

ПРОЦЕДУРА МОДЕЛИ ПРОГНОЗИРОВАНИЯ ЦЕПЕЙ В СЕТЕВОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

В данной работе было выполнено процедуру модели прогнозирования цепей в сетевой вычислительной среде. Было исследовано работу процессора компьютера и смоделировано его ход действий при его работе. Предсказано доступность, надежность работы процессора компьютера.

Ключевые слова: процессор, компьютер, вычисления, прогнозирование.

A. Mordik, V. Novitsky. Procedure of the circuit-forecasting model in the network computer environment. In this work, the procedure for the chain prediction model in a networked computing environment was performed. The work of the computer processor was investigated and its behavior was modeled during its operation. The availability and reliability of the computer processor is predicted.

Keywords: processor, computer, computing, forecasting.

О.О. Мордик, В.Ю. Новицкий. Процедура моделі прогнозування кіл в мережі обчислювального середовища. У даній роботі було виконано процедуру моделі прогнозування ланцюгів в мережевий обчислювальної середовищі. Було досліджено роботу процесора комп'ютера і змодельоване його хід дій при його роботі. Передбачено доступність, надійність роботи процесора комп'ютера.

Ключові слова: процесор, комп'ютер, обчислення, прогнозування.

В настоящий момент активно развиваются компьютерные сети, и их пропускная способность увеличивается. Он разрешает отдельным независимым исследователям использовать распределенные вычислительные системы в локальных и корпоративных компьютерных сетях. Эти сети имеют следующие возможности:

- они состоят из относительно небольшого числа компьютеров (до нескольких сотен);
- сегменты сети могут либо быть частью локальной сети, либо удалаться друг от друга географически и подключается через Интернет;
- компьютеры, входящие в такую сеть, могут существенно различаться в зависимости от спецификаций, типа сетевого подключения и режимов использования;
- компьютеры неотъемлемы, то есть они также могут выполнять задачи пользователя с более высоким приоритетом.

Принципы организации распределенных вычислений для таких сетей до настоящего времени недостаточно развиты.

Суть идеи распределенных вычислений в описанных сетях состоит в том, чтобы разделить исходную вычислительную задачу на фрагменты, которые называются подзадачами. Каждая такая подзадача состоит из набора данных и алгоритма обработки, который реализуется как программный модуль [1].

Все компьютеры, доступные для вычислений, поддерживают активное сетевое подключение к компьютеру с программным обеспечением координатора и известны как вычислительные элементы (СЕ). В соответствии с этой процедурой координатор выделяет компьютер для решения интенсивных вычислений, отправляя список вычислительных элементов в диспетчер выполнения, который представляет собой программу, которая выполняет общее управление прогрессом вычислений. Менеджер выполнения распределяет подзадачи на доступные компьютеры, а затем собирает и сохраняет результаты вычислений.

Одним из наиболее важных аспектов организации таких вычислений является предсказание нагрузки на ПК с пользовательскими задачами за определенный промежуток времени [6, 7]. Его предсказание позволяет оценить ключевые характеристики, описывающие процесс вычисления:

- оценка времени решения подзадачи на основе конкретного вычислительного элемента, начиная с заданного момента времени;
- оценка общего времени расчета задачи.

В дополнение к прогнозированию нагрузки процессора СЕ, здесь также являются задачами, которые схожи по своему значению: прогнозирование доступности и надежности вычислительных элементов в заданный временной интервал. На основе сгенерированных прогнозов становится возможной реализация политик назначения эффективных фрагментов (подзадач) для вычислительной задачи для отдельных компьютеров, которая учитывает характер использования компьютеров.

Прежде чем перейти к описанию алгоритма генерации предсказательного моделирования, рассмотрим способ представления состояния вычислительного элемента во времени.

В каждый t момент времени вычислительный элемент характеризуется вектором состояния V_t . Вектор состояния V состоит из трех компонентов:

- загрузка СЕ с пользовательскими задачами описывается реальным номером L (загрузка), $L \in [0,1]$. Таким образом, $L = 0$ означает, что компьютер неактивен, а $L = 1$ означает, что все процессорное время компьютера зарезервировано для выполнения пользовательских задач;

- доступность СЕ (доступность сетевого подключения к координатору) описывается повторно (доступность), $A \in [0,1]$. Значения значений параметров описаны ниже;

- загрузка подзадачи СЕ описывается реальным номером S (подзадачной нагрузки), $S \in [0,1]$. Если $S = 0$, СЕ не выполняет вычисления подзадач; если $S = 1$, все время процессора зарезервировано для реализации разрешимых подзадач.

Таким образом, $V_t = \{L_t, A_t, S_t\}$. Очевидно, что $(S + L) \leq 1$.

