докторів наук до 45 років. Тема роботи "Розроблення інформаційних технологій автоматизації структурного синтезу та аналізу мікроелектромеханічних систем", 2011 р. (Ф35/541-2011, №0111U009116).

Поступила 19.08.2011р.

УДК 621.395.7

Е.Г.Игнатенко, В.В.Турупалов,
Донецкий национальный технический университет

АЛГОРИТМ БАЛАНСИРОВКИ НАГРУЗКИ ДЛЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ С КЛАСТЕРНОЙ АРХИТЕКТУРОЙ

Введение. В последнее время наблюдается бурное развитие информационных технологий, что приводит к необходимости построения сетей согласно концепции NGN (Next Generation Network) [1]. Концепция NGN предусматривает иерархическую структуру телекоммуникационной сети с отделением уровня предоставления услуг. Как правило, этот уровень строится на основе кластерной архитектуры и требует эффективных средств балансировки нагрузки между источниками услуг (серверами). Применение систем балансировки нагрузки (СБН) для распределения увеличивающейся нагрузки позволяет повысить эффективность функционирования сетей с кластерной архитектурой.

Постановка задачи. Недостатки известных алгоритмов балансировки нагрузки (БН), такие как, низкая производительность, высокое время ответа, неравномерное распределение нагрузки между серверами, недостаточный учет динамики системы, большие накладные расходы ресурсов [2-5] позволяют говорить об актуальности разработки алгоритма БН в сети с кластерной архитектурой. Разрабатываемый алгоритм БН должен быть адаптирован к изменениям интенсивности входящего потока для уменьшения времени ответа, вероятности потерь, повышения производительности системы и снижения количества служебной информации.

Основная часть. Рассмотрим структуру телекоммуникационной сети с кластерной архитектурой, приведенную на рисунке 1. В сети с кластерной архитектурой может быть от двух до нескольких десятков узлов (серверов), при этом это могут быть узлы различной аппаратной конфигурации. Для клиента все они представляются в виде единого виртуального сервера.

Важную роль в построении сети с такой конфигурацией играет система балансировки нагрузки, в основе которой лежит алгоритм БН. СБН серверов - это инструментальное средство, предназначенное для переадресации

клиентских запросов на наименее загруженный или наиболее подходящий сервер из группы машин. При построении сети такой конфигурации используется структура с общей системой хранения данных (СХД).

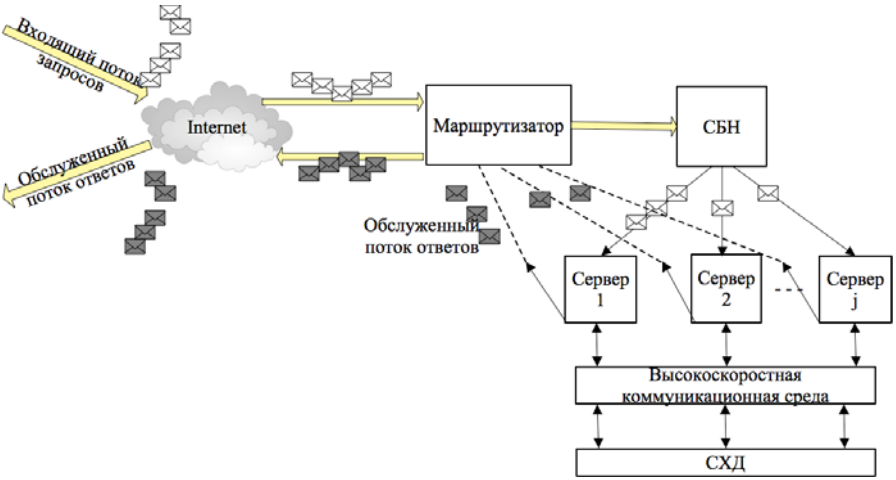


Рис. 1 - Телекоммуникационная сеть с кластерной архитектурой

К приложениям, требующим балансировки, как правило, относят web-серверы, серверы электронной почты и DNS-серверы, которые обслуживают сеть Интернет и корпоративные сети организаций.

