

УДК [004.032.34+004.451.2].942:004.738.2

**М. В. Тюлькин**, аспирант, Пермский национальный исследовательский политехнический университет**И. В. Капгер**, Пермская печатная фабрика (филиал ФГУП «Гознак»)**Е. Л. Кротова**, кандидат физико-математических наук, доцент, Пермский национальный исследовательский политехнический университет**Л. Н. Кротов**, доктор физико-математических наук, профессор, Пермский национальный исследовательский политехнический университет

## РАЗРАБОТКА АРХИТЕКТУРЫ И ОРГАНИЗАЦИЯ ИНФОРМАЦИОННЫХ ПОТОКОВ В COMET-СЕРВЕРАХ ДЛЯ WEB-ПРИЛОЖЕНИЙ МОДЕЛИ COMET СО СХемой ВЗАИМОДЕЙСТВИЯ WEBSOCKET. МОДЕЛИ АРХИТЕКТУРЫ COMET-СЕРВЕРА\*

Описаны две базовые модели архитектуры Comet-сервера – простая однопоточковая модель и многопоточковая модель с распределенными функциями. Дополнительно каждый вариант архитектуры анализируется с точки зрения возможной работы под неравномерно распределенной нагрузкой ввиду возможной активной атаки DDOS-типа.

**Ключевые слова:** архитектура программ, высоконагружаемое приложение, информационный обмен, вычислительный поток, Comet-сервер, сокет-сервер, сокет, сокеты Беркли, клиентсерверное приложение.

Однопоточковая модель, представленная на рис. 1, довольно проста в реализации и понимании. Она состоит из одного потока, который выполняет все функции. На этапе инициализации основной поток инициализирует все структуры данных в памяти ( $K$ ,  $H$ ,  $S$ ), интерфейс взаимодействия ( $I$ ) и слушающие сокеты ( $S_s$ ,  $S_k$ ).

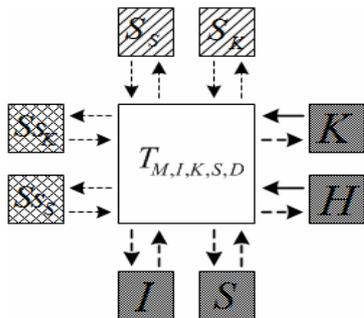


Рис. 1. Однопоточковая модель

Затем запускает бесконечный цикл, в котором поток переходит в состояние ожидания одного или нескольких событий на каком-либо сокете или интерфейсе. При переходе в состояние ожидания операционная система (далее ОС) сервера понижает приоритет процесса Comet-сервера до минимального, тем самым высвобождая вычислительные ресурсы ЭВМ-сервера и возлагая на себя задачу по отслеживанию событий, что достигается соответствующими функциями конкретного языка программирования, реализующего данную программу. При возникновении в слушающих сокетах события на открытие соединения поток открывает сокет ( $s_s$  или  $s_k$ ) для обмена данными с соответствующим удаленным процессом. В случае возникновения события на сокете данного

типа обрабатываются полученные данные или сокет закрывается в зависимости от типа события. Стоит отметить, что через определенные интервалы времени ОС необходимо пробуждать поток, генерируя событие тайм-аута, чтобы он выполнил сервисные операции – очистку истории от старых сообщений.

Главное достоинство данной модели – простота – является ее главным недостатком: на практике при большой нагрузке поток никогда не переходит в спящее состояние, при этом поглощая ресурсы одного вычислительного ядра. Достигнув потолка 100%-й нагрузки на одно ядро, процесс Comet-сервера перестает справляться с нагрузкой, в результате чего появляются временные задержки в работе всей программы, ощутимые всеми сторонами информационного обмена (клиентами и web-серверами). Указанную нагрузку смогут создать или ей способствовать:

- один или несколько клиентов, управляемые злоумышленником, постоянно посылающие запросы на модификацию своего статуса и получение истории сообщений;
- скомпрометированный web-сервер, рассылающий сообщения в большом количестве через Comet-сервер;
- скомпрометированный интерфейс, постоянно делающий запросы статусной информации;
- одновременное подключение большого количества клиентов, сопровождающееся стандартной пересылкой статуса и запроса истории.

### Многопоточковая модель

Более логичным и эффективным решением будет распределение функций одного потока модели из предыдущего раздела по отдельным потокам. При этом с учетом специфики задач Comet-сервера потре-

буется переплетение двух парадигм многопоточного программирования [1, с. 27]:

- производители и потребители – процесс принимает на вход выходные данные своего предшественника и производит входные для своего последователя;
- клиенты и серверы – клиентский процесс посылает запрос серверному процессу, а серверный ожидает запроса от клиента и затем действует в соответствии с этим запросом.

Данная модель изображена на рис. 2.