Фиксация вектора состояния осуществляется с дискретными временными интервалами (Δt). Состояние, когда СЕ не функционирует (т. е. физически отключен), описывается вектором $V = \{0,0,0\}$ и определяется как интервал, при котором нет сбор данных был выполнен. В дальнейшем это состояние называется сбоем СЕ, поскольку оно приводит к потере прогресса в решении текущей подзадачи.

СЕ в течение временного интервала $[t_1, t_2]$ описывается набором пар (V_t, t) , так что $t \in [t_1, t_2]$; число таких пар определяется интервалом Δt . Интервал Δt должен быть достаточно мал, чтобы уменьшить влияние ударов на короткие «скачки» загрузки процессора, связанные с работой операционной системы и других приложений. В будущем фильтры будут устранять эти прыжки.

Сбор данных о недоступности СЕ осуществляется путем измерения суммы потерь ($A = 0$ соответствует 100% потерь; $A = 1$, к общему отсутствию потерь) при отправке тестовых пакетов СЕ координатору и обратно (так называемое Heartbeat).

Теперь мы можем предоставить официальное утверждение о задачах генерации прогноза. Задача прогнозирования нагрузки СЕ указывается следующим образом:

Необходимо предсказывать нагрузку СЕ с пользовательскими задачами в заданный промежуток времени $[t_1, t_2]$ приращениях Δt , т. е. получить набор пар (L, t) . Это предсказание послужит основой для расчета оцененного времени решения подзадачи на заданном интервале в будущем.

Задача прогноза доступности СЕ указывается следующим образом:

Необходимо оценить вероятность доступности СЕ ($A1$ и $A2$) на границах временного интервала $[t_1, t_2]$. Необходимым условием является наличие СЕ в начале и в конце вычислений для целей передачи исходных данных и результатов вычислений.

Задача прогноза надежности СЕ определяется следующим образом:

Необходимо оценить вероятность отказа СЕ (E) на временном интервале $[t_1, t_2]$. В то же время верхний предел (t_2) определяется в ходе оценки времени решения, то есть прогнозирующее моделирование для доступности СЕ и нагрузки фактически реализуется одновременно с оценкой времени решения подзадачи. Это связано с тем, что сама загрузка L влияет на время решения подзадачи.

В этой статье мы рассмотрим алгоритм, позволяющий генерировать долгосрочные прогнозы загрузки ПК задачами своего владельца. Решение задач прогнозирующего моделирования для доступности СЕ и оценки вероятности сбоя СЕ остается за рамками этой работы, поскольку они основаны на хорошо известных подходах [3].

Термин «долгосрочное прогнозирование» относится к оценке значений загрузки ПК в отдельных точках времени от нескольких часов до нескольких дней. Разработка алгоритма была обусловлена тем, что обычные типы предсказательного моделирования, в том числе основанные на рядах Фурье [2], дают большую ошибку и требуют интенсивного вычисления при формировании долгосрочных прогнозов.

Алгоритм основан на выраженной суточной частоте изменений нагрузки СЕ, наблюдаемой автором в исследовании. В этом случае можно выделить несколько стандартных циклов, которые в дальнейшем называются шаблонами нагрузки, и непрерывный временной ряд значений нагрузки ЛП

для ПК можно рассматривать как последовательность этих узоров. Набор шаблонов будет зависеть от конкретного места назначения компьютера. Другим основным фактом, лежащим в основе алгоритма, является наличие статистических закономерностей в цепочке шаблонов.

Велоспорт может быть вызван различными причинами; однако в большинстве изученных случаев он привязан к 24-часовым периодам. В случаях, когда не наблюдается выраженного циклирования, необходимо использовать другие методы прогнозирования. Позвольте нам объяснить это ниже, используя следующий пример. Для компьютеров, которые используются в доме, существует несколько шаблонов нагрузки, которые номинально соответствуют дням недели: будний день, пятница, суббота и воскресенье. Эти рисунки показаны на рисунке 1. Изображение А соответствует будним дням; изображение В, до конца рабочей недели; и С, до выходных. Как видно, почти без нагрузки в течение дня и относительно низких нагрузок характерны будние дни вечером.

Пятница характеризуется высоким уровнем нагрузки с раннего вечера и до субботы. Выходные дни характеризуются средними уровнями нагрузки в течение дня и высокими уровнями нагрузки в вечернее время.

Наличие статистических закономерностей в цепочке шаблонов обусловлено изменением дней недели. Алгоритм прогнозирования состоит из следующих шагов:

- получение статистических данных из файла;
- очистка данных с помощью метода скользящего среднего;
- поиск лучшего деления на суточные периоды;
- разделение данных на суточные периодограммы;
- вычисление корреляции периода с помощью метода стандартного отклонения;
- кластеризация;
- получение ежедневных паттернов путем усреднения значений нагрузки суточных периодов, которые включены в кластер соответствующего шаблона;
- сопоставление шаблонов с символическими именами;
- кодирование исходного набора данных в последовательность символов;
- создание таблиц для построения цепей Маркова;
- подготовка прогнозирующего моделирования последовательности символов;
- декодирование сгенерированной последовательности с помощью шаблонов;
- вычисление запрошенных оценок.

