ИНФОРМАТИКА

УДК 519.686.1

ИССЛЕДОВАНИЕ ДИНАМИКИ ПОВЕДЕНИЯ АГЕНТОВ В СРЕДЕ РАЗДЕЛЯЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ

канд. техн. наук, доц. Д.О. ГЛУХОВ, Т.М. ГЛУХОВА
(Полоцкий государственный университет)

Объектно-ориентированные системы могут рассматриваться как системы массового обслуживания. Агент в нашем случае – это система с ожиданием и отказами, а событие – отработка определенной функции за конечное время в технологическом цикле обработки запроса клиента. Агент, функционирующий в локальной вычислительной сети, имеет системную очередь запросов и функционирует в двух режимах работы: в режиме образования и разбора очереди и режиме наличия простоя процессора.

Таким образом, в работе делается попытка описания поведения системы агентов нелинейными дифференциальными уравнениями, аппроксимирующими известную модель системы массового обслуживания с очередью запросов. Для оценки вероятностных характеристик вычислительных процессов в объектно-ориентированных средах разработан и реализован программный модуль.

Введение. В настоящее время для создания различного рода программных продуктов широко применяется объектно-ориентированная технология. В основе технологии лежит ряд принципов, формирующих идеологию объектно-ориентированного подхода, к ним относятся: принцип инверсии зависимостей (DIP); принцип релиза (OCP); принцип сегрегации интерфейсов (ISP); принцип подстановки Лесковой (LSP) [1]. Данный подход требует специфической технологии проектирования, который в настоящий момент не мыслится без стандартного языка моделирования UML [2, 3].

Объектно-ориентированный подход находит широкое применение в областях, допускающих абстракции данных на уровне объектов и классов объектов. Одной из областей применения этой технологии является разработка систем управления, функционирующих по принципу событие – состояние. Обычно принимается передача объектам информации о событии в системе посредством сообщений. Однако, из-за задержки, определяемой временем реакции и отработки объектом сообщения, появляется необходимость в использовании спулинга сообщений в очередь. Таким образом, объектно-ориентированные системы могут рассматриваться как системы массового обслуживания и, следовательно, можно говорить о вероятностных характеристиках таких систем (среднее время обслуживания, вероятность отказа и т.д.), определяющих надежность системы.

В условиях развития распределенных систем, мультипроцессорных и мультимашинных, быстрое развитие получают многоагентные системы. В таких системах стоящая перед системой задача решается путем совместного действия программных агентов, каждый из которых решает свою локальную задачу.

Изучение функционирования многоагентных систем поставило ряд вопросов, связанных с моделированием динамики коллектива агентов и способов оценки эффективности построенной структуры агентов [4, 5], разделяющих ресурс времени одного или множества центральных процессорных устройств.

Для оценки эффективности функционирования системы необходимы математические модели, описывающие поведение системы при различных внешних воздействиях, и параметры, характеризирующие производительность, вероятность отказа и т.д. Необходимость в построении таких моделей все более возрастает в условиях тенденции повышения автономности компонентов распределенных объектных и многоагентных систем. При этом возникает проблема координации действий коллектива программ, актуальная для настоящего времени [6, 7].

В данной работе в качестве модели поведения системы агентов используются нелинейные дифференциальные уравнения, аппроксимирующие известную модель системы массового обслуживания с очередью запросов. Переход к непрерывной вероятностной модели динамики агента, более информативной, чем адаптированная модель системы массового обслуживания, является предпочтительным.

Агент в нашем случае рассматривается как система с ожиданием и отказами. В этом случае понятие потока запросов можно рассматривать как поток событий, что позволяет получить ряд интересующих нас параметров. Так, вероятность отказа агента, время ожидания и эффективность дают нам достаточный объем информации для анализа состояния системы в целом.

1. Модель динамики функционирования агента

Рассматриваемая нами событийная система является неэргодической, нестационарной относительно вероятностных характеристик событий и обладает крайне сложной структурой. Поэтому она не допускает прямого моделирования вероятностными, структурными и иными методами, требующими знания внутренней структуры системы [8].

Предлагаемый метод основан на расчете интегральных показателей объектно-ориентированной системы, функционирующей на основе событий. Интегральные показатели описывают суммарный эффект от действия колоссального количества внутренних подпроцессов и событий, протекающих в системе. Под событием мы понимаем отработку определенной функции за конечное время в технологическом цикле обработки запроса клиента. В общем случае цикл обработки запроса можно представить следующим образом (рис. 1).

