

УДК 539.216:546.824-31

КООПЕРАЦИОННЫЙ МЕТОД ПЛАНИРОВАНИЯ ЗАПРОСОВ В СИСТЕМАХ УПРАВЛЕНИЯ ИДЕНТИФИКАЦИОННЫМИ ДАННЫМИ

П.Ю. ПИНЮТА, В.В. БАХТИЗИН

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь*

Поступила в редакцию 28 июля 2014

Рассмотрен ряд методов планирования задач. Произведен их сравнительный анализ с точки зрения адаптации к системам управления идентификационными данными. Предложен кооперационный метод планирования запросов в системах управления идентификационными данными. Для уменьшения количества блокировок запросов предложен алгоритм подсчета наилучшего момента запуска долгосрочных запросов на основе истории запросов к целевой системе. Приведены результаты экспериментального исследования.

Ключевые слова: идентификационные данные, кооперационный метод планирования, распределение задач.

Введение

Одним из главных элементов информационной инфраструктуры является система управления идентификационными данными (СУИД). Она обеспечивает ролевое управление доступом к ресурсам на основе единой политики безопасности и должностных обязанностей сотрудника. Безопасность любого крупного бизнеса опирается на централизованное управление идентификационными данными совместно с корпоративной стратегией безопасности, обеспечением защиты корпоративных сетей и межсетевых взаимодействий [1].

Внедрение на предприятии СУИД позволяет в значительной степени минимизировать риски, связанные с корпоративной безопасностью. С ростом числа корпоративных приложений некоторые запросы СУИД способны заблокировать работу целевой системы (ЦС) на некоторое время. Под ЦС следует понимать систему, дающую авторизованному пользователю определенные права и привилегии. Количество пользователей может варьироваться от сотен до нескольких десятков тысяч. ЦС делегирует СУИД механизмы управления идентификационными данными пользователей. Процесс передачи этих механизмов называется интеграцией ЦС в СУИД. Примерами ЦС служат операционные системы, базы данных, системы электронного документооборота и т.п.

При использовании современных СУИД риск простоя оборудования и персонала остается высоким. Для его уменьшения в статье предлагается кооперационный метод планирования запросов в СУИД.

Теоретический анализ

Каждый запрос, пришедший от пользователя, делится на две категории: краткосрочный и долгосрочный. Под краткосрочным запросом понимается такой запрос, время выполнения которого незначительно, до одной минуты, а приоритет выполнения низкий. Например, добавление, удаление, блокирование пользователя, выделение ресурсов пользователю и т. д.

Приоритет долгосрочного запроса выше. Его обработка занимает больше ресурсного времени ЦС (от нескольких минут до нескольких дней в зависимости от типа ЦС и количества

пользователей). Это связано с тем, что для обработки запроса требуется опрос большого числа серверов со значительным числом пользователей. Одним из главных требований к выполнению подобных запросов является требование к непрерывности работы долгосрочных запросов. Оно продиктовано тем, что изменение идентификационных данных в ЦС во время работы долгосрочных запросов крайне нежелательно. Примерами таких запросов могут являться аудит ЦС, генерация различных отчетов и т.п.

В настоящее время на предприятиях при управлении проектами, организации мультизадачности, разработке программного обеспечения, тестировании широко применяется планирование. Однако внедрение в СУИД методов планирования из большинства областей является нецелесообразным. Например, такие методы области управления проектами, как мозговой штурм, диаграммы Ишикавы, диаграммы Ганта, анализ критического пути (PERT графы), необходимы для упрощения работы сотрудников, но не автоматизации всего процесса планирования [2]. По этой причине их адаптация и внедрение в СУИД непродуктивны.

Для операционных систем все разработанные методы планирования основываются на следующих трех алгоритмах [3]: невытесняющем, вытесняющем, кооперационном. Они изначально разрабатывались с целью автоматического управления процессами в компьютерах. Рассмотрим возможность адаптации указанных алгоритмов планирования к СУИД.

Суть работы невытесняющего алгоритма планирования заключается в следующем. Один процесс загружается на выполнение. Ресурс освобождается только после полного выполнения данного процесса [4]. В контексте СУИД это видится как недостаток, так как запросы часто требуют согласования с пользователями. Следовательно, все остальные запросы из очереди будут находиться в состоянии ожидания, и общая скорость работы снизится.

Альтернативным невытесняющему алгоритму планирования является вытесняющий алгоритм. При его использовании операционная система сама передает управление от одной выполняемой программы другой в случае появления некоторых событий [4]. Отдельно стоит отметить, что момент передачи ресурсного времени от одной задачи другой не зависит от состояния выполняемой задачи. Это означает, что она может быть прервана в любой момент времени. При адаптации к СУИД указанная особенность алгоритма крайне негативно влияет на работу с ЦС. При поступлении нового запроса выполняющийся может быть прерван. А это противоречит требованию к непрерывности работы долгосрочных запросов.