Решение задачи балансировки загрузки состоит из следующих шагов (рис.2): оценка загрузки узлов сети; принятие решений о балансировке; распределение запросов. Математически задачу БН можно представить следующей зависимостью:

$$X = f(N, G, \bar{v}), \qquad (1)$$

где N – множество серверов кластера $N = \{N_j\}$;

G – множество характеристик серверов $G = \{U_{j\max}, U_j\}$,

$U_{j\max}$ - максимальная загруженность сервера;

U_j - текущая загруженность сервера;

$\bar{v}$ - характеристика входящего потока запросов.

Основной идеей разработанного алгоритма БН является распределение пользовательских запросов на основании прогноза входящего потока по его типам. При этом горизонт прогноза на каждом шаге изменяется в зависимости от изменений во входящем потоке. Также алгоритм учитывает не только текущее состояние сервера, но и его производительность. Рассмотрим последовательность выполняемых операций для алгоритма БН.

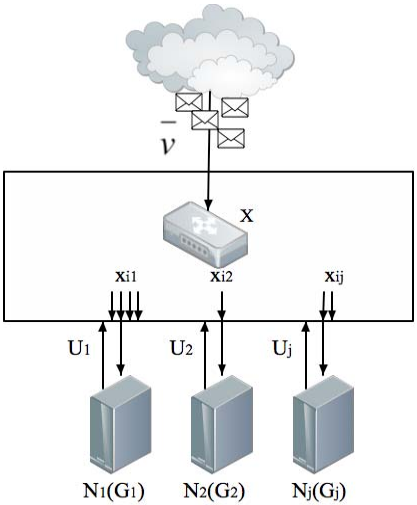


Рис.2 - Распределение запросов при динамической балансировке

1. Сбор и анализ статистической информации: интенсивности входящего потока запросов $\lambda(k-\omega)...\lambda(k-1)$, состояния серверов (утилизации CPU - $U_j(k-1)$, загруженность *j* сервера, создаваемая *i* классом запросов - z_{ij}).

2. Вычисление коэффициента пачечности входящего потока запросов $b_m(k-2), b_m(k-1)$ (на основе данных п.1) [6]:

$$b_m = \frac{g}{n} \cdot \left(1 + \frac{\lambda(k) - \lambda(k-1)}{\lambda(k-1)}\right). \quad (2)$$

3. Расчет интервала мониторинга загруженности серверов и горизонта прогноза $d(k)$ по формуле:

$$d(k+1) = \frac{b_m(k-1)}{b_m(k)} \cdot d(k). \quad (3)$$

4. Прогноз интенсивности входящего потока по каждому типу запросов a_i^* на длину горизонта $d(k)$, определяемую в п.3.

5. Расчет матрицы распределения запросов по узлам сети $\overline{X}(k)$, с учетом типа запроса и загруженности серверов. На основе полученных данных прогнозируется загруженность серверов на следующем шаге.

6. Распределение запросов по серверам, согласно алгоритму RR в пределах каждого класса запросов.

7. Распределение недооценки прогнозируемого количества запросов $a_i(k) - a_i^*(k)$ согласно алгоритму CAP. Переоценка - не учитывается алгоритмом, т.к. не вносит существенных изменений.

8. Снятие данных о загруженности серверов $U_j(k)$ и передача их на СБН, для расчета нового распределения запросов $\overline{X}(k+1)$.