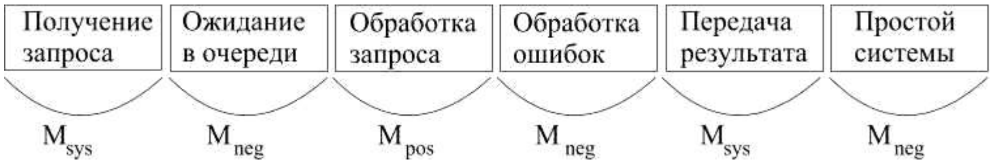


Рис. 1. Цикл обработки запроса

Пусть M – множество классов всех событий системы, а M_{sys} – множество служебных событий; M_{pos} – множество полезных (разрешающих) событий; M_{neg} – множество событий обработки ошибок, конфликтов, ожиданий в очереди:

$$M_{sys} \supset M;\quad M \supset N;\quad M_{pos} \supset M;\quad M_{neg} \supset M;\quad M = M_{sys} \cup M_{pos} \cup M_{neg}.$$

Пусть E – множество всех произошедших в системе событий: $E = \{e_i\}$, $e_i = <m_i, \Delta t_i>$.

Введем альтернативное обращение к элементам объединения: если некоторое событие e есть $e = <m, \Delta t>$, то запись $e.m$ эквивалентна m.

Введем группировку E_{sys}, E_{pos}, E_{neg}:

$$e_i \in E_{sys} \mid e_i.m \in M_{sys};\quad e_i \in E_{pos} \mid e_i.m \in M_{pos};\quad e_i \in E_{neg} \mid e_i.m \in M_{neg}.$$

Все события можно разделить на 2 вида:

- события, длительность которых определена на шкале времени центрального процессора (ЦПУ);
- события, длительность которых определена в параллельном временном пространстве (событие ожидания задачи в очереди).

Так, события полезной обработки, простоя системы, обработки ошибок, системных действий используют время ЦПУ, а событие ожидания задачи в очереди – параллельно некоторой полезной работе процессорного устройства.

Пусть Ψ – множество циклов обработки, порождающих в системе потоки событий:

$$\Psi = \{\psi_k\};\quad k = 0, \ldots, s.$$

Так как событийная объектно-ориентированная система обязательно выполняет сериализацию, то поток циклов обработки может рассматриваться как последовательный поток, характеризующийся интенсивностью поступления циклов обработки $\lambda(t)$.

Предположим, что с каждым ψ_k циклом обработки связан протокол событий системы:

$$Log_k = \{e\};\quad \{e\} \supset E.$$

Тогда суммарное время событий выделенных классов в одном цикле обработки можно определить следующим образом:

$$Tsys_k = \sum_{Log_k} Log_k.t \,\Big|\, Log_k.m \in M_{sys};$$

$$Tpos_k = \sum_{Log_k} Log_k.t \,\Big|\, Log_k.m \in M_{pos};$$

$$Tneg_k = \sum_{Log_k} Log_k.t \,\Big|\, Log_k.m \in M_{neg}.$$

Общее время событий цикла обработки:

$$Tall_k = \sum_{Log_k} Log_k.t \,|\, Log_k.m \in M\,.$$

Агент, функционирующий в локальной вычислительной сети, имеет системную очередь запросов и функционирует в двух режимах работы: в режиме образования и разбора очереди и режиме наличия простоя процессора.

Исходное дифференциальное уравнение, описывающее динамику времени ожидания в очереди запросов, запроса, поступившего в систему в момент времени *t*, можно записать в следующем виде:

$$\frac{dT_{all}}{dt} = \left|\begin{array}{l} \lambda(t)(T_{pos} + T_{sys} + T_{neg}^{proc}) - 1 \quad if \quad T_{all} > 0; \\ otherwise \\ \quad \left|\begin{array}{l} \lambda(t)(T_{pos} + T_{sys} + T_{neg}^{proc}) - 1 \quad if \quad \lambda(t)(T_{pos} + T_{sys} + T_{neg}^{proc}) - 1 > 0. \\ 0 \quad otherwise \end{array}\right. \end{array}\right.$$

Рассмотрим функционирование группы из *s* агентов, разделяющих временной ресурс одного процессора.

