ДИСПЕТЧЕРИЗАЦИЯ ПРОГРАММ В РАСПРЕДЕЛЕННЫХ СРЕДАХ

К.С. Ткаченко

Севастопольский национальный технический университет

г. Севастополь, ул. Университетская, 33
E-mail: kvt@sevgtu.sebastopol.ua

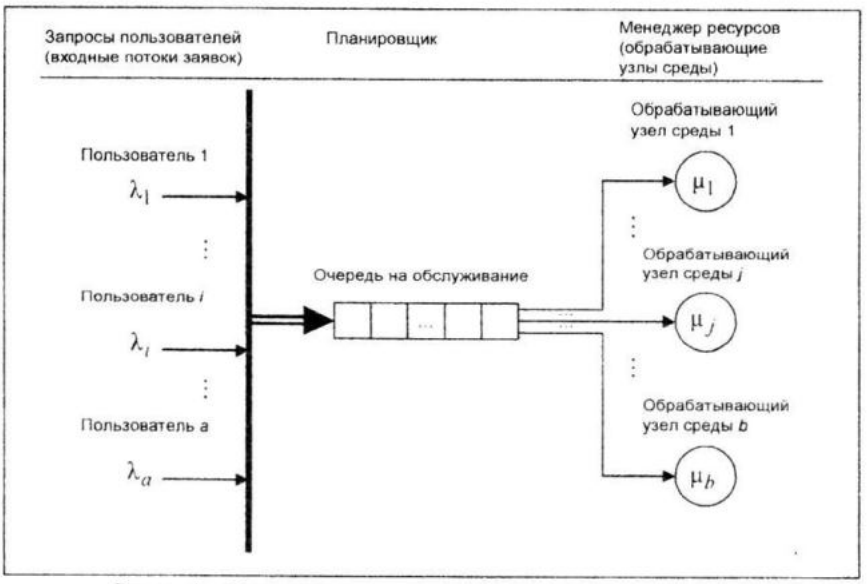
Приводится постановка задачи диспетчеризации программ в распределенной среде. Описываются принципы решения задачи для обеспечения безусловной минимизации наибольшего предельного значения средних текущих потерь.

Введение. Задача диспетчеризации программ в распределенных средах (РС) [1] имеет большое практическое значение [2], и при решении ее необходимо построить и исследовать соответствующее математическое и программное обеспечение вычислительных РС, при этом возможно использовать адаптацию технических систем – неотъемлемую составляющую жизнедеятельности различных программных и аппаратных систем, направленных на решение конкретных задач [3]. Изначально центры обработки данных состояли из малого числа мэйнфреймов, на каждом из которых исполнялось несколько прикладных задач со множеством пользователей [4]. Специалисты по планированию мощности отвечали за то, чтобы достаточная совокупная мощность была доступна именно в то время, когда она была нужна. С появлением географически распределенных вычислений новые нагрузки приложений стали назначаться каждая на отдельный небольшой сервер. Наращивание мощности для небольших серверов обходилось гораздо дешевле, чем на мэйнфреймах, что позволяло прогнозировать нагрузку и вводить новый сервер достаточной мощности с упреждением, чтобы окупить возможность дорастания нагрузки до предела. Взрывной рост корпоративных вычислений и вычислений через Интернет привел к разрастанию вычислительных центров. Производственные ЦОД, как правило, состоят из множества слабо используемых серверов, что влечет за собой высокую стоимость владения (включая затраты на аренду и электроэнергию для вычислений и охлаждения), лицензий на ПО и администрирования с участием операторов. Поэтому и возникает задача диспетчеризации программ в распределенных средах.

Имитационная модель и инструментальное средство. Выполняется разработка и исследование модели и программного обеспечения системы поддержки принятия решений (СППР) в РС, включая анализ, разработку и верификацию методов, алгоритмов, программ поддержки диспетчеризации. Пусть управление процессом приема, обработки и выдачи информации при обслуживании пакетов программ осуществляется управляющим вычислительным узлом РС. На данном вычислительном узле имеется программное обеспечение для диспетчеризации выполнения пакетов поступающих в РС заданий между коммутационной системой и целевыми вычислительными узлами, используемое для выполнения балансировки нагрузки и увеличения реальной производительности. Рассматривается использование двух различных компонентов – менеджера ресурсов и планировщика, при этом планировщик определяет очередность выполнения пакетов заданий и использование ими тех или иных ресурсов [2]. Использование данных компонентов в процессе обслуживания программ требует определенных затрат ресурсов РС, приводящих к снижению пропускной способности РС. Если часть процессоров перегружена, часть простаивает, то вычислительный процесс организован плохо, что при малых нагрузках не имеет значения, а при большой нагрузке низкая величина пропускной способности РС приведет к отказам в обслуживании программ. Вследствие обусловленных выше проблем необходимо изменять совокупность используемых отдельных процессоров при изменении вычислительной нагрузки на РС.

