УДК 519.213:257, 519.657

Е.И. Максименко, Е.В. Крылов, В.К. Аникин

ЗАДАЧА ОПТИМИЗАЦИИ НАГРУЗКИ НА СЕРВЕР С ТОЧКИ ЗРЕНИЯ МИНИМИЗАЦИИ ВРЕМЕНИ ОТКЛИКА

Введение

При эксплуатации почти всех систем наступает момент, когда рассчитанные ранее возможности системы уже не удовлетворяют текущие потребности. Тогда постает выбор: заменить систему более мощной (способной удовлетворить потребности), что чревато финансовыми, временными расходами и иногда вообще не реализуемо; или оптимизировать нагрузку на систему, что более реально, так как чаще всего перегрузка системы проявляется не на всем временном диапазоне ее работы, а является локальной. В текущей статье мы рассмотрим оптимизацию работы сервера, обслуживающего систему планирования ресурсов предприятия (ERP), который автоматизирует выполнение определенного числа бизнес процессов (продаж, закупок, поставок . . .), которые выполняются в течение суток и создают определенной нагрузку на сервер. Характеристики бизнес процессов заранее известны.

Описание задачи

Есть сервер, способный выполнять определенное число операций (транзакций) в час. Есть 24-часовой график загрузки сервера (операций в час), причем в некоторые моменты наблюдается перегрузка. Загрузка сервера формируется набором задач определенной длительности, определенным образом расположенных во времени, которые требуют определенного числа операций (транзакций) сервера в час. Задачи неразрывны. Так как задачи инициализирует и контролирует человек, то критерием удовлетворения потребностей является приемлемое время отклика сервера на каждое действие (операцию, транзакцию) человека. То есть время выполнения одной операции должно быть таким, чтобы человеку не казалось, что он ждет сервер. По мнению специалистов, это время равно двум секундам [1].

Детализация задачи

Для решения задачи необходимо оптимизировать расположение задач во времени (учитывая, что у каждой задачи есть допустимые границы начала и конца выполнения) таким образом, чтобы время выполнения одной операции (время отклика сервера) было меньше заданного (2 сек).

Так как сервер может выполнять множество операций в час, одна операция незначительно загружает сервер и операции независимы друг от друга, то можем представить сервер в виде системы M/M/1 [2].

 ISSN 1562-9945

Тогда время отклика сервера, то есть время выполнения одной операции T будет равно $T = 1/(\mu - \lambda)$. Исходя из этого, зная граничное значение времени отклика $T_{гран}$, мы можем найти максимально допустимую нагрузку $\lambda_{гран} = \mu - 1/T$. В итоге задача сводится к оптимизации расположения задач во времени с учетом того, что нагрузка в любое время суток $\lambda(t)$ должна быть меньше либо равна $\lambda_{гран} (\forall t \in [1, \ldots, 24] : \lambda(t) \leq \lambda_{гран})$.

Имеется N задач, каждая из которых характеризуется временем начала выполнения ($t_{на\ чi}$), временем выполнения ($t_{вып\ i}$) и нагрузкой на процессор λ_i, где $i = 1, 2, 3, \ldots, N$. То есть

$$\lambda_i(t) = \begin{cases} \lambda_i, \text{ если } t_{нач\ i} \leq t \leq t_{нач\ i} + t_{вып\ i} \\ 0, \text{ если } t < t_{нач\ i},\ t > t_{нач\ i} + t_{вып\ i} \end{cases} \quad (1)$$

На каждую из задач наложено ограничение минимального времени начала выполнения ($t_{мин i}$) такое, что $t_{нач\ i} \geq t_{мин\ i}$ и максимального времени конца выполнения ($t_{макс i}$) такое, что $t_{нач i} + t_{вып i} \leq t_{макс i}$.