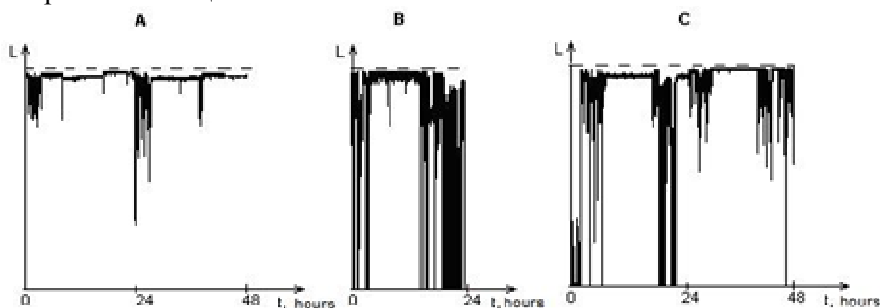


Рис. 1 Загрузка шаблонов типичного домашнего компьютера

Рассмотрим детали алгоритма и их реализацию более подробно. Приобретение статистических данных из средств хранения принимают ранее сохраненную последовательность значений V_t . Этот алгоритм использует только компонент вектора L , который будет обозначен L_t . Для анализа может быть выбрана только часть данных. Отметим количество полных дней в выбранном диапазоне как D .

Значения нагрузки фильтруются с использованием фильтра скользящей средней [5]; окно усреднения устанавливается равным 15 минут.

На основе полученной цепочки генерируется таблица вероятностей перехода из заданного состояния системы в другое состояние [4]. Состояние системы определяется как последовательность символов, которые соответствуют состоянию, в котором система находится в данный момент, а также нескольким предыдущим состояниям. В настоящее время длина последовательности SL составляет 4 элемента, т. е. используется таблица вероятностей переходов 4-го порядка. Вероятности вычисляются путем подсчета числа переходов из текущего состояния во всех возможных новых состояниях с последующим переходом от статистических к математическим вероятностям (нормализации).

Предиктивное моделирование выполняется следующим образом:

- часть периодограммы, содержащая полные дни SL и интервал от конца последнего ежедневного периода и до момента выбора настоящего момента;

- последовательность полных ежедневных периодограмм кодируется путем поиска наиболее сходного шаблона для каждого периода;
- прогнозируется текущий дневной период на основе цепей Маркова и вычисляется значение функции расстояния между текущим периодом и прогнозируемым рисунком;
- поиск ближайшего шаблона для текущего периода и вычисление значения функции расстояния;
- один из двух символов (один с наименьшим отклонением), которые были получены с использованием вышеуказанной процедуры, выбирается и помещается в цепочку;
- цепочка дополняется заранее определенным количеством периодов;
- сгенерированная последовательность символов декодируется на основе шаблонов;
- рассчитывается требуемая оценка.

Рассмотрение вероятности перехода в каждом из состояний в течение следующего периода с использованием таблиц для построения цепей Маркова и, следовательно, также возможность определения нескольких вариантов профилей нагрузки и их соответствующих вероятностей. То есть создаются все возможные варианты последовательностей шаблонов в течение заданного периода с вычислением соответствующих вероятностей.

Однако из-за резкого увеличения количества комбинаций при увеличении числа периодов прогнозирования этот метод применим только для относительно коротких интервалов (а именно, 2 -4 периодов) в зависимости от количества шаблонов и числовых значений вероятностей переходов в разные состояния, которые были идентифицированы путем кластеризации. В итоге, продолжительность заданного интервала достаточна для решения большинства задач.

Перечень ссылок:

1. Andrews, G. Foundations of Multithreaded, Parallel, and Distributed Programming. / Andrews, Gregory. - М: Izdat .dom «Vil'jams», 2003. - 512 p.
2. Брилингджер, Д. Временные ряды. Обработка данных и теория/ Дэвид Р. Брилингджер. - Изд.: МИР, 1980. - 536 с.
3. Кругликов В. К. Вероятностный компьютерный эксперимент в приборостроении/ В.К. Кругликов. - Изд.: Машиностроение, Ленингр. отделение, 1985. - 247 с.
4. Саульев В. К. Математические модули в базовых теориях / В. К. Саульев – Изд.: Статистика, 1979. – 96 с.
5. Tjurin, J.N. Analysis of the data on the computer / J.N. Tjurin, A.A. Makarov. - 3-rd izd., pererab. i dop. - М.:INFRAM, 2003. - 544 p.
6. Smith, W. Using run-time predictions to estimate queue wait times and improve scheduler performance / W. Smith, V. Taylor, I. Foster //Proceedings of the IPPS/SPDP '99 Workshop on Job Scheduling Strategies for Parallel Processing. - Springer Verlag, 1999.