Предлагается имитационная модель распределенной среды, использующей алгоритмы адаптивного выбора вариантов для распределения заявок, приведенная на рис. 1. Модель обеспечивает организацию диспетчеризации программ. В мо-

дели можно выделить следующие элементы: запросы пользователей – это входные потоки заявок, интенсивности которых λ_i, в общем случае, неизвестны; планировщика – это компонента адаптивного выбора вариантов, обеспечивает организацию обслуживания работ, назначение их на ресурсы; менеджер ресурсов – это обрабатывающие узлы среды, между которыми распределены ресурсы μ_j (на основе приоритетов, заданных планировщиком). В модели выделены потоки входящих заявок, поток заявок на обслуживание, потоки распределенных по ресурсам заявок. Основными параметрами модели являются: число пользователей a, число обрабатывающих узлов среды b, производительности обрабатывающих узлов среды μ_j. Известно, что обрабатывающие узлы сети географически распределены.



Р и с. 1. Имитационная модель распределенной среды

Число максимально доступных процессоров в РС равно $K = b$, сами эти процессоры есть $q_1, \ldots, q_K$. Время, необходимое для выполнения пакета на том или ином процессоре, в общем случае зависит от состояния распределенной среды и может рассматриваться как случайная величина. На вход распределенной среды поступают запросы на обработку программ (при этом тип запроса и его трудоёмкость заранее неизвестны), для них образуется очередь, в которой запросы ожидают начала обслуживания. Выборе на обслуживание того или иного пакета принимается в порядке *FIFO*. При этом планировщик указывает, какие процессоры должны быть подключены. Процесс планирования можно организовать следующим образом. Время разбивается на последовательные интервалы некоторой длительности T с номерами 1, 2, … в порядке наступления. Каждому интервалу с номером n ставится в соответствие вектор $x_n = \left(x_n^1, \ldots, x_n^K\right)$, каждая компонента которого принимает два значения: 0 или 1, таким образом, что если $x_n^k = 1$, то в течение n-го интервала времени процессор q_k включается в процесс обслуживания пакета запросов, и не включается, если $x_n^k = 0$ $(k = 1, \ldots, K)$, причем таким образом, что номер выбранного адаптивного варианта в двоичном коде соответствует данному вектору.

Вводится функция потерь системы на интервале времени (t_n, t_{n+1}) как $\xi_n = C_1\left(Q_n^{BX} - Q_n^{BЫX}\right) + C_2 N_{ПР}$, где Q_n^{BX} – число вошедших пакетов заданий, ед., $Q_n^{BЫX}$ – обслуженных, ед., C_1

– штраф за отказ в обслуживании заданий, у.е./ед, C_2 – штраф за простой процессоров, у.е./ед., $N_{ПД}$ – количество простаивающих процессоров (загрузка которых менее $1/K$). Необходимо обеспечить безусловную минимизацию наибольшего предельного значения средних текущих потерь $\Phi_n = \frac{1}{n}\sum_{t=1}^{n}\xi_t$.

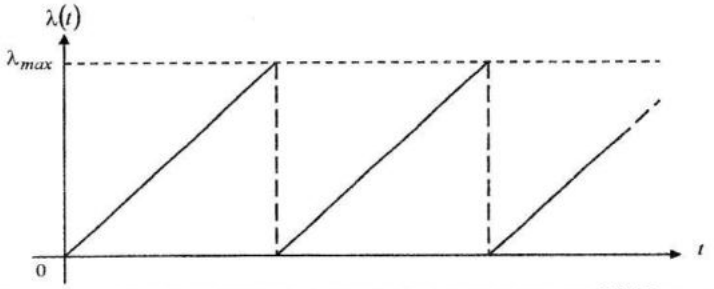
Для выполнения безусловной параметрической оптимизации модели распределенной среды разработано инструментальное средство информационного обеспечения принятия решений на языке программирования *Java*, и является консольным приложением с текстовым интерфейсом пользователя, что немаловажно вследствие низкого потребления системных ресурсов. В основном классе программы имеются методы для организации вычислений, поддержки ввода-вывода, проектирования на ε-симплекс, выполнения одного шага оптимизации, ввод сведений о потерях на текущем шаге, и иных, требуемых при выполнении оптимизации. Наличие априорной неопределенности приводит к необходимости использовать в программном средстве рандомизированные стратегии, чему соответствуют стохастические автоматы с переменной структурой. Одним из проекционных алгоритмов стохастической аппроксимации является следующий проекционный алгоритм стохастической аппроксимации с небинарными потерями [3]

$$p_{n+1} = \pi^{N}_{\varepsilon_{n+1}}\left\{p_n - \gamma_n \frac{\xi_n - \Delta}{e^T(x_n)p_n} e(x_n)\right\},$$

где γ_n – длина шага, Δ – числовой параметр, $\pi^{N}_{\varepsilon_n}$ – оператор проектирования на ε-симплекс, $\{\varepsilon_n\}$ – последовательность чисел $\varepsilon_n \in [0, N^{-1})$, отделяющая компоненты вектора p_n от нуля (для обеспечения на каждом шаге ненулевой вероятности выбора любого варианта $x_i \in X$), n – номер шага. При этом $\gamma_n = \frac{\gamma}{(n+a)^{\chi}}$, $\varepsilon_n = \frac{\varepsilon}{(n+a)^{\nu}}$, причем γ и ε – начальные значения, a, $b > 0$, $0 < \nu < 2\chi - 1, \chi \le 1$. В программном средстве предусмотрен ввод параметров алгоритма Δ, γ, ε, χ, ν, a, b, а также сведений о потерях на текущем шаге и числа шагов. Интерфейс программы СППР в работе приведен на рис. 2. Изменение входного потока заявок во времени изображено на рис. 3.