Для определения количества задействованных агентов воспользуемся формулой:

$$n = floor\left(\frac{T_{all} + T}{T}\right), \quad n = 0, 1, \ldots, s,$$

где для простоты обозначений примем *T* как время обработки одного запроса.

Таким образом, в процессе обработки находятся одновременно *n* запросов, и все *n* агентов поочередно (последовательно) занимают процессорное время. Это эквивалентно уменьшению интенсивности поступления запросов, приходящейся на одного агента, в *n* раз, а также замедлению времени обработки в *n* раз. В таком случае скорость обработки всей очереди запросов остается неизменной, т.е. получаем следующую систему:

$$\frac{dT_{all\_N}}{dt} = \left|\begin{array}{l} \sum_{i=1}^{n}\left(\frac{\lambda(t)}{n}T - \frac{1}{n}\right), \quad T_{all\_N} > 0; \\ otherwise \\ \quad \left|\begin{array}{ll} \sum_{i=1}^{n}\left(\frac{\lambda(t)}{n}T - \frac{1}{n}\right), & \sum_{i=1}^{n}\left(\frac{\lambda(t)}{n}T - \frac{1}{n}\right) > 0; \\ 0, & otherwise. \end{array}\right. \end{array}\right.$$

Интегрируя исходное дифференциальное уравнение при заданных начальных условиях и функции интенсивности запросов, мы получим функцию времени ожидания.

2. Расчет показателя эффективности системы

Введем интегральный показатель эффективности работы системы в заданном режиме *Ф(t).*

Пусть показатель *Ф(t)* есть отношение времени обработки запроса к полному времени пребывания запроса в системе:

$$\varPhi(t) = \left|\begin{array}{l} \frac{T_{pos}}{\left(T_{all} + T_{pos} + T_{sys} + T_{neg}^{proc}\right)} \quad if \quad T_{all} > 0; \\ T_{pos}\lambda(t) \quad otherwise. \end{array}\right.$$

На рисунке 2 приводится пример функции эффективности функционирования системы при периодическом законе изменения $\lambda(t)$, на рисунке обозначенном L.

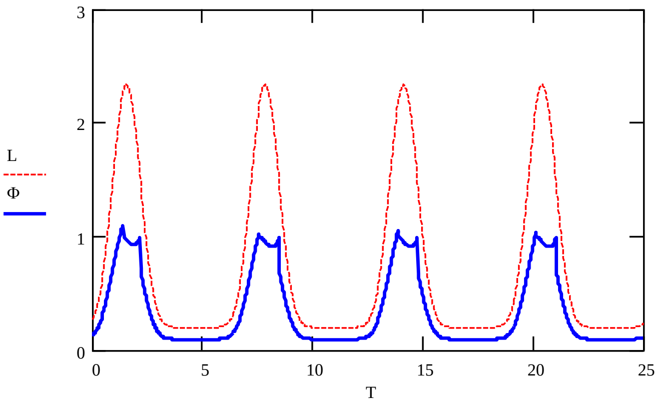


Рис. 2. Зависимость $\Phi(t)$ от $\lambda(t)$

Тогда эффективность функционирования системы на интервале от t_o до t определяется отношением интеграла функции эффективности к временному интервалу, т.е. средней эффективностью системы:

$$I = \frac{\int_{to}^{t} \Phi(t)dt}{t - t_o}.$$

Траектория системы на фазовой плоскости при вынужденных колебаниях и переходе между режимами функционирования с очередью и без очереди запросов изображена на рисунке 3.

Так, для рассматриваемого выше примера, рисунок 3 иллюстрирует функцию эффективности, а рисунок 4 иллюстрирует интеграл этой функции при различных значениях времени обработки запроса.