Таким образом, график загрузки с ограничениями системы можно оценить следующими параметрами:

i – № задачи

$t_{нач i}$ – время начала (ч)

$t_{вып i}$ – время выполнения (ч)

λ_i – нагрузка (опер/ч)

$t_{мин i}$, – минимальное время начала (ч)

$t_{макс i}$, – максимальное время конца (ч)

Текущая нагрузка на сервер $\lambda(t)$ будет равна:

$$\lambda(t) = \sum_{i=1}^{N} \lambda_i(t). \quad (2)$$

Для того чтобы распределить нагрузку, как писалось выше, необходимо, чтобы $\forall t \in [1, \ldots, 24] : \lambda(t) \leq \lambda_{гран}$, то есть необходимо разгрузить сервер во время, когда загрузка превышает максимально дозволенною, переместив некоторые задачи в менее нагруженные места. Для этого воспользуемся следующим алгоритмом.

Для начала проверим, возможно ли в принципе, без учета ограничений, вместить суммарную трудоемкость всех задач в 24-х часа так, чтобы она не превышала граничную. То есть $\frac{\sum_{i=1}^{N} \lambda_i(t)}{24} < \lambda_{гран}$. Если условие не выполняется, то оптимизацию до заданного уровня можно даже не пытаться проводить.

Так как наибольшим образом на загрузку сервера влияет задача с самой большой нагрузкой λ_i, то сортируем порядок рассмотрения задач по убыванию нагрузки на сервер, то есть $\lambda_i > \lambda_{i+1}$.

Находим $t_{экстр}$ такое, что $\lambda(t_{экстр}) = \max\{\lambda(t)\} = \lambda_{экстр} > \lambda_{гран}$. Если время перегрузки не найдено, то есть удовлетворяется условие $\forall t \in [1, \ldots, 24] : \lambda(t) \leq \lambda_{гран}$, то задача считается выполненной.

Выбираем первую из отсортированных задач $i = 1$.

Передвигаем выбранную задачу в разрешенном диапазоне таким образом, чтобы она не выполнялась во время $t_{экстр}$ и, при этом, во время ее выполнения не возникал пик, такой же как $\lambda_{экстр}$ или больше. Под передвижением понимается нахождение перебором нового $t_{нач}$, удовлетворяющего все заданные ограничения (включая ограничения самой задачи). В случае если есть несколько удовлетворительных расположений задачи, то критерием оптимальности расположения является промежуток времени (длительностью $t_{выпi}$) с минимальной суммарной нагрузкой на сервер, не учитывая нагрузки текущей задачи.

$$\begin{cases} t_{нач\,i} = t_{мин\,i} + j, \text{ где } j \in [0,\ t_{макс\,i} - t_{мин\,i} - t_{вып\,i}]; \\ t_{нач\,i} + t_{вып\,i} < t_{экстр} < t_{нач\,i};\quad t_{нач\,i} \geq t_{мин\,i};\quad t_{нач\,i} + t_{вып\,i} \leqslant t_{макс\,i}; \\ \forall t \in [t_{нач\,i}, t_{нач\,i} + t_{вып\,i}]: \lambda(t) < \lambda_{экстр}; \\ \sum\limits_{t=t_{нач\,i}}^{t_{нач\,i}+t_{вып\,i}} (\lambda(t) - \lambda_i(t)) \to \min. \end{cases} \quad (3)$$

Если перемещение текущей задачи не дало результата (невозможно переместить задачу, удовлетворив все условия), то выбираем следующую по нагрузке задачу $i = i + 1$ и переходим к п.4, в противном случае запоминаем новое состояние таблицы нагрузки и переходим к п.2.

Если перемещение всех задач не дало результата, значит, задача не может быть решена, то есть сервер не сможет обрабатывать заданные задачи со временем отклика меньше либо равному требуемому.

Постановка задачи

Допустим, мы имеем процессор мощностью 6600 операций в час и хотим получить время отклика две секунды. В таблице 1 представлено распределение нагрузки и ограничения.