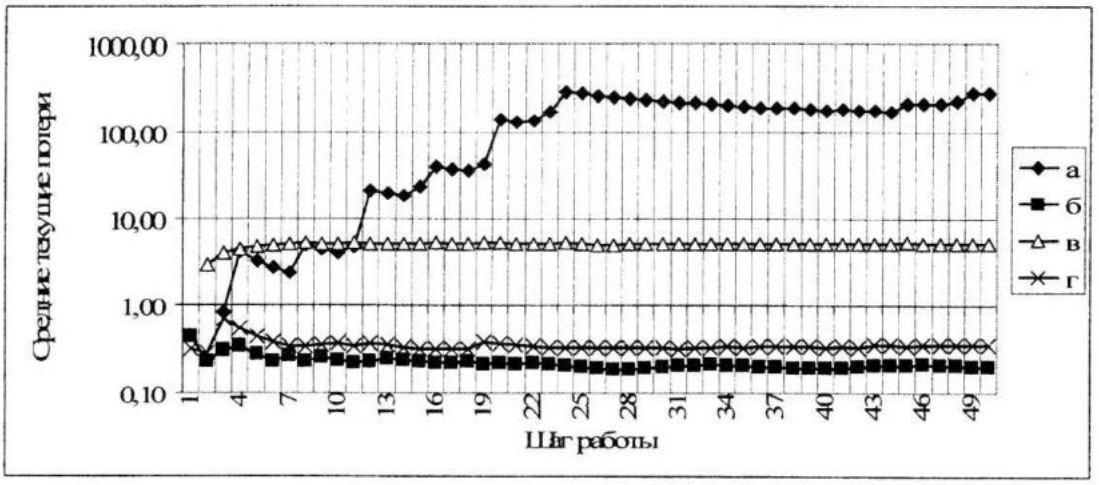
```
run:
Введите число шагов        : 50
Введите delta              : 0.5
Введите gamma начальное    : 0.5
Введите epsilon начальное  : 0.1
Введите hi                 : 0.5
Введите nu                 : 0.4
Введите A                  : 0.1
Введите B                  : 0.1
```

Р и с. 2. Интерфейс программы СППР в работе



Р и с. 3. Зависимость интенсивности входного потока заданий $\lambda(t)$ от модельного времени t (фрагмент), где λ_{max} – максимальное значение $\lambda(t)$

По результатам работы получена функциональная зависимость средних текущих потерь от шага работы, приведенная на рис. 4: 4, а – случай, когда $C_1 > C_2$ (дорогие заявки), без адаптации; 4, б – случай, когда $C_1 > C_2$ («дорогие» заявки), с адаптацией; 4, в – случай, когда $C_2 > C_1$ («дорогие» вычислительный узлы), без адаптации; 4, г – случай, когда $C_2 > C_1$ («дорогие» вычислительный узлы), с адаптацией.



Р и с. 4. Функциональная зависимость средних текущих потерь (логарифмическая шкала) от шага работы (варианты а, б, в, г)

Заключение. Результатами данной работы явились разработанная модель и инструментальное средство СППР. Перспективой дальнейших исследований станет детализация модели и, соответственно, СППР. Необходимо, чтобы предложенный комплекс программного обеспечения позволял улучшить сходимость при калибровке имитационной модели, использовал понятийное описание произвольной распределенной среды с позиций системного подхода, а также получить обобщенные выражения для функции штрафа, исключив противоречия, связанные с неопределенностью входных данных.

Целесообразно рассмотреть вопрос о соответствии результатов модельных прогонов реально функционирующим в настоящее время распределенным средам, в т.ч. используемым в ЦОД. В дальнейшем планируется использовать результаты работы для управления мощностью эффективным образом по соотношению «стоимость к эффективности».

СПИСОК ЛИТЕРАТУРЫ

1. *Распределенная* среда обработки данных [Электронный ресурс]. – Электрон. текстовые, граф. дан. (269457 bytes). – Режим доступа: http://dic.academic.ru/dic.nsf/fin_enc/28263 Monday, July 11 2011 11 12:46.
2. *Топорков В.В.* Модели распределенных вычислений / В.В. Топорков. – М.: ФИЗМАТЛИТ, 2004. – 320 с.
3. *Назин А.В.* Адаптивный выбор вариантов. Рекуррентные алгоритмы / А.В. Назин, А.С. Позняк. – М.: Наука, 1986. – 288 с.
4. *Управление* вычислительной мощностью и прогнозирование потребностей центров обработки данных [Электронный ресурс]. – Электрон. текстовые, граф. дан. (511782 bytes) – http://www.bytemag.ru/articles/detail.php?ID=11622 Saturday, July 02 2011 14:34.