Основное отличие кооперационного алгоритма от вытесняющего заключается в том, что следующая задача выполняется только после того, как текущая явно объявит себя готовой отдать ресурсное время [4]. Таким образом, следующая задача из очереди выполняется только после того, как текущая либо завершится, либо приостановится в связи с синхронизацией с каким-либо объектом. Применение данного алгоритма в классическом виде связано с монополизацией процессора, что, в общем, является недостатком. Однако с точки зрения СУИД это становится достоинством: управляющий процесс блокирует работу всех запросов к ЦС, удовлетворяя требование к непрерывности работы долгосрочных запросов.

На основе кооперационного алгоритма был предложен кооперационный метод планирования запросов к ЦС в СУИД. Его суть заключается в следующем. Любой запрос монополизует ЦС, блокируя выполнение других запросов. Освобождение ЦС происходит при завершении обработки запроса либо его приостановке. Так как разные категории запросов имеют различный приоритет, а обработка долгосрочного запроса не может быть прервана другим запросом, число блокировок краткосрочных запросов в период выполнения долгосрочного запроса должно быть минимальным. Для этого момент запуска долгосрочных запросов необходимо смещать. При этом он должен определяться наименьшим количеством заблокированных задач за время от начала до завершения работы долгосрочного запроса. С этой целью следует выбирать такой момент времени, при котором предполагаемая загрузка ЦС минимальна. Для расчета указанного момента предлагается алгоритм подсчета наилучшего момента запуска долгосрочных запросов на основе истории запросов к ЦС. Согласно данному алгоритму, наилучший момент времени определяется выражением:

$$\frac{\sum_{i=1}^N W_{\Pi}(\tau + \frac{T_{\Pi}}{N} \cdot i)}{N} \rightarrow \min_{\tau \in [0, T_{\Pi}]}, \quad (1)$$

где $W_{II}(\tau + \frac{T_{II}}{N} \cdot i)$ – функция предполагаемой на основе истории запросов загрузки ЦС; T_{II} – предполагаемое время, необходимое для выполнения долгосрочного запроса; τ – момент времени запуска долгосрочного запроса; N – количество разбиений временного отрезка $[\tau, \tau + T_{II}]$; T_d – наибольший допустимый момент времени запуска долгосрочного запроса. Момент времени окончания обработки долгосрочного запроса может быть больше T_d .

В качестве метода минимизации функции выбран метод перебора, так как его применение для данной задачи является наиболее простым. Алгоритм подсчета наилучшего момента запуска долгосрочных запросов выглядит следующим образом:

- 1) выбрать некоторое число n , и разделить отрезок $[0, T_d]$ на n равных частей;
- 2) инициализировать счетчик $m = 0$ и значение $\tau_m = (T_d \cdot m)/n$;
- 3) если $m = n$, перейти к шагу 6. Иначе к шагу 4;
- 4) если для всех $\tau_k = (T_d \cdot k)/n$, $k = 0, 1, 2, \dots, n$ выполняется условие:

$$\frac{\sum_{i=1}^N W_{II}(\tau_m + \frac{T_{II}}{N} \cdot i)}{N} \leq \frac{\sum_{i=1}^N W_{II}(\tau_k + \frac{T_{II}}{N} \cdot i)}{N} \quad (2)$$

перейти к шагу 6; иначе к шагу 5;

- 5) увеличить на 1 счётчик m . Обновить значение $\tau_m = (T_d \cdot m)/n$. Перейти к шагу 3;
- 6) момент времени τ_m является искомым значением. Завершить алгоритм.

Для развертывания СУИД с разработанным кооперационным методом планирования запросов к ЦС следует выполнить следующие действия: 1) установить СУИД; 2) интегрировать в нее ЦС; 3) сформировать историю загрузки ЦС. Под историей загрузки понимается зависимость количества запросов, запущенных ранее, от времени. Последнее действие производится автоматически в течение некоторого времени после интеграции ЦС в СУИД. Это необходимо для корректного функционирования алгоритма. Если данная история не сформирована, то улучшенная СУИД будет работать так же, как и не улучшенная. Таким образом, достоинства разработанного метода проявляются по прошествии некоторого времени.

Методика эксперимента

Для проведения эксперимента необходимо развернуть СУИД. В качестве ЦС был выбран домен Windows. Количество пользователей ЦС составляет 1000. Домен расположен в одном часовом поясе, пользователи работают в одну смену. Пусть рабочее время пользователей с 9 до 18 ч. В ходе эксперимента следует выполнить следующие действия.