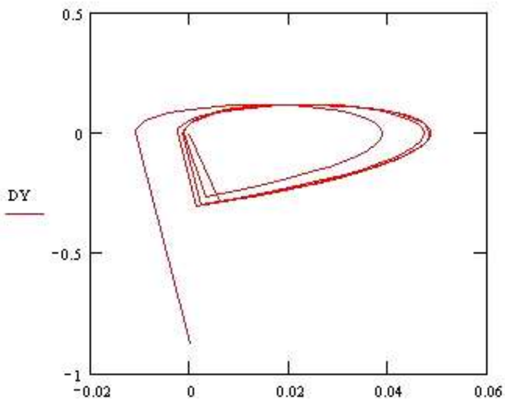


Рис. 3. Траектория системы на фазовой плоскости

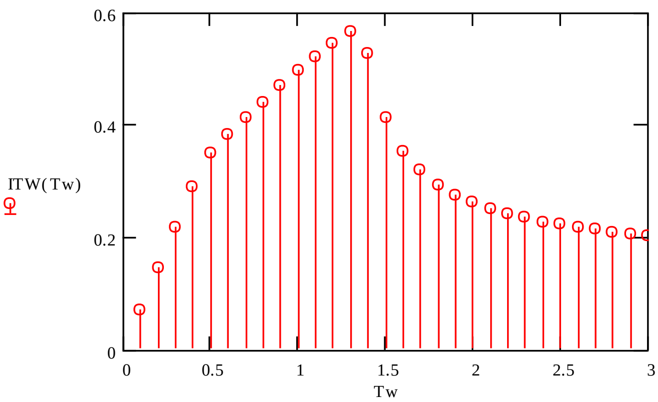


Рис. 4. Интеграл функции эффективности загрузки сервера при различном времени обработки запроса

Очевидно, что для определенного времени обработки рассматриваемый режим загрузки сервера обеспечивает максимум интеграла эффективности.

3. Расчет вероятности отказа агента при случайном времени обработки и заданной интенсивности потока запросов

Возьмем за основу динамическую модель агента с очередью запросов и временем T_{timeout}. Будем считать, что если запрос ожидает в очереди дольше T_{timeout}, то имеет место отказ.

Итак, поведение агента с очередью запросов описывается дифференциальным уравнением первого порядка или для агента с непустой очередью [9; 10]:

$$\frac{dT_{all}}{dt} = \lambda(t)T - 1.$$

Период поступления запросов:

$$T_{ПЗ} = \frac{1}{\lambda(t)}.$$

Пусть время обработки T есть случайная величина, имеющая нормальный закон распределения, с вероятностными характеристиками m_T – математическое ожидание T; D_T – дисперсия T.

Рассмотрим правомерность замены случайной величины T_{all}, описывающей время ожидания в очереди запросов к моменту поступления в эту очередь N запросов.

Предполагается, что поток запросов является простейшей

$$T_{all} = N(T - T_{ПЗ}), \tag{1}$$

случайной величиной $T_{all}'(NT_{ПЗ})$, имеющей интерпретацию времени ожидания в очереди запросов в момент времени $t = NT_{ПЗ}$:

$$T_{all}'(t) = \int_0^t \left(\lambda(s)T - 1\right) ds. \tag{2}$$

Для этого сравним вероятностные характеристики обеих реализаций T_{all}.

Из формулы (1) имеем:

$$m_{Tall} = N\, m_T - NT_{ПЗ}; \quad D_{Tall} = D[NT] = N^2 D_T.$$

Математическое ожидание и дисперсию интеграла случайной функции определим, найдя вероятностные характеристики случайной функции $x(t) = \lambda(t)T - 1$:

$$m_x(t) = \lambda(t)m_T - 1,\ D_x(t) = \lambda(t)^2 D_T,$$

тогда

$$m_{Tall'}(NT_{ПЗ}) = \int_0^{NT_{ПЗ}} x(t)dt,$$

и для $\lambda(t) = const = \lambda$ будет равно

$$m_{Tall'}(NT_{ПЗ}) = (\lambda\, m_T - 1)\, N\, T_{ПЗ} = N\, m_T - NT_{ПЗ};$$

$$D_{Tall'}(NT_{ПЗ}) = K_{Tall'}(NT_{ПЗ}, NT_{ПЗ}) = \int_0^{NT_{ПЗ}} \int_0^{NT_{ПЗ}} x(t)dtdt = \lambda^2 D_T \left[NT_{ПЗ}\right]^2 = N^2 D_T.$$

Следовательно, случайная функция, заданная уравнением (2), аппроксимирует случайную величину T_{all}, определяемую выражением (1).

