

Реализация сетевого интерфейса на основе формул

Пастернак И.И.¹, Морозов Ю.В.²

¹ Кафедра электронных вычислительных машин, Национальный университет "Львовская политехника", УКРАИНЫ, Львов, ул.С.Бандеры, 28а, E-mail: pas_irusj@ukr.net

² Кафедра электронных вычислительных машин, Национальный университет "Львовская политехника", УКРАИНЫ, Львов, ул.С.Бандеры, 28а, E-mail: m_urij@ukr.net

Аннотация—Предложен вариант клиент-серверного взаимодействия на основе сетевого интерфейса, который реализуется с помощью формул.

Ключевые слова: клиент, сервер, клиент-серверное взаимодействие, компьютерная сеть, сетевое программное обеспечение, сетевой интерфейс.

I. ВВЕДЕНИЕ

Сетевой интерфейс - это два или более компьютера, объединенных кабелем таким образом, чтобы они могли обмениваться информацией. Для настройки подключения персонального компьютера к сети необходимо последовательно выполнить следующие операции [1]:

1 этап. Физически подключить модем к персональному компьютеру и телефонной линии. После этого необходимо установить его драйверы и настроить параметры работы.

2 этап. Настроить параметры подключения к серверу удаленного доступа.

3 этап. Настроить параметры протокола ТСР / IP.

Соединения персональных компьютеров может происходить с помощью кабеля и сетевых адаптеров, или кабеля подключенного к портам (прямое кабельное соединение), телефонной или оптоволоконной линии и модема, или радиосвязью с помощью радио модема.

II. ПОСТАНОВКА ЗАДАЧИ

Оценить границы применения сетевого интерфейса при взаимодействии клиента с сервером. А также его доступность и реализации на основе формул и протестировать его на запросы от клиента к серверу.

III. ОСНОВНЫЕ РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ

Исходя из основных принципов описания сетевого интерфейса, выделим его основные компоненты - входы и выходы. Входом сетевого интерфейса есть множество запросов от клиентов, выходами - множество ответов на запросы. В общем случае реакция определяется не только текущим запросом, но и всеми предыдущими. Для устранения необходимости каждый раз обрабатывать все предыдущие запросы сетевого интерфейса является агрегированной историей запросов данного сетевого интерфейса. В сетевом интерфейсе выделяются

отдельные уровни (рис.1), он является многоуровневым и на каждом уровне содержится не менее одной автономной подсистемы (которая фактически является отдельной системой) [2].

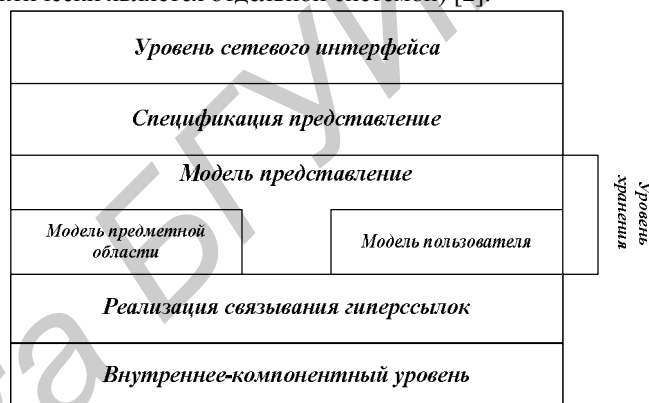


Рис.1. Вид уровней сетевого интерфейса

На каждом из уровней (кроме верхнего и нижнего) традиционно рассматривается три входных потока - высшего уровня - спецификация представления, поток реакций низшего уровня – внутренне компонентный.

Сетевой интерфейс должен обеспечивать возможность иерархического соединения клиента с сервером в сети. Такое соединение базируется на направлении исходящего потока низшего уровня в высший уровень сетевого интерфейса. В формировании ответа сервиса на запрос пользователя с использованием сетевого интерфейса (рис.1) выделены следующие этапы:

1. получения запроса от пользователя или высшего сервиса - уровень модели представление;
2. формирования запросов до нижнего сервиса - уровень спецификации представление;
3. получения ответов от низших сервисов – внутренне компонентный уровень;
4. формирования ответа на запрос - уровень связывания для гиперссылок.

Сетевой интерфейс описывается традиционным упорядоченным набором:

$$IS = (Q, R, A, St, \phi, \psi) \quad (1)$$

где Q - входящие запросы; IC, R - ответы низкого уровня сетевого интерфейса; A - выходной алфавит; IC, St-множество состояний системы; ϕ, ψ - функции переходов и выходов.

Рассмотрим подробнее каждый из объектов. Множество символов, составляющих входной алфавит

сетевого интерфейса, опишем следующим образом:

$$Q = \{Qi\} \quad (2)$$

$$Qi = \{Id_Q^{(i)}, Pq_1^{(1)}, \dots, Pq_{N_Q^{(i)}}^{(1)}\} \quad (3)$$

$$P_j^{(i)} \subset D_{Q_j^{(1)}}^{(1)} \times \dots \times D_{Q_j^{(l)}}^{(l)}, l = N_{P_{Q_i}}^{(j)} \quad (4)$$

где $Id_Q^{(i)}$ - множество уникальных идентификаторов запросов.

