

Message Broker

Введение

Соглашения о терминах

Сетевое сообщение (сообщение) – законченный, неделимый (с точки зрения клиента) блок информации, пригодный для переправки по компьютерной сети.

Область применения

Применяется для обеспечения взаимодействия между модулями в распределенной системе обработки информации и управления радиолокатором.

Общее описание

Назначение и возможности продукта

Брокер сетевых сообщений предоставляет сервис по обмену сообщениями между абонентами. Весь трафик проходит через одну точку системы – сервер, который в свою очередь занимается переадресацией и пересылкой сообщений другим адресатам - клиентам.

Среда функционирования продукта

Исполняется в среде операционных систем Windows и Ubuntu Linux. Модуль поставляется в качестве исходных текстов – заголовочные (.h) и компилируемые (.cpp) файлы или статической библиотеки – заголовочный (.h) файл и объектный код (.lib и .a) файл.

Допущения и зависимости

При реализации используются библиотеки:

- ZeroMQ
- Cereal

Описание системы

Для обмена сообщениями применяется архитектура типа издатель/подписчик (publisher/subscriber). При данном подходе имеется возможность не указывать адресат сообщений и соответственно последовательно подключать другие модули и расширять функциональность без изменений в ранее подключенных модулях.

Фильтрация сообщений происходит по их типу в текстовом представлении. Протокол обмена определяется промежуточным программным обеспечением (middleware) сетевого обмена, в данном случае ZeroMQ. Требования к сообщениям определяются библиотекой сериализации сообщений.

Конфигурирование клиента и сервера осуществляется через файл конфигурации. Состав и формат конфигурационных файлов описан в Приложении А. Все сообщения должны иметь обязательное поле `message_type`.

При вызове интерфейсных функций возможны следующие типы исключений:

- `SystemException` – ошибки, вызванные ошибками при работе с функциями операционной системы или сторонних библиотек.
- `ConfigException` – ошибка при чтении или разборе описания конфигурации.

Более детальное сообщение об ошибке выводится в файл `<Дата и время запуска>.log` в каталоге `<Исполняемый файл>/logs/`.

Серверная подсистема

Описание

Подсистема предназначена для обеспечения обмена сетевыми сообщениями между клиентами.

Функциональные требования

- `load_config(config_manager)`.
Загрузка конфигурации с помощью менеджера конфигураций. В случае ошибки генерирует исключение `SystemException` или `ConfigException`. Вызов функции возможен только при остановленном сервере.
- `start()`.
Запуск сервера на исполнение согласно параметрам, представленным в файле конфигурации. Вызов блокируемый. В случае ошибки исключение `SystemException`. Если запуск производится до загрузки конфигурации, то генерируется `ConfigException`.
- `stop()`.
Остановка сервера и закрытие всех установленных соединений. В случае ошибки возвращает `SystemError`. В случае остановки незапущенного сервера ошибка не генерируется. После остановки возможен повторный запуск сервера на исполнение.

Клиентская подсистема

Описание

Предназначена для предоставления интерфейса обмена сообщениями с абонентами. Так же имеет возможность запуска автоматической отправки контрольного heartbeat сообщения серверу (состав и формат heartbeat сообщения приведен в Приложении Б).

Функциональные требования

- `load_config(config_manager)`.
Загрузка конфигурации с помощью менеджера конфигураций. В случае ошибки генерирует исключение `SystemException` или `ConfigException`. Вызов функции возможен только при остановленных обработке очереди сообщений и посылки heartbeat сообщения.
- `connect()`.
Установка соединения с сервером. В случае ошибки генерирует исключение `SystemException`. Если попытка установки соединения было до загрузки конфигурации, то генерируется исключение `ConfigException`.
- `disconnect()`.
Отмена текущих операций приема или передачи данных и разрыв текущего соединения с сервером. В случае ошибки генерирует исключение `SystemException`. В случае разрыва соединения до установки соединения ошибка не генерируется. После разрыва соединения возможна повторная установка соединения с сервером.
- `send_message(message)`.
Переслать сообщение `message`. Вызов неблокируемый. В случае ошибки возвращает `SystemError`. При попытке переслать сообщение до загрузки конфигурации или установки соединения с сервером генерируется исключение `ConfigException`.
- `register_handler(message_type, handler_function)`.
Регистрация функции-обработчика `handler_function` для сообщения типа `message_type`. При повторной регистрации обработчика для одного типа – новый обработчик игнорируется.
- `start_heartbeat()`.
Запуск автоматической рассылки heartbeat сообщения с периодом, определенным в конфигурационном файле. Вызов неблокируемый. В случае ошибки генерирует исключение `SystemException`. При вызове этой функции до загрузки конфигурации генерируется исключение `ConfigException`.
- `stop_heartbeat()`.
Остановка рассылки heartbeat сообщения. В случае ошибки генерирует исключение `SystemException`.
- `start_massge_processing()`.
Запуск обработки входных сообщений для зарегистрированных ранее функций обработчиков. Вызов блокируемый. В случае ошибки генерирует исключение `SystemException`. При вызове этой функции до загрузки конфигурации или при пустой таблице обработчиков генерируется исключение `ConfigException`.
- `stop_massge_processing()`.
Остановка обработки входных сообщений.

Приложение А. Состав и формат файла конфигурации.

Состав конфигурационных параметров

| Имя | Тип | Описание |
|--------------------|--------|---|
| server_ip | string | IP адрес сервера. |
| server_pub_port | string | Номер порта для отправки сообщений сервером. |
| server_sub_port | string | Номер порта для приема сообщений сервером. |
| client_pub_port | string | Номер порта для отправки сообщений клиентом. |
| client_sub_port | string | Номер порта для приема сообщений клиентом. |
| buffer_size | uint32 | Размер буфера входных сообщений. В байтах. |
| poll_interval | uint32 | Интервал опроса входных сообщений. В микросекундах. |
| heartbeat_interval | uint32 | Период передачи контрольных heartbeat сообщений. В миллисекундах. |

Формат конфигурационного файла

```
{
  "message_broker_settings" :
  {
    "server_ip" : "XXX.XXX.XXX.XXX",
    "server_pub_port" : "XXXXX",
    "server_sub_port" : "XXXXX",
    "client_pub_port" : "XXXXX",
    "client_sub_port" : "XXXXX",
    "buffer_size" : XXXXX,
    "poll_interval" : XXXXX,
    "heartbeat_interval" : XXXXX
  }
}\n
```

Приложение Б. Состав и формат сообщений обмена.

Состав сообщений

Внутренние сообщения

Внутренние сообщения – сообщения циркулирующие внутри системы, между отдельными клиентами брокера сетевых сообщений.

Каждое внутреннее сообщение должно представлять собой структуру языка C++ и иметь:

- строковое поле тип (type), с помощью которого будет осуществляться фильтрация сообщений на клиентах при используемой архитектуре – издатель/подписчик.
- интерфейс serialize (deserialize) для упаковки (распаковки) сообщений из формата структуры языка C++ в последовательность байтов (и наоборот).

Сообщение heartbeat

| heartbeat | | |
|-----------|--------|--|
| Имя | Тип | Описание |
| from | string | Идентификатор клиента источника данного сообщения. |

Формат сообщений

Сообщение heartbeat

```
{  
  "message" : "hearteat",  
  "from" : "XXXXXXXX"  
}\n
```

Внутренние сообщения

Формат внутреннего сообщения, для передачи по сети, определяется форматом представления, принятом в библиотеке Cereal.