Алгоритм балансировки нагрузки должен распределять запросы по серверам так, чтобы отклонение загруженности серверов от среднего значения было минимальным, т.е.

$$s = \frac{\sum_{j=1}^{N}(\overline{U} - U_j(k))^2}{N} \to \min, \tag{4}$$

где

$$\begin{cases} U_1(k-1) + \sum_{i=1}^{M} a_i^*(k) \cdot x_{i1}(k) \cdot z_{i1} = U_1(k) \\ U_2(k-1) + \sum_{i=1}^{M} a_i^*(k) \cdot x_{i2}(k) \cdot z_{i2} = U_2(k) \\ \ldots \\ U_j(k-1) + \sum_{i=1}^{M} a_i^*(k) \cdot x_{ij}(k) \cdot z_{ij} = U_j(k) \\ \overline{U} = \frac{\sum_{j=1}^{N} U_j(k)}{N} \end{cases};$$

$$\sum_{j=1}^{N} x_{ij}(k) = 1, i = (\overline{1,M}); \sum_{j=1}^{N} a_{ij}^*(k) = a_i^*, i = (\overline{1,M});$$

$$x_{ij}(k) > 0; z : i \times j, i = (\overline{1,M}), j = (\overline{1,N}).$$

где a_i^* - прогнозное количество запросов.

В качестве искомой матрицы выступает матрица распределения запросов:

$$X(k) = \left[x_{i,j} \right], (i = 1, M; j = 1, N), \tag{5}$$

в результате расчета которой обеспечивается динамическое распределение нагрузки по серверам на *k*-ом шаге.

Для оценки эффективности разработанного алгоритма проведено моделирование работы сети с кластерной архитектурой. Для распределения запросов на узлы сети, с целью их обработки, применяются следующие алгоритмы: CAP, LARD, WRR и разработанный алгоритм.

Для сравнения эффективности работы оценены следующие параметры: равномерность загруженности серверов (4); среднее время ответа; вероятность потери запроса; пропускная способность сети.

График равномерности загруженности серверов для алгоритмов WRR, CAP, LARD и разработанного приведен на рис.3:

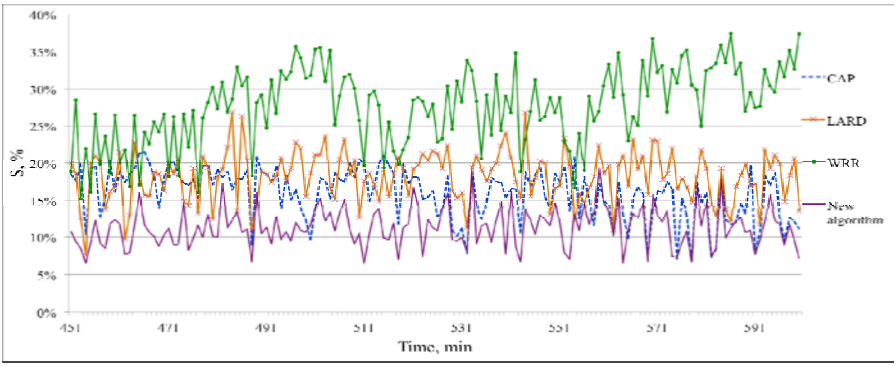


Рис.3 – Равномерность загруженности серверов

Среднее значение равномерности загруженности для разработанного алгоритма составило 8,69%, в то время как для остальных алгоритмов этот параметр колеблется от 11,13% до 29,43%. В результате равномерного распределения нагрузки по серверам наблюдается повышение эффективности по таким параметрам как среднее время ответа, вероятность потерь и пропускная способность сети. Результаты приведены на рис. 4-5:

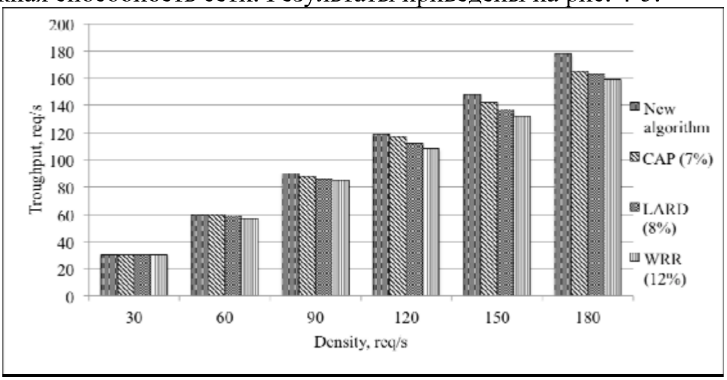


Рис.4 - Зависимость пропускной способности сети от интенсивности входящего потока