Так как случайная функция $T_{all}'(t)$ распределена по нормальному закону, то для каждого момента времени t мы можем определить вероятность того, что ожидание в очереди превысит время тайм-аута. Мы можем ее определить, используя интеграл вероятностей [10]. Эта вероятность и является вероятностью отказа агента и является функцией времени.

$$P(T_{all}'(t) > T_{timeout}) = 1 - \Phi^*\left(\frac{T_{timeout} - m_{Tall'}(t)}{\sqrt{D_{Tall'}(t)}}\right) = 1 - \int_{-\infty}^{\frac{T_{timeout} - m_{Tall'}(t)}{\sqrt{D_{Tall'}(t)}}} e^{-\frac{s^2}{2}} ds.$$

Однако следует заметить, что данная оценка справедлива только для случаев такой функции интенсивности запросов $\lambda(t)$, что очередь имеет тенденцию не обращаться в 0.

В противном случае необходимо смещать пределы интегрирования при вычислении математического ожидания и корреляционной функции до точки начала ближайшего накопления очереди, что позволит придать корреляционной функции (дисперсии) значение 0 при нулевом размере очереди.

4. Функционирование коллектива агентов, разделяющих один или несколько процессорных устройств

Рассмотрим модель системы, когда связь между событиями одного процесса определяется марковской моделью n-ного порядка. Тогда функционирование каждого агента можно описать некоторой функцией, характеризующей его с точки зрения требуемых ресурсов. В таком случае под разными процессами будем понимать взаимонезависимые процессы [11].

Рассмотрим функцию $\lambda(t)$ как скрытую функцию, непрерывную и ограниченную на интервале [0, 1]; $\lambda(t)$ – интенсивность потока событий, или порожденная работа процессорного устройства, тогда весь объем работы за время $t_1 - t_0$ будет равно:

$$S = \int_{t_0}^{t_1} \lambda(t)dt.$$

Пусть $P = \{p_i\}$, $i = 1, ..., K$ – множество всех K агентов (процессов), запущенных за определенный промежуток времени $t_1 - t_0$, тогда каждый из них характеризуется интенсивностью потока событий $\lambda_i(t)$.

При работе нескольких агентов одновременно процессорное устройство попеременно переключается между различными процессами. В таком случае функцию времени, отведенного под процесс, можно выразить через несобственный интеграл:

$$\tau(t) = \int_{t_0}^{t} \alpha_i(t)dt,$$

где $\alpha_i(t)$ – ограничивающая функция, непрерывная и ограниченная на интервале [0, 1].

Тогда интенсивность потока событий процесса в многоагентной системе будет определяться по формуле:

$$\eta_i(t) = \alpha_i(t)\,\lambda_i(\tau).$$

Ограничивающая функция отражает доступность ресурсов (процессора) данному агенту в момент времени t и, следовательно, зависит от количества разделяющих эти ресурсы K агентов. В случае бездействия агентов процессор свободен, поэтому функция принимает значение единицы, обозначающей идентичность времен τ и t для данного процесса.

Таким образом, величина τ выступает как коэффициент растяжения(сжатия) функции $\lambda(t)$ вдоль оси времени, а функция $\alpha(t)$ трансформирует ее в зависимости от доступности ресурсов:

$$\alpha_i(t) = \begin{cases} 1, & \lambda_i(t) = 0; \\ 1, & \sum_{i=1}^{K} \lambda_i(t) = 0; \\ \dfrac{\lambda_i(t)}{\sum_{i=1}^{K} \lambda_i(t)} & \textit{otherwise}. \end{cases}$$

В таком случае работа над i-тым процессом, выполненная процессором за время $t_1 - t_0$, будет равна:

$$S_i = \int_{t_0}^{t_1} \alpha_i(t)\lambda_i(\tau)dt = \int_{t_0}^{t_1} \eta_i(t)dt.$$

Общая работа процессора равна суммарной работе по всем p_i процессам:

$$S = \sum_{i=1}^{K} S_i .$$

При таком рассмотрении мы явно выделяем относительно постоянную описательную часть процесса, которая должна быть отработана и при нехватке ресурсов. Это позволяет нам исследовать процесс (агент) в идеальных условиях и считать, что его характеристика $\lambda(t)$ остается постоянной в любой ситуации.

Для повышения точности прогнозирования времени работы процессов использовались многоузловые квадратурные правила (Гаусса – Кронрода).

