

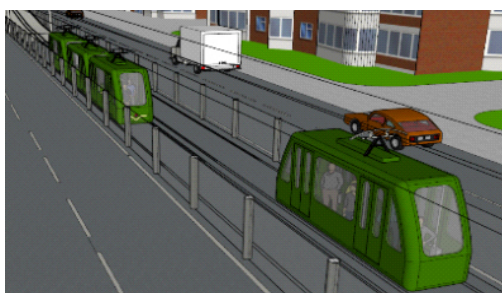
ИСПОЛЬЗОВАНИЕ СУБД POSTGRESQL ДЛЯ ОБРАБОТКИ ДАННЫХ В ГОРОДСКОЙ ПАССАЖИРСКОЙ ИНФОРМАЦИОННО-ТРАНСПОРТНОЙ СИСТЕМЕ

Основной функцией большинства СУБД и баз данных в распределенной архитектуре клиент-сервер является хранение данных. Частично СУБД может взять на себя часть их обработки, но основной её объем в таких архитектурах располагается вне контекста СУБД, зачастую удаленно. Такое построение системы обработки данных имеет существенный недостаток: временные потери при обмене данными между СУБД и сторонним клиентским звеном, обрабатывающим данные.

Целью предложенной работы является описание использования возможностей серверных вызовов сетевой СУБД PostgreSQL, написанных на языках высокого уровня (C, Perl, Python), для построения системы обработки данных в городской пассажирской информационно-транспортной системе (ИТС) на базе беспилотных электрокаров. Эта СУБД отличается надёжностью и отличной производительностью, обладает расширяемой системой встроенных языков программирования. Рассматриваемая ИТС включает в себя [1-4]:

- Парк беспилотных транспортных средств небольшого объема (6-15 мест), называемых инфобусами. Действия инфобуса координируется собственной бортовой компьютерной системой, действующей под управлением единого информационного сервера. Инфобусы движутся по выделенной линии (рельсовому пути) (Рис. 1а).

- Система терминалов на остановках для сбора заявок на перевозку (Рис. 1б).



а) Инфобусы на линии



б) Система терминалов

Рисунок 1 – Элементы ИТС

- Информационный сервер, собирающий заявки пассажиров, сделанные через терминалы остановок, в матрице корреспонденций (рис. 2а). Через постоянный опрос поступивших данных сервер отслеживает наступление момента достаточного накопления заявок и фиксирует матрицу

¹ Швецов Елена Владимировна

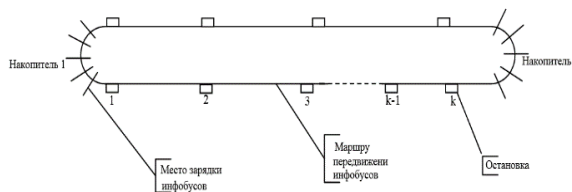
Брестский государственный технический университет

Старший преподаватель кафедры интеллектуальных информационных технологий

корреспонденций. На основании этих данных составляется план перевозки, пересылаемый бортовым системам инфобусов, для его реализации. После чего открывается новый цикл сбора заявок

- Маршрут движения инфобусов содержит по k остановок в двух направлениях, а также два Накопителя, из которых инфобусы начинают выполнение плана перевозки и в которые возвращаются после его реализации (Рис. 2б)

$$M = \begin{pmatrix} 0 & m_{12} & m_{13} & \dots & \dots & m_{1j} & \dots & m_{1k} \\ 0 & 0 & m_{23} & \dots & \dots & m_{2j} & \dots & m_{2k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & m_{i+1} & \dots & m_{ij} & \dots & m_{ik} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & m_{k-1k} \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix}$$



а) Матрица корреспонденций

б) Маршрут движения

Рисунок 2 – Элементы ИТС

Функционирование транспортной системы имеет циклический характер и состоит из процедуры сбора заявок, процедуры определения момента достаточного накопления заявок (выполняются одновременно), процедуры фиксации матрицы корреспонденций, процедуры составления и реализации плана перевозки (Рис.3) [1-4].

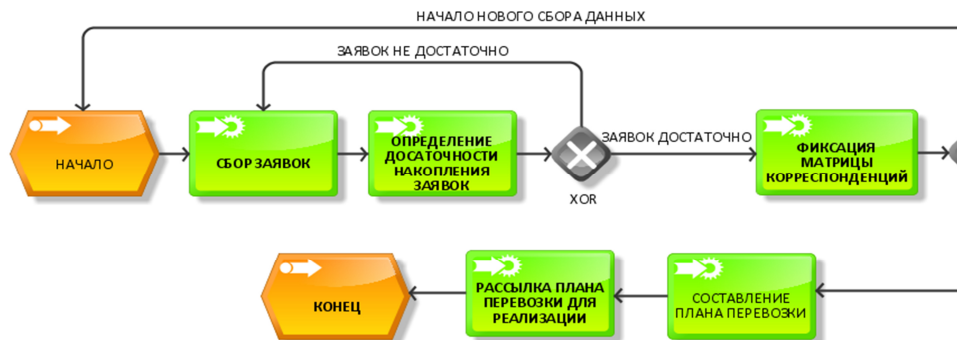
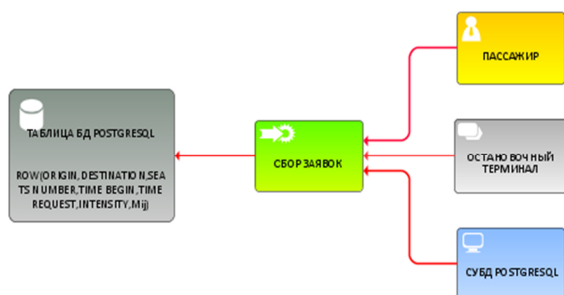