На основе результатов моделирования, получена вероятность потери запроса как отношение количества потерянных запросов к общему количеству запросов. Количество потерянных и полученных запросов по каждому серверу приведено на рис.6:

Экспериментально, с помощью моделирования, показано, что при использовании разработанного алгоритма достигается выигрыш в уменьшении времени ответа сети на 20%, вероятности потерь на 12% и увеличении пропускной способности на 10% в сравнении с WRR, LARD и CAP за счет более эффективного распределения нагрузки и использования

ресурсов. Заметный выигрыш разработанного алгоритма наблюдается при высокой нагрузке на сервера.

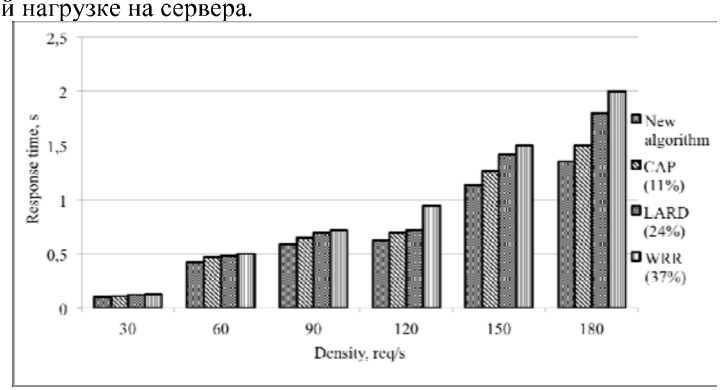
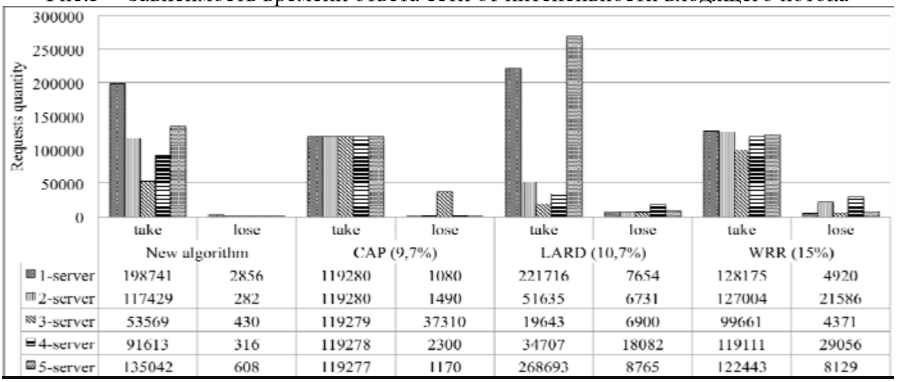


Рис.5 - Зависимость времени ответа сети от интенсивности входящего потока



	take	lose	take	lose	take	lose	take	lose
	New algorithm		CAP (9,7%)		LARD (10,7%)		WRR (15%)	
1-server	198741	2856	119280	1080	221716	7654	128175	4920
2-server	117429	282	119280	1490	51635	6731	127004	21586
3-server	53569	430	119279	37310	19643	6900	99661	4371
4-server	91613	316	119278	2300	34707	18082	119111	29056
5-server	135042	608	119277	1170	268693	8765	122443	8129