Рассмотрим вопрос о разделении агентами нескольких процессорных устройств.

Если в системе работают N процессоров, то существует связь между их количеством и ограничивающей функцией $\alpha_{i\_n}(t)$.

Рассмотрим случай, когда превышение требуемых ресурсов над имеющимися в системе распределяется равномерно по всем процессорам, что означает воздействие на все процессы в системе.

В таком случае функция $\alpha_{i\_n}(t)$ ограничена на промежутке $[0, N]$ и имеет следующий вид:

$$\alpha_{i\_n}(t) = \begin{cases} 1, & \lambda_i(t) = 0; \\ 1, & \sum\limits_{i=1}^{K} \lambda_i(t) = 0; \\ \dfrac{k \cdot \lambda_i(t)}{\sum\limits_{i=1}^{K} \lambda_i(t)} & \textit{otherwise}. \end{cases}$$

где k – количество процессоров, имеющихся в системе и необходимых в данный момент времени для обработки запросов агентов без задержки во времени.

Нужно отметить, что величина k не отражает количество действительно работающих в данный момент времени процессоров; k имеет такое значение лишь в случае, когда система работает по следующему алгоритму: агенты работают с одним процессором до полной его загрузки, и задействование нового процессора начинается только тогда, когда суммарная потребность в ресурсах превышает ресурс одного процессора. В основном же агенты распределяются по всем процессорам, поэтому k будет отличаться от их количества и примет значения в интервале $1, ..., N$.

Для оценки результатов расчетов и экспериментов требуется проверить численное равенство величин работ, выполненных процессором в системе без других процессов и без помощи других процессоров и с ними. Это равенство обусловлено тем, что временные ресурсы процессора, необходимые для обработки одного и того же запроса, не зависят от количества запущенных процессов и количества процессоров в системе. Данные, полученные при моделировании таких систем, удовлетворяют последнему требованию, что подтверждает правильность предложенных методов.

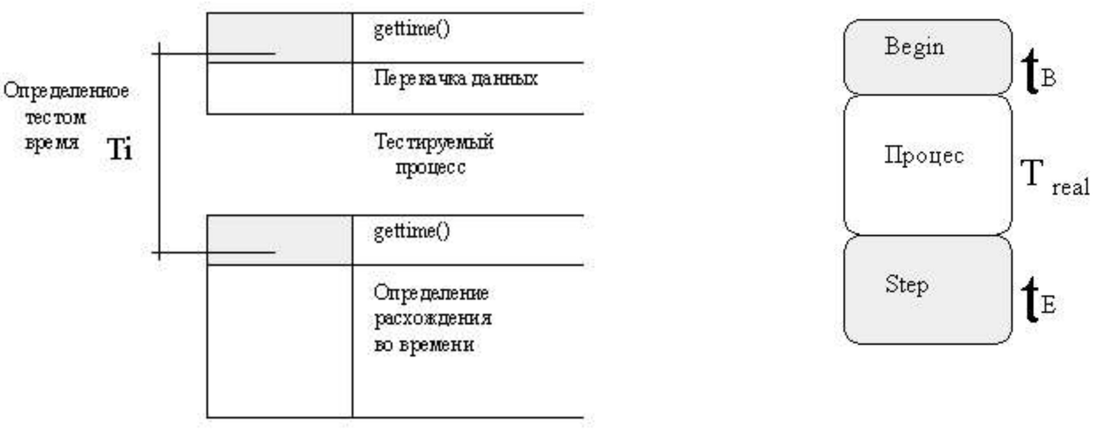
5. Модуль оценки временных характеристик вычислительных процессов

5.1. Класс СТТ. Для оценки вероятностных характеристик вычислительных процессов в объектно-ориентированных средах разработан и реализован программный модуль.

Та особенность, что модуль выполнен в виде объекта, позволяет инкапсулировать в нем функции расчета вероятностных характеристик тестируемого процесса и формы настройки и отчета [12].

Модуль создан как класс объектов-тесторов, описанный на языке С++. Разрабатывался для использования в среде WINDOWS, но пользуясь переносимостью С++ программ на другие платформы, может быть использован и в иных средах.

Схематично цикл тестирования во времени и по процедурам представлен на рисунке 5.



а) б)

Рис. 5. Цикл тестирования во времени (а); цикл тестирования по процедурам (б)

Для определения основных параметров цикла тестирования необходимо провести два эксперимента.