Рисунок 3 – Схема функционирования ИТС

Процедура сбора заявок заключается в записи заявки, сделанной пассажиром через терминал на остановке, в БД PostgreSQL. База данных и сама СУБД располагаются на стороне информационного сервера транспортной системы (рис.4а). Заявка содержит данные начальной остановки (ORIGIN (i), рис. 4б), с которой пассажир начинает поездку, и целевой остановки (DESTINATION (j), рис. 4б), на которую пассажир желает ехать, и требуемое количество мест (SEATS NUMBER, рис. 4б). Также фиксируется время начала цикла сбора заявок (BEGIN TIME), время поступления заявки (REQUEST TIME). При записи заявки в БД автоматически рассчитывается интенсивность (INTENSITY) поступления заявок на перевозку с остановки i на остановку j. Данная

характеристика используется для получения некоторых прогнозных значений, используемых в ИТС.



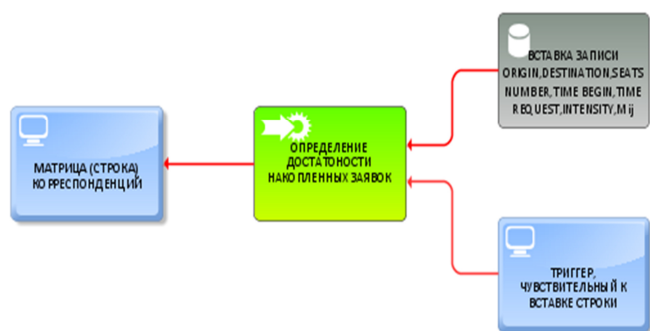
STATUS	BEGIN TIME	REQUEST TIME	ORIGIN (i)	DESTINATION (j)	SEATS NUMBER	INTENSITY
Start	00:26:06.1866667	00:26:06.1700000	1	0	0	0
Flow	00:26:06.1866667	00:26:25.9466667	1	2	2	0.1052632
Flow	00:26:06.1866667	00:26:47.5433333	1	5	1	0.02439024
Flow	00:26:06.1866667	00:27:01.8033333	1	3	4	0.07272727
Flow	00:26:06.1866667	00:27:19.0233333	1	4	1	0.01369863
Flow	00:26:06.1866667	00:27:35.3133333	1	6	3	0.03370786
Flow	00:26:06.1866667	00:27:52.9933333	1	7	2	0.01886792
Flow	00:26:06.1866667	00:28:11.1400000	1	3	3	0.056
Flow	00:26:06.1866667	00:29:13.3300000	1	8	2	0.01069519
Flow	00:26:06.1866667	00:29:30.1000000	1	3	5	0.05882353
End	NULL	00:29:30.1000000	1	0	0	0
Start	00:29:30.1000000	00:29:30.1000000	1	0	0	0

а) Схема процедуры

б) Таблица БД с данными заявок

Рисунок 4 – Сбор заявок пассажиров в ИТС

Для определения наступления момента достаточности накопления заявок используется условие, которое отслеживается триггером базы данных (процедурой, привязанная к таблице БД, автоматически запускающейся при вставке строки в таблицу БД) (Рис. 5а).



ЗАМЕЧАНИЕ: 0 3 5 0 0 1 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 1 0 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 0 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 6 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 4 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 0 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 0 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 0 0 0 0 0
 ЗАМЕЧАНИЕ: 0 0 0 0 0 0 0 0 0 0
 CALL

Query returned successfully in 118 msec.

а) Схема процедуры

б) Матрица корреспонденций

Рисунок 5 – Определение достаточности накопления заявок

При определении наступления момента достаточного накопления заявок в ИТС алгоритм триггера запускает новый сбор заявок, формирует матрицу корреспонденций (Рис.5б), делает серверный вызов процедуры обработки данной матрицы (C, Python, Perl) и формирует на основе неё план перевозки. Сформировав план перевозки, вызванная триггером процедура рассылает его бортовым системам инфобусов (Рис.6).



Рисунок 6 – Составление и рассылка плана перевозки

ЗАКЛЮЧЕНИЕ. Предложенный подход позволяет осуществлять всю алгоритмическую обработку данных на стороне сервера СУБД PostgreSQL, избегая использования вызовов стороннего клиентского программного обеспечения, что приводит к сокращению времени выполнения необходимых алгоритмов, т.к. не требуется установление связи со сторонним клиентским программным обеспечением и обмен данными с ним.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Швецова, Е.В. Алгоритм составления плана перевозок на городских линиях в интеллектуальной системе управления беспилотными транспортными средствами / Е.В. Швецова, В.Н. Шуть // Вестник Херсонского национального технического университета. – Т. 2(69), № 3. – Херсон: ХНТУ, 2019. – С. 222-230

2. Shuts, V. System of urban unmanned passenger vehicle transport / V. Shuts, A. Shviatsova // ICCPT 2019: Current Problems of Transport: Proceedings of the 1st International Scientific Conference. – Ternopol: TNTU, 2019 – С. 172-184

3. Shviatsova A., Shuts V. The Smart Urban Transport System / V. Shuts, A. Shviatsova // Research Papers Collection of Open Semantic technologies for Intelligent System. - Minsk: Belarusian State University of Informatics and Radioelectronics, 2020. - p. 349-352.

4. Shviatsova A., Shuts V. The smart urban transport system based on robotic vehicles // National Academy of Sciences of Ukraine Institute of Artificial Intelligence Problems ARTIFICIAL INTELLIGENCE 2019 № 3-4 (85-86). - Kiev: BNTU, 2019. - p. 40-49.