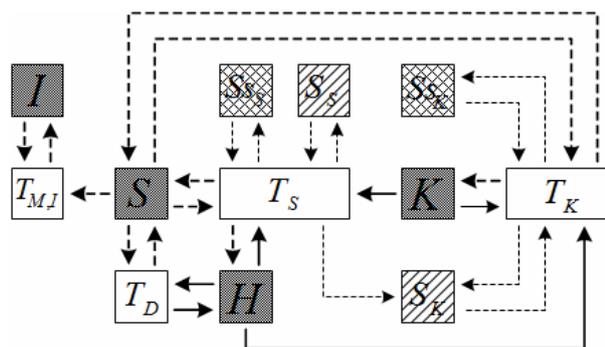


Рис. 2. Многопоточковая модель

Как и в предыдущем случае, на этапе инициализации главный поток процесса программы инициализирует все структуры данных, слушающие сокеты и интерфейс, затем он порождает три потока, каждый из которых выполняет свою функцию в бесконечном цикле, переходя в состояние ожидания, если нет событий, требующих обработки. После этого главный поток берет на себя функцию обслуживания интерфейса взаимодействия, переходя в спящий режим до возникновения какого-либо события в интерфейсе.

Клиентский поток  $T_K$  находится в состоянии ожидания, пока не произойдет событие в слушающем сокете  $S_K$  или одном из клиентских  $S_K$ . Если пришел запрос на соединение в  $S_K$ , то поток открывает новый сокет  $S_K$ . Если пришли данные о статусе в  $S_K$ , то поток модифицирует данные в структуре  $K$  в блокирующем режиме. Однако читать из данной структуры клиентский поток может без ее блокировки, т. е. даже если она заблокирована серверным потоком, поскольку только он модифицирует в ней данные, а следовательно, во время длительной процедуры чтения он может быть уверен, что данные не будут модифицированы. Затем пользовательский поток обновляет статусные данные в  $S$  в блокирующем режиме, поскольку в  $S$  могут писать сразу три потока. Если в сокет  $S_K$  прислан запрос на получение истории сообщений, то клиентский поток в блокирующем режиме просматривает историю и высылает сообщения клиенту. Если в сокете  $S_K$  произошло событие разъединения, то клиентский поток закрывает данный сокет и в блокирующем режиме обновляет структуры  $S$  и  $K$ .

Работа серверного потока  $T_S$  аналогична работе клиентского за исключением того, что в сокет  $S_S$

приходят сообщения от web-сервера с инструкциями по их рассылке. Получив такое сообщение, серверный поток записывает его в историю в блокирующем режиме, затем также в блокирующем режиме путем чтения получает дескрипторы сокетов клиентов из структуры  $K$  и рассылает сообщения клиентам через сокеты  $S_K$  в неблокирующем режиме.

Сервисный поток  $T_D$  большую часть времени находится в состоянии ожидания, поскольку пробуждается ОС только по таймеру с целью очистки истории от старых сообщений и обновления статусных данных в блокирующем режиме. Запись в структуру  $H$  (процедура удаления) отмечена на рис. 1 как длительная, поскольку сопряжена с чтением.

Такая организация элементов Comet-сервера значительно трудней в реализации, но значительно эффективней в работе (по закону Амдаля [2]), поскольку распределяет нагрузку по решению задач Comet-сервера на вычислительные ресурсы ЭВМ, так как ОС может разнести выполнение потоков по разным вычислительным ядрам [3, с. 80]. По сравнению с предыдущим данное решение обладает рядом достоинств:

- полностью независимая работа программы от нагрузок на интерфейс;
- одновременное подключение большого количества клиентов по стандартной процедуре создаст помехи, выражающиеся в замедлении работы только относительно передачи статуса и получения истории уже подключенным клиентам. Однако уже подключенные клиенты будут продолжать получать без помех новую информацию от web-сервера;

– передача большого количества сообщений от web-сервера создаст нагрузки только на серверный поток и через какой-то временной интервал – на сервисный поток, однако клиенты не заметят торможения работы программы в отношении передачи статуса и получения истории сообщений. Сообщения, которые серверный поток не успевает обработать, будут становиться в очередь на уровне сокета.

Данная модель не устраняет полностью торможение всей программы при единственной атаке, создающей нагрузку на выполнение одной функции Comet-сервера, но делает эту атаку значительно менее влиятельной или незаметной для выполнения других функций.

#### Библиографические ссылки

1. Эндрюс Г. Р. Основы многопоточного, параллельного и распределенного программирования : пер. с англ. – М. : Изд. дом «Вильямс», 2003. – 512 с. : ил.
2. Эффективные методы распараллеливания. Закон Амдаля. Август, 2004. – URL : <http://www.rsdn.ru/article/baseserv/RUThreadingMethodology.xml#ELGAC> (дата обращения: 22.03.2012).
3. Митчелл М., Оулдем Д., Самьюэл А. Программирование для Linux. Профессиональный подход : пер. с англ. – М. : Изд. дом «Вильямс», 2003. – 288 с. : ил.