Таблица 1

Распределение нагрузки и ограничения

i	$t_{начi}$	$t_{выпi}$	λ_i	$t_{минi}$	$t_{максi}$	i	$t_{начi}$	$t_{выпi}$	λ_i	$t_{минi}$	$t_{максi}$	i	$t_{начi}$	$t_{выпi}$	λ_i	$t_{минi}$	$t_{максi}$
1	7	18	1000	3	23	**6**	15	9	730	14	24	**11**	8	11	210	3	18
2	1	23	990	1	24	**7**	1	17	540	1	18	**12**	6	16	160	1	22
3	6	6	870	2	16	**8**	8	2	530	3	18	**13**	1	23	120	1	24
4	8	15	810	4	24	**9**	1	16	340	2	12	**14**	8	17	110	6	24
5	1	13	740	1	18	**10**	5	9	300	4	24	**15**	8	7	70	8	23

Решение задачи

Для работы со временем реакции 2 секунды (2/3600 часа) сервер с мощностью 6600 операций в час должен быть нагружен не больше чем на $\lambda_{гран} = \mu - 1/T$ =6600-3600/2=4800 оп/час.

Для наглядности анализа представим наши задачи во временно-нагрузочной плоскости в виде прямоугольников в высоту занимающих λ_i единиц нагрузки, а в длину $t_{вып\ i}$ времени выполнения. Задачи расположены вертикально друг над другом, а горизонтально смещены вдоль временной оси так, что начинаются с $t_{нач\ i}$. Сумма высот прямоугольников над конкретной координатой временной оси дает нам суммарную загрузку в указанное время. Суммарную нагрузку представим в виде

диаграммы, на которой проведем линию мат ожидания нагрузки (серая), которая будет показывать минимально возможную нагрузку с геометрической точки зрения (без учета ограничений) и линию $\lambda_{гран}$ (пунктирная). Для удобства раскрасим столбцы так, что более нагруженные будут темнее, а менее нагруженные – светлее. Для иллюстраций будем использовать программу, любезно предоставленную Борщаговским П.Е. В результате получим (рис. 1.):

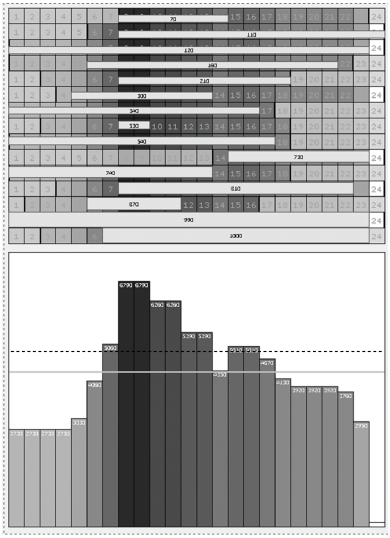

Рис. 1 – Первый проход алгоритма

Как видно из рисунка, максимальная нагрузка 6790 оп/час превышает 4800 оп/час и даже превышает максимальную пропускную способность сервера 6600 оп/час (то есть такая система не то, что не удовлетворяет требование к времени отклика, а вовсе работать не будет), а минимально возможная с геометрической точки зрения нагрузка равна 4238 оп/час и говорит о том, что есть смысл попробовать провести оптимизацию. Оптимизируем полученную нагрузку согласно указанному выше алгоритму.

Как видим, самое нагруженное время (если анализировать график слева на право) – это 8 часов. Берем самую трудоемкую задачу на это время (то есть первую из отсортированной таблицы задачу с нагрузкой 1000). Как видно – это задача не может удовлетворять условие невыполнения во время максимума, так как не помещается ни слева, ни с права от него. По этому, выбираем следующую задачу с нагрузкой 990. Она, как и предыдущая, не может быть перемещена со времени максимума. В результате доходим до третей задачи с нагрузкой 87. Перебрав все ее положения из возможных, приходим к выводу, что минимальная суммарная

нагрузка, с учетом всех ограничений, будет, если задача выполняется, начиная с 2-х часов. В итоге получаем новую таблицу задач, которая в графическом виде показана на рис. 2.:

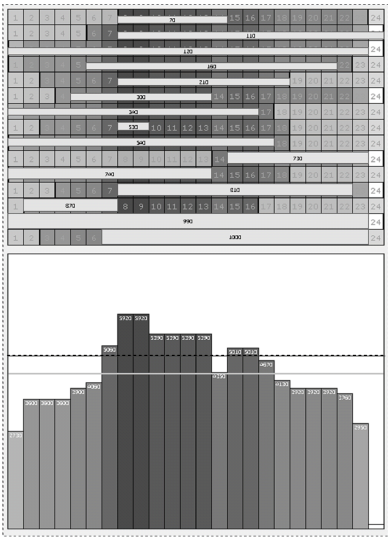

Рис. 2 – Второй проход алгоритма

Находим новый максимум – 5920 оп/ч. Так как он все еще превышает требуемое граничное значение, то продолжаем оптимизацию. Как и в предыдущий раз, максимум - это 8 часов. Первая и вторая задачи не могут быть перемещены, а третья и так находится в оптимальном положении. По этому, анализируем следующие задачи. Четвертая задача дает минимальную нагрузку, если будет начинаться в 10 часов. Это положение удовлетворяет все условия. В результате оптимизации получим следующую картину (рис. 3.):

Как видим, появился новый максимум – 10 часов 5390 оп/час, который все еще больше 4800 оп/час, по этому продолжаем. Первая и вторая задача не могут быть перемещены из времени максимума, третья не влияет на это время. Четвертая и пятая так же не может быть перемещена. В результате находим 10-ю задачу с нагрузкой 30 и, согласно критерию, перемещаем ее так, чтобы она начиналась в 5 часов. Продолжая дальше, согласно указанному алгоритму, доходим до такой картины (рис. 4.):

Как видно из рисунка, время максимума наблюдается в 16 часов, но исходя из ограничений, нет ни одной задачи, которая может быть перемещена так, чтобы не занимать время максимума либо не создавать пики больше текущего максимума, по этому оптимизация остановлена.

 ISSN 1562-9945

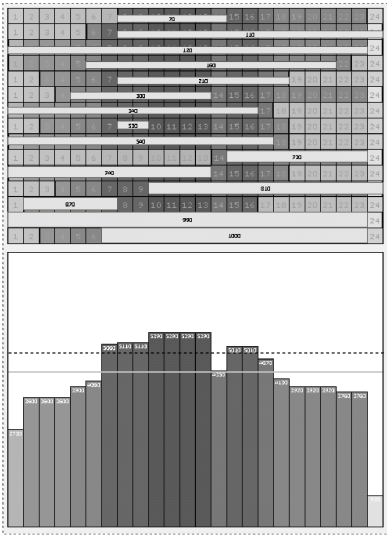

Рис. 3 – Третий проход алгоритма

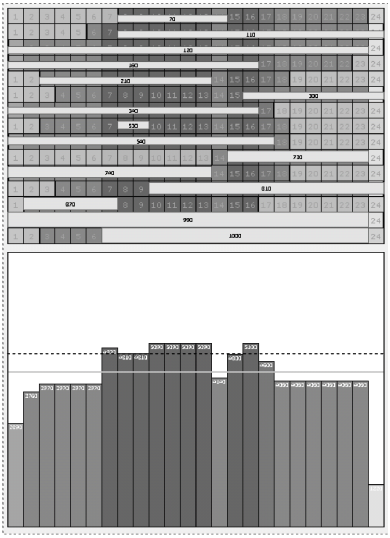

Рис. 4 – Последний проход алгоритма

Вывод

Данная система не может быть оптимизирована для работы со временем отклика 2 сек, но, исходя из того, что максимальная полученная нагрузка 5100 оп/час меньше пропускной способности сервера, то он может выполнять текущие задачи, только со временем $T = 1/(\mu - \lambda) = 1/(6600 - 5100)\lambda 0,000(6)$час $= 2,4$сек. Для работы со временем отклика 2 секунды, данной системе нужен сервер с мощностью $\mu = \lambda_{гран} + 1/T = 6900$ оп/час.

Литература

1. В.В. Крылов, С.С. Самохвалова “Теория телетрафика и ее приложения. Основы теории систем массового обслуживания для задач телекоммуникаций”, издательство BHV, 2005 г., 288 стр., ISBN: 5-94157-569-6.
2. http://www.eventhelix.com/RealtimeMantra/CongestionControl/m_m_1_queue.htm.