персональных компьютеров. URL: <http://earchive.tpu.ru/handle/11683/40967> (дата обращения 23.11.2018)

1. Распределенные вычислительные системы / Г.И. Радченко. – Челябинск.: Фотохудожник, 2012. -184с
2. Введение в грид технологии / Радченко Г.И., Демичев А.П., Ильин В.А., Крюков А.П. и др. М.: НИИЯФ МГУ, 2007. -87с
3. Hyperic SIGAR. URL: <https://github.com/hyperic/sigar> (дата обращения 20.08.2017).
4. Java Architecture for XML Binding. URL: <https://docs.oracle.com/javase/8/docs/tech-notes/guides/xml/jaxb/> (дата обращения 05.09.2017).
5. Увеличение вычислительной мощности распределенных вычислительных систем с помощью грид-систем из персональных компьютеров. URL: [http://desktopgrid.ru/articles/increasing\\_computing\\_power.pdf](http://desktopgrid.ru/articles/increasing_computing_power.pdf) (дата обращения 21.01.2018)

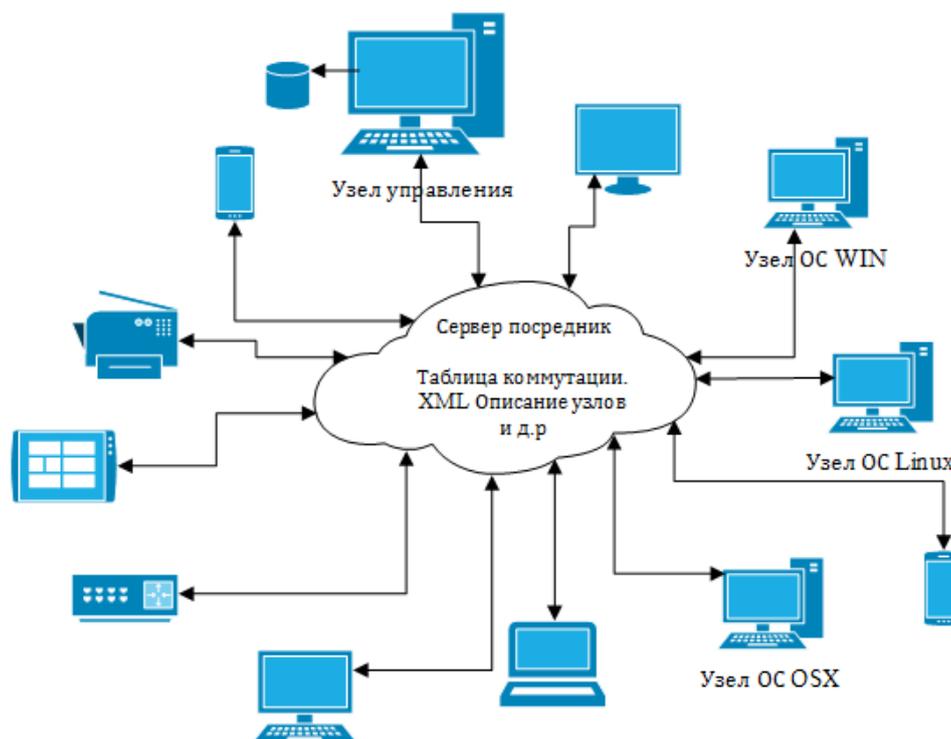


Рис. 1. Архитектура модельного вычислительного кластера