5.2. Эксперименты

Эксперимент 1. Определение t_B: из рисунка 6, с учетом рисунка 5, видно, что

$$\overline{T_i} = t_B \cdot (m+1).$$

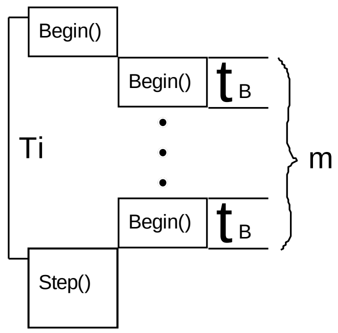


Рис. 6. Схема эксперимента 1

Учитывая тот факт, что компьютерное время дискретно времени (d = 0,055 с), при планировании эксперимента необходимо выполнить условие $T_i >> d$.

Тогда

$$t_{B\_real} = \frac{\overline{T_i}}{(m+1)}. \tag{3}$$

Эксперимент 2 (рис. 7). Для определения нефиксируемой задержки фонового процесса при тестировании его отдельного участка.

Условия протекания эксперимента:

- минимум возмущающих воздействий;
- количество однотипных экспериментов $N_i >> 1$;
- время вложенного процесса $T_{1i} >> d$;

накопленная за вложенность ошибка $\left|\overline{T_{ni}} - \overline{T_{li}}\right| > \overline{T_{li}}$.

Тогда, учитывая (3):

$$T_{n_{real}} = \frac{\sum_{i=1}^{N_i} T_{ni} - N_i \cdot t_{B_{real}}}{N_i}; \quad T_{l_{real}} = \frac{\sum_{i=1}^{N_i} T_{li} - N_i \cdot t_{B_{real}}}{N_i}. \tag{4}$$

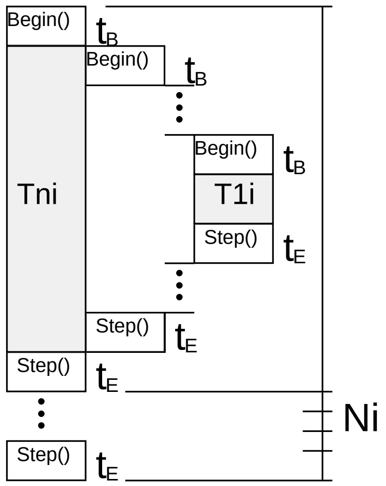


Рис. 7. Схема эксперимента 2

Учитывая, что

$$\overline{T_{ni}} = \frac{\sum_{i=1}^{Ni} T_{ni}}{N_i}; \quad \overline{T_{li}} = \frac{\sum_{i=1}^{N_i} T_{li}}{N_i}, \tag{5}$$

Из рисунков 7 и 5 по формулам (4), (5) находим t_E:

$$t_{Ereal} = (T_{n real} - T_{1real} - t_{Breal}\ (n - 1\ ))\ /\ (n\ - 1),$$

или

$$t_{Ereal} = (T_{ni} - T_{1i} - t_{Breal}\,(n - 1))\ /\ (n - 1).$$

5.3. Отчет по экспериментам. Необходимо отметить, что исходные данные для экспериментов выбирались исходя из ограничений, накладываемых на эксперимент характеристиками компьютера (объем свободной памяти, быстродействие), и имеющегося в нашем распоряжении ПО (модель памяти в WINDOWS, длительность выполнения системных функций, определение системного времени). Полученные данные описывают характеристики инструмента, функционирующего на PC 386DX/40. Доверительный интервал определен по правилу трех среднеквадратических отклонений.

Эксперимент 1:

m = 100000; N_i = 100

$\overline{T_i}$ = 0,702 ± 3 $\sqrt{D}$ = 0,702 - доверительный интервал (0,6257; 0,7783) с

D = 0,000647 с - дисперсия

t_{Breal} = 7,019 - (6,259; 7,779) E – 6 с

Эксперимент 2:

N_i = 400 n = 301 - степень вложенности

$\overline{T_1}$ = 0,0548 ± 3 $\sqrt{D_1}$ = 0,0548 - (0,0288; 0,0808) с

D_1 = 0,00008 с

$\overline{T_{301}}$ = 0,223 ± 3 $\sqrt{D_{301}}$ = 0,223 - (0,177; 0.269) с

D_{301} = 0,00024 с

t_{Ereal} = 0,00056 ± 3 $\sqrt{D_{E\_real}}$ = 0,00056 - (0; 0,0015) с

5.4. Предоставляемый сервис. В результате тестирования объект рассчитывает следующие характеристики вычислительного процесса:

- время тестирования;

- математическое ожидание времени протекания процесса. Этот параметр завышен на величину t_{Breal}, которую необходимо определить применительно к имеющейся технике, поставив эксперимент по схеме 1 (см. рис. 6);

- дисперсию случайной величины – времени протекания процесса;

- проверка истинности гипотезы о нормальном распределении случайной величины.

Зная основные статистические характеристики, можно определить вероятность попадания случайной величины X в интервал или вероятность не превышения отклонения случайной величины от некоторого значения величины δ (6):

$$P(\alpha < X < \beta) = \Phi\left(\frac{\beta - M(X)}{\sqrt{D(X)}}\right) - \Phi\left(\frac{\alpha - M(X)}{\sqrt{D(X)}}\right); \qquad P(|X-\alpha| < \delta) = 2\Phi\left(\frac{\delta}{\sqrt{D(X)}}\right); \tag{6}$$

где $\Phi(X) = \frac{1}{\sqrt{2\pi}}\int_0^X e^{-\frac{t^2}{2}}dt$ – функция Лапласа; $M(X)$, $D(X)$ – математическое ожидание и дисперсия случайной величины X.

5.5. Рекомендации по использованию компоненты. Производить оценку малых по длительности процессов (меньших по длительности дискреты системного времени d) пакетно с дальнейшим пересчетом характеристик по единичному процессу. Планировать тест таким образом, чтобы отсутствовала или имелась возможность вычислить влияние одного процесса тестирования на другой. Определять характеристики сложных систем, оценивая характеристики отдельных ее элементов. Поэтому рекомендуется объектно-ориентированный синтез сложной системы, хотя вполне реально применение модуля для теста процедурного кода.

ЛИТЕРАТУРА

1. Cummings, B. Object Oriented Analysis and Design with Applications Grady Booch / B. Cummings. – 1994.
2. Meyer, B. Object Success / B. Meyer. – London: Prentice Hall, 1995.
3. OOA, Coad, et. al., Yourdon Press, 1990.
4. Durfee, E.H. Coherent Cooperation Among Communicating Problem Solvers / E.H. Durfee, V.R. Lesser, D.D. Corkill // IEEE Transactions on Computers. – 1987. – P. 1275 – 1291.
5. Durfee, EH. Cooperative Distributed Problem-solving / E H. Durfee, V.R. Lesser, D.D. Corkill // In The Handbook of Artificial Intelligence. – Vol. IV. – A. Barr, P.R. Cohen, E.A. Feigenbaum (ed.), 1989. – P. 83 – 148.
6. Chapman, D. Planning for Conjunctive Goals / D. Chapman // Artificial Intelligence Journal, 1987. – № 32. – P. 333 – 367.
7. Barraquand, J. Robot Motion Planning: A Distributed Representation Approach / J. Barraquand, J.C. Latombe // Research Report. – STAN-CS-89-1257. – Stanford: Department of Computer Science, Stanford University, 1989.
8. Биркгоф, Дж.Д. Динамические системы / Дж.Д. Биркгоф. – Ижевск: Издат. Дом «Удмуртский университет», 1999. – 408 с.
9. Венцель, Е.С. Теория вероятностей: учеб. для вузов / Е.С. Венцель. – 7-е изд. стер. – М.: Высш. шк., 2001. – 575 с.
10. Гмурман, В.Е. Теория вероятностей и математическая статистика: учеб. пособие для вузов / В.Е. Гмурман. – 5-е изд., перераб. и доп. – М.: Высш. шк., 1977. – 479 с.
11. Севастьянов, Б.А. Эргодическая теорема для марковских процессов и ее приложение к телефонным системам с отказами / Б.А. Севастьянов // Теория вероятностей и ее применение. – Т. 2. – Вып. 1. – 1957.
12. Телло, Э. Объектно-ориентированное программирование для WINDOWS / Э. Телло. – М.: Высш. шк., 1993. – 347 с.

Поступила 15.08.2007