Рис. 6 - Количество полученных и потерянных запросов

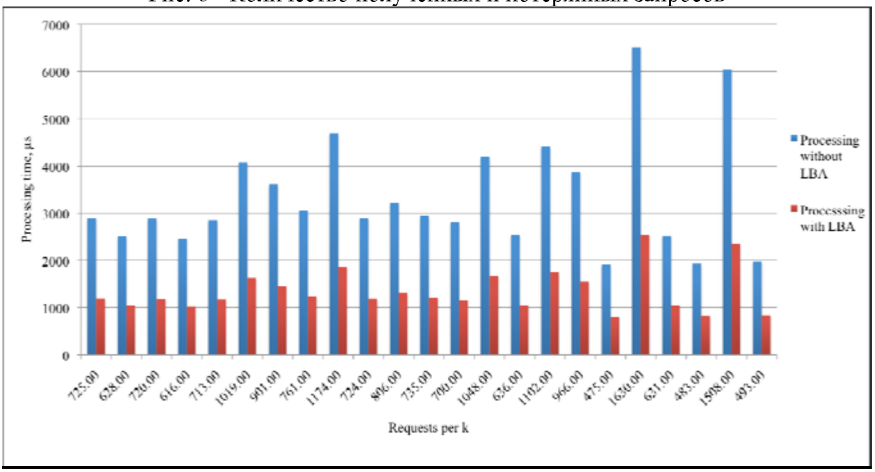


Рис.7 - Время обработки запросов системой балансировки нагрузки

Рассмотрим выигрыш во времени обработки запросов системой балансировки нагрузки в случае последовательной обработки каждого запроса и обработкой с использованием предложенного алгоритма БН:

Как показано на рис.7 обработка запросов пачками, при использовании предложенного алгоритма, сокращает суммарное время обработки запросов, что позволяет сократить время ответа сети с кластерной архитектурой.

Разработанный динамический алгоритм позволяет непрерывно предоставлять услуги в случае выхода из строя узла сети как следствие непрерывного мониторинга жизнеспособности. Алгоритм позволяет поддерживать динамическое равновесие системы, посредством скоординированных реакций; обеспечивает более высокий уровень производительности системы без необходимости модернизации существующей технологической базы.

Выводы. Предложен динамический алгоритм балансировки нагрузки в сетях с кластерной архитектурой, учитывающий результаты прогноза потока запросов, что позволяет повысить производительность, оперативность принятия решения в условиях высокой сетевой нагрузки. Алгоритм обеспечивает равномерное распределение нагрузки по серверам.

С помощью моделирования, показано, что при использовании разработанного алгоритма достигается выигрыш в уменьшении времени ответа сети на 20%, вероятности потерь на 12% и увеличении пропускной способности на 10% в сравнении с WRR, LARD и CAP за счет более эффективного распределения входящего потока и использования ресурсов.

1. *Атцик А.А., Гольдштейн А.Б.* Построение NGN: IPCC vs. TISPAN. // Мир связи. – 2006. – №4.

2. *T. Schroeder, S. Goddard, B. Ramamurthy,* Scalable web server clustering technologies, IEEE Network. – 2000, pp. 38–45.

3. *A. Kamra, V. Misra, E.M. Nahum*, Yaksha: a self-tuning controller for managing the performance of 3-tiered web sites, in: 12th IEEE International Workshop on Quality of Service, IWQOS 2004, 2004, pp. 47–56.

4. *L. Cherkasova, P. Phaal*, Session-based admission control: a mechanism for peak load management of commercial web sites, IEEE Transactions on Computers 51. - 2002.

5. *V. Cardellini, E. Casalicchio, M. Colajanni, Ph.S. Yu*, The state of the art in locally distributed web-server systems, ACM Computing Surveys. CSUR 34. – 2002, pp. 263–311.

6. *Игнатенко Е.Г.* Адаптивный алгоритм мониторинга загруженности серверов web-кластера в системе балансировки нагрузки/ *Е.Г.Игнатенко, В.И. Бессараб, И.В. Дегтяренко//* Наукові праці Донецького національного технічного університету. Серія «Обчислювальна техніка та автоматизація». Вип. 21 (183). – Донецьк: ДонНТУ, 2011. – 193с. – С. 95-102.

Поступила 10.10.2011р.