

Протокол

December 20, 2014

1 Инициализация системы

Клиенты разделяются на три типа:

1. Первый
2. Промежуточный
3. Последний

Определим используемые методы:

- *broadcast()* - рассылка широковещательных сообщений на порт 33333. Сообщения содержат порт и ip-адрес для входящих соединений TCP. После подключения к данному узлу рассылка прекращается.
- *connect()* - прослушивание широковещательных сообщений; создание TCP соединения после получения номера порта.
- *wait_last_connection()* - ожидание широковещательного сообщения последнего узла. Порт 33334.
- *create_connection()* - создание TCP сервера на определенном порту.
- *last_connection_broadcast()* - рассылка широковещательного сообщения для индикации запуска последнего узла. Содержание сообщения - "*last_[node id]*", где *node id* - идентификатор последнего узла; используется для определения количества узлов.

Пометка: первый узел имеет порт для входящего TCP соединения 30000, затем каждый следующий добавляет 1. По порту узел узнает о своем номере в построенной линейной топологии. Передача номера порта и ip производится в строковом формате: "[port] [ip]".

1.1 Использование методов в зависимости от типа узла

- Первый: *broadcast()* -> *create_connection()* -> *wait_last_connection()*
- Промежуточный: *connect()* -> *broadcast()* -> *create_connection()* -> *wait_last_connection()*
- Последний: *connect()* -> *last_connection_broadcast()*

2 Общение

Передача команд двумя сообщениями, содержащими строку. В конце передается символ переноса строки.

Первое сообщение - команда.

Второе - данные.

2.1 Команды

[] - обозначение опции; сами скобки не пишутся в команду.

“transfer[node id]” - передача блока данных на определенный узел. Содержание данных не специфицируется.

“get_data[destination node]” - запрос на фрагмент данных на [destination node]. В строке данных передается идентификатор фрагмента данных. (пометка: каждый узел имеет функцию определения местонахождения адреса обитания фрагмента данных по его идентификатору)

“put_data[destination node]” - ответное сообщение на запрос фрагмента данных. Поле [destination node] отвечает за адрес назначения сообщения. В строке данных передается сериализованный объект фрагмента данных или строка *“nil”* в качестве идентификатора отсутствия фрагмента.

“remove” - безадресная команда удаления фрагмента данных. В строке данных содержится идентификатор фрагмента. (пометка: каждый узел имеет функцию определения местонахождения адреса обитания фрагмента данных по его идентификатору)

“status” - отладочная команда; производит печать информации о фрагментах данных и вычислений. Пробрасывается через узел.

“kill” - команда завершения работы узла. Пробрасывается через узел.

3 Сериализация

3.1 Фрагменты данных

Сериализация фрагментов данных производится в строку:

“Data [id] [type] [internal data]”

Где:

- *id* - идентификатор фрагмента данных.
- *type* - формат внутренних данных.
- *internal data* - сериализованное представление внутренних данных.

3.2 Фрагменты вычислений

Сериализация фрагментов данных производится в строку:

“Task [id]”

Где:

- *id* - идентификатор фрагмента данных.

Пометка: динамическое порождение фрагментов вычислений не предусматривается и весь набор фрагментов известен в начале работы.