Множество символов, составляющих выходной алфавит сетевого интерфейса, опишем следующим образом:

$$A = \{Ai\} \quad (5)$$

$$Ai = \{Id_A^{(i)}, Pa_1^{(1)}, \dots, Pa_{N_A^{(i)}}^{(1)}\} \quad (6)$$

$$P_j^{(i)} \subset D_{A_j^{(1)}}^{(1)} \times \dots \times D_{A_j^{(l)}}^{(l)}, l = N_{P_{A_i}}^{(j)} \quad (7)$$

где $Id_A^{(i)}$ - множество уникальных идентификаторов ответов.

Множество символов, составляющих ответа низкого уровня сетевого интерфейса, опишем следующим образом:

$$R = \{Ri\} \quad (8)$$

$$Ri = \{Id_R^{(i)}, Pr_1^{(1)}, \dots, Pr_{N_R^{(i)}}^{(1)}\} \quad (9)$$

$$P_j^{(i)} \subset D_{R_j^{(1)}}^{(1)} \times \dots \times D_{R_j^{(l)}}^{(l)}, l = N_{P_{R_i}}^{(j)} \quad (10)$$

где $Id_R^{(i)}$ - множество уникальных идентификаторов ответов низкого уровня сетевого интерфейса.

Таким образом, мы получим реализацию сетевого интерфейса на основе формул.

На основе формул для описания сетевого интерфейса мы можем провести тестирование этого интерфейса и посмотреть время его реакции на запрос клиента к серверу в сети

Нагрузочного тестирования сетевого интерфейса необходимо в случаях, когда нужно получить ответы на следующие вопросы:

- что произойдет с системой при увеличении числа пользователей;
- где предел реализации сетевого интерфейса;
- какое программное обеспечение (ПО) лучше отвечает потребностям сетевого интерфейса в плане производительности;
- какое оборудование необходимо для комфортной работы с сетевым интерфейсом;
- что является потенциальным узким местом сетевого интерфейса с точки зрения производительности.

На первый взгляд кажется, что на вышеперечисленные вопросы можно ответить и без проведения комплексных исследований и тестирования сетевого интерфейса. Но это справедливо только для сравнительно простых сетевых интерфейсов и даже для них справедливо далеко не всегда.

В любом проекте нагрузочного тестирования можно выделить следующие стадии:

- 1) исследование сетевого интерфейса;
- 2) подготовка проектной документации;
- 3) создание инструментария тестирования;
- 4) проведения тестирования;
- 5) обработка результатов и подготовка отчета.

Нагрузочное тестирование сетевого интерфейса с клиент-серверной архитектурой предназначено для определения максимального количества запросов от клиента (пропускной способности) и оптимальной конфигурации сервера при следующих состояниях сетевого интерфейса: нормальном (base), нагруженном (load), критическом (stress).

Во всех случаях при тестировании происходит определение времени стабильного пребывания в фиксированном состоянии и рубеж перехода в следующий (рис.2).

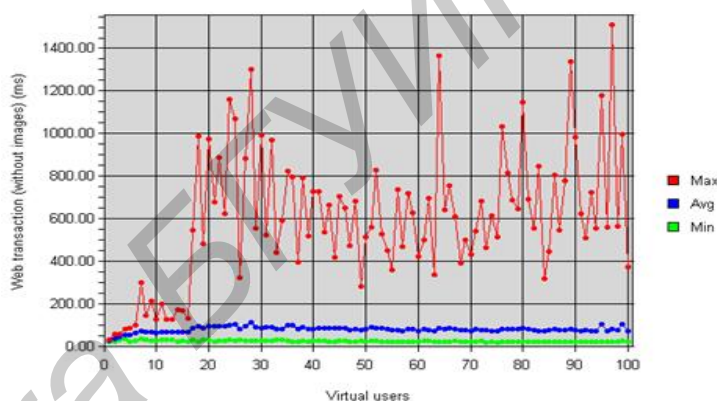


Рис.2. Зависимость времени отклика сервера от количества запросов от клиента

В зависимости от количества запросов от клиента реакция сервера будет меняться, чем больше запросов от клиента тем время реакции сервера больше (рис.2). Отметим, что после превышения порога критической нагрузки происходит сбой сетевого интерфейса различной степени сложности.

ЗАКЛЮЧЕНИЕ

Итак, для того чтобы лучше понять работу и функций сетевого интерфейса при модульной клиент-серверной взаимодействия использую формулы. При использовании этих формул можно быстро и эффективно заменить один параметр сетевого интерфейса, что позволит изменить и весь его облик.

Время реакции этого интерфейса будет зависеть от нагрузки запросов от клиента к серверу, но благодаря реализации сетевого интерфейса на основе формул мы можем улучшить эффективность работы сервера через множество параметров, заданная через формулы, как мы видим на рис.2.

- [1] Липаев В.В. «Обеспечение качества программных средств. Методы и стандарты». - М.: Синтег, 2001. - С. 246.
- [2] Макгрегор Дж., Сайкс Д. «Тестирование объектно-ориентированного программного обеспечения». - К: Диасофт, 2002. - С. 432.