*M. V. Tyulkin*, Post-graduate, Perm National Research Polytechnic University

*I. V. Kapper*, Perm Printing Factory (branch) of the Federal State Unitary Enterprise "GOZNAK"

*E. L. Krotova*, PhD (Physics and Mathematics), Associate Professor, Perm National Research Polytechnic University

*L. N. Krotov*, DSc (Physics and Mathematics), Professor, Perm National Research Polytechnic University

### Development of Architecture and Organization of Information Flows in Comet-Servers for Web-Applications Based on COMET Model with Interaction Scheme "WebSocket Streaming". Models of Comet-Server Architecture

*The paper describes two basic models of Comet-Server architecture, a simple single-threaded model and multi-threaded model with distributed functions. In addition, each version of the architecture is analyzed in terms of the possible non-uniformly distributed load operation due to possible active DDOS attacks.*

**Key words:** program architecture, high-rate application, information exchange, computational flow, Comet-server, server socket, socket, Berkeley sockets, client-server application.

УДК 681.5:621.315.5:621.311

**А. Л. Ахтулов**, доктор технических наук, профессор, Сибирская государственная автомобильно-дорожная академия, Омск

**Л. Н. Ахтулова**, кандидат технических наук, доцент, Сибирская государственная автомобильно-дорожная академия, Омск

**Н. Н. Петухова**, аспирант, Сибирская государственная автомобильно-дорожная академия, Омск

**С. И. Смирнов**, кандидат физико-математических наук, доцент, докторант, филиал «Тобольский индустриальный институт» Тюменского нефтегазового университета

## АЛГОРИТМ ОЦЕНКИ ВЛИЯНИЯ СВОЙСТВ СЕГНЕТОЭЛЕКТРИКА С РАЗМЫТЫМ ФАЗОВЫМ ПЕРЕХОДОМ ПРИ РЕШЕНИИ ЗАДАЧ САПР СИСТЕМ ЭЛЕКТРОСНАБЖЕНИЯ

*Данная статья посвящена разработке систем автоматизации проектирования на основе свойств сегнетоэлектриков в области фазовых переходов. Рассматривается задача создания модели исследования сегнетоэлектрических материалов для выявления их параметров и возможного использования в качестве компонентов при проектировании систем электропитания.*

**Ключевые слова:** система автоматизации проектирования, сегнетоэлектрики, размытый фазовый переход, энергосистема, детализация.

**В** работе [1] авторы показали, что применение систем автоматизации проектирования (САПР) при производстве элементов энергетических систем позволяет создавать оптимальные и эффективные составы этих систем. Одним из элементов энергетических систем являются выпрямители, предназначенные прежде всего для питания нагрузок постоянным током, а через инверторные преобразователи – для электроснабжения потребителей первой категории.

Наиболее распространенным элементом выпрямителя является конденсаторный фильтр, материалы для изготовления которого должны иметь резко выраженные нелинейные электрические свойства. Такими свойствами обладают сегнетоэлектрики, представляющие собой твердые растворы.

Одной из основных характеристик сегнетоэлектриков является коэффициент нелинейности, равный отношению максимального значения диэлектрической проницаемости при некоторой максимальной для данного материала напряженности электрического поля к начальному значению диэлектрической проницаемости.

Значения коэффициентов нелинейности в переменном поле варьируются в широком диапазоне (4...50). При увеличении приложенного напряжения

диэлектрическая проницаемость, а следовательно, и электрическая емкость возрастают, а затем, после достижения максимума, снижаются, что является основным свойством вариконды, для которого электрическая емкость и коэффициент линейности сильно зависят от температуры.

Высокие нелинейные свойства позволяют применять вариконды для умножения частоты, стабилизации тока и напряжения, а также для преобразования синусоидального напряжения в импульсное, усиления электрической мощности и т. д.

В работе [2] исследованы основные свойства сегнетоэлектриков на основе пирониобата кадмия  $Cd_2Nb_2O_7$ , танталата калия-лития  $K_{0,9}Li_{0,1}TaO_3$ , германата свинца  $Pb_3Ge_3O_{11}$  и магнониобата свинца  $Pb(Mg_{1/3}Nb_{2/3})O_3$ .

Кристаллы магнониобата свинца  $Pb(Mg_{1/3}Nb_{2/3})O_3$  характеризуются размытым фазовым переходом со средней температурой Кюри  $T_0^{cp} \gg 270$  К (температура Кюри сегнетоэлектрического перехода  $T_c = 273 \pm 3$  К). В параэлектрической фазе ( $T_a \approx 600$  К) кристаллы имеют кубическую симметрию (точечная группа  $O_h$ ). Симметрия сегнетоэлектрической фазы точно не установлена. По данным комбинационного рассеяния света, симметрия по-