1. Запуск в рабочие часы краткосрочных запросов на предоставление пользователям некоторых прав. Частота запускаемых краткосрочных запросов выражается формулой:

$$f(t) = \begin{cases} C, & t \in [9, 18] \\ 0, & t \notin [9, 18] \end{cases}, \quad (3)$$

где $f(t)$ – частота запускаемых краткосрочных запросов; t – текущее время; C – константа, равная 60 запросов в час для данного эксперимента.

2. Запуск аудита ЦС. Время обработки данного запроса составляет около одного часа для данной ЦС. Момент запуска определяется началом рабочего дня администратора ЦС.

3. Сбор данных о заблокированных запросах. Фиксация количества запросов в очереди в момент времени t , где t находится на отрезке $[0, 24]$.

4. Определение реальной загрузки ЦС. Для этого необходимо выявить, какой процент времени в течение промежутка Δt ЦС обрабатывала запросы. Для вычисления значения реальной загрузки ЦС следует использовать следующую формулу:

$$W_p(t) = \frac{\sum_{i=1}^K W_3(t + \frac{\Delta t}{K} \cdot i)}{K} \cdot 100 \% \quad (4)$$

где $W_p(t)$ – реальная нагрузка ЦС; t – текущее время; Δt – промежуток времени; $w_3(t + \frac{\Delta t}{K} \cdot i)$ – функция загрузки ЦС, принимающее значение 1, если ЦС обрабатывает запрос в указанный момент времени, или 0, если запросов на обработке нет; K – количество разбиений временного отрезка $[t, t + \Delta t]$.

5. Развертывание СУИД с разработанным методом планирования запросов к ЦС и повторение шагов 2, 3, 4 алгоритма эксперимента, используя улучшенную СУИД.

Анализ результатов эксперимента

На рис. 1 изображен график зависимости количества заблокированных запросов (L) от времени суток (t) для СУИД без улучшений. На этом графике заметен резкий рост количества заблокированных запросов в период работы аудита. После окончания обработки аудита ЦС была загружена ещё около часа, так как она обрабатывала запросы из очереди, сформировавшейся за время работы аудита. Как итог, для несложной и небольшой ЦС время ожидания предоставления прав некоторым пользователям составило более одного часа.

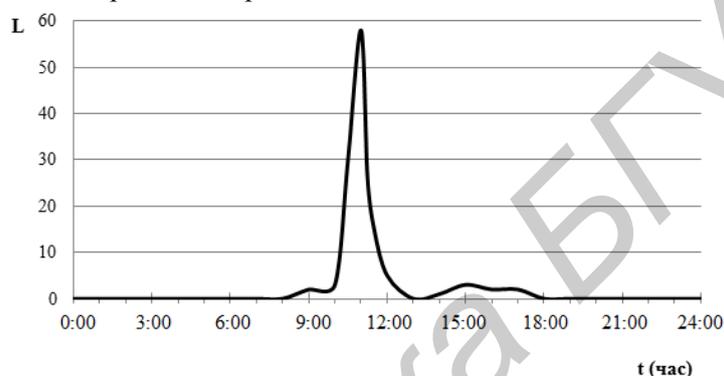


Рис. 1. График зависимости количества заблокированных запросов от времени суток

Применяя формулу (4) к полученным данным, была получена зависимость загрузки ЦС от времени суток. График указанной зависимости представлен на рис. 2. Как видно на графике, при обработке краткосрочных запросов загрузка ЦС не превышает 3%. В момент фактического запуска аудита (10:00) заметен всплеск загрузки ЦС. После завершения обработки этого запроса ЦС продолжала быть загруженной в течение одного часа, обрабатывая запросы из очереди.

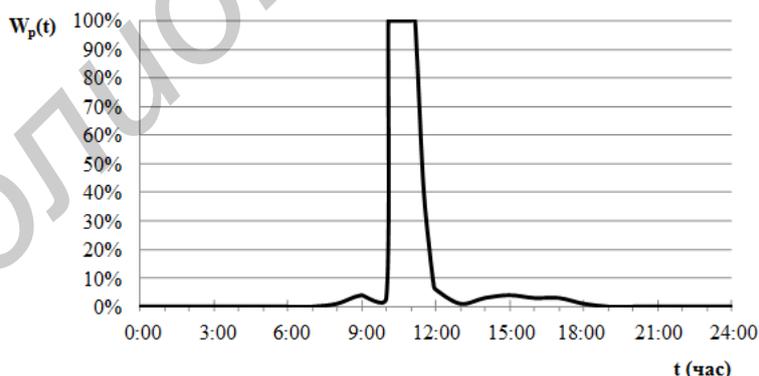


Рис. 2. График зависимости загрузки ЦС от времени суток

В ходе следующей части эксперимента с применением улучшенной СУИД были получены данные, значительно отличающиеся от описанных ранее. График зависимости количества заблокированных запросов от времени суток изображен на рис. 3.

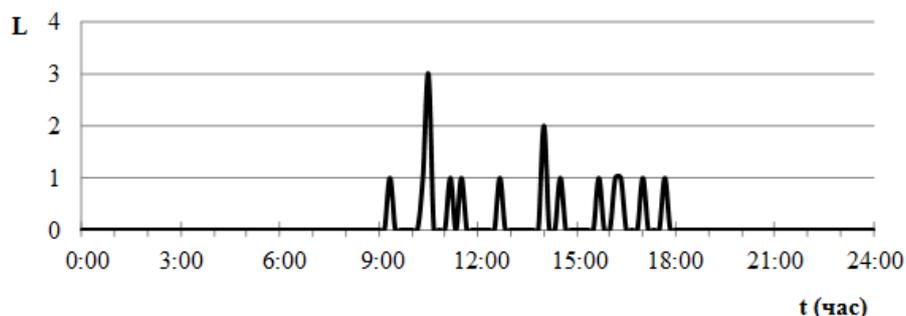


Рис. 3. График зависимости количества заблокированных запросов от времени суток с применением улучшенной системы управления идентификационными данными

При рассмотрении данного графика легко обнаружить, что отсутствует большая очередь в момент инициирования долгосрочного запроса администратором ЦС. Это связано с тем, что, применяя разработанный метод, СУИД сместила момент запуска обработки долгосрочного запроса на более позднее время. Данный запрос выполнялся в такое время, когда число заблокированных краткосрочных запросов ожидалось минимальным. Таким образом, продолжительность обработки всех краткосрочных запросов находилась в рамках обычных значений, то есть до одной минуты.

Для демонстрации позитивных изменений целесообразно показать зависимость загрузки ЦС. Полученный график изображен на рис. 4.

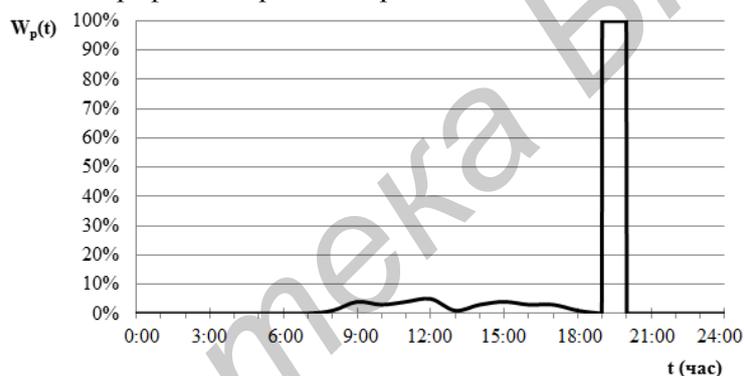


Рис. 4. График зависимости загрузки ЦС от времени суток с применением улучшенной системы управления идентификационными данными

На приведенном графике видно, что загрузка ЦС в момент фактического запуска обработки долгосрочного запроса (19:00) резко поднимается до 100 % так же, как и на рис. 2. Однако в отличие от ранее полученного графика, сразу после завершения обработки долгосрочного запроса, загрузка резко упала до нуля. Данный факт положительно влияет на общую производительность ЦС, так как период максимальной загрузки ЦС занимает значительно меньший промежуток времени. Тем самым у ЦС остаётся больше ресурсов на выполнение своих основных функций.

Заключение

В ходе эксперимента была рассмотрена простая ЦС с небольшим количеством пользователей. Сотрудники работают в одну смену. Распределение моментов времени запуска краткосрочных запросов не учитывало социальных, культурных и других факторов. Эксперимент показал, что для описанной ЦС применение разработанного кооперационного метода планирования в СУИД уменьшает время ожидания сотрудниками запрошенных ресурсов. Данный факт положительно влияет на производительность труда на предприятии, так как персоналу больше не требуется ожидать окончания работы долгосрочных запросов, таких как аудит ЦС. Так же применение разработанного метода уменьшает период максимальной загрузки ЦС, что минимизирует влияние на выполнение непосредственных функций ЦС.

COOPERATIVE METHOD OF REQUEST PLANNING IN IDENTITY MANAGEMENT SYSTEMS

P.Yu. PINIUTA, V.V. BAKHTIZIN

Abstract

A number of planning methods was reviewed. The comparative analysis of the methods from the identity management adaptation point of view was produced. Cooperative method of request planing was created. Algorithm of best longterm request execuion time calculation was created for minimization of blocked requests. It was based on history of requests to a target system. The results of experimental research were shown.

Список литературы

1. *Birch D.* Digital Identity Management: Technological, Business and Social Im. ВКР, 2007.
2. *Федюкин В.К.* Управление качеством процессов. СПб, 2005.
3. *Таненбаум Э.* Современные операционные системы: 3-е изд. СПб, 2010.
4. *Олифер В.Г., Олифер Н.А.* Сетевые операционные системы: учебник для вузов. СПб, 2008.