

УДК 004.7

К.С. Ткаченко*Севастопольский национальный технический университет**ул. Университетская, 33, г. Севастополь, Украина, 99053**E-mail: kvf@sevgtu.sebastopol.ua***АДАПТИВНОЕ УПРАВЛЕНИЕ РАСПРЕДЕЛЕННОЙ СРЕДОЙ НА БАЗЕ ИМИТАЦИОННОЙ МОДЕЛИ GRID-СИСТЕМЫ**

Рассматривается адаптивное управление распределенной средой на базе имитационной модели GRID-системы для обеспечения безусловной минимизации наибольшего предельного значения средних текущих потерь. Приведены необходимые рисунки.

Ключевые слова: *распределенная среда, имитационная модель, адаптивное управление.*

Адаптивное управление техническими системами является неотъемлемой составляющей жизнедеятельности многих аппаратных и программных систем, направленных на решение конкретных задач [1–3], в том числе задачи управление распределенной средой (РС) на основе GRID-системы, имеющей большое практическое значение [4, 5].

Изначально центры обработки данных состояли из малого числа мэйнфреймов, на каждом из которых исполнялось несколько прикладных задач со множеством пользователей [6]. Специалисты по планированию мощности отвечали за доступность достаточной совокупной мощности в нужное время. С появлением распределенных вычислений появилась возможность назначать каждую новую нагрузку приложений на отдельный небольшой сервер. Нарастивание мощности для небольших серверов обходилось гораздо дешевле, чем на мэйнфреймах. Потребности в ресурсах для каждого сервиса соответствуют его прикладной нагрузке. Используются пулы серверных ресурсов совместно с механизмами виртуализации, что позволяет исполнять на каждом сервере множество приложений, что увеличивает гибкость, возможность быстро передиспетчеризировать серверную мощность для лучшего соответствия потребностям владельцев приложений, и обеспечения снижения общей стоимости владения. Сложность этих сред создает дополнительные проблемы при администрировании. Имеется много нагрузок, на каждый сервер можно назначить некое конечное их число, и каждая нагрузка имеет требования по мощности, которые могут часто меняться в зависимости от потребностей как производства, так и бизнеса. Известно [6], что не существует способов администрирования мощности, позволяющих управлять такими средами оптимальным образом по соотношению «стоимость к эффективности».

Целью данной работы является разработка и исследование имитационной модели распределенной среды при наличии априорной неопределенности о поступающей в среду информации, которая ляжет в основу разрабатываемых инструментальных средств для информационного обеспечения поддержки принятия решений в тех областях производства и науки, где адекватна эта модель. Модель позволит обеспечить информацию о предлагаемой мощности ресурсов и потребности в них, зависящую от времени, что является необходимым при диспетчеризации выполнения процессов в распределенной среде, балансировке нагрузки, назначении пакетов запросов (заданий, заявок) на конкретные обслуживающие приборы (узлы распределенной среды), рекомендации по планированию размещения пакетов.

Решающий элемент инструментального средства и имитационная модель проектировались исходя из того, что принятие решений основано на проекционном алгоритме стохастической аппроксимации по Назину-Позняку [2], выбор которого обусловлен характером функции штрафа. Штрафная функция — это наибольшее предельное значение средних текущих потерь. Для обеспечения высокого уровня надежности обслуживания пакетов задач в программное обеспечение управляющей ЭВМ для РС должно быть включено ПО для диспетчеризации выполнения пакетов запросов, поступающих в среду между целевыми обрабатывающими узлами среды на основании адаптивного подхода. Это необходимо потому, что если часть процессоров перегружена, часть простаивает, то вычислительный процесс организован плохо. При малых нагрузках на распределенную среду это не имеет значения, а в то же время при большой нагрузке низкая величина пропускной способности среды приведет к отказам в обслуживании. Функция потерь системы на интервале времени (t_n, t_{n+1}) запишется как

$$\xi_n = C_1(Q_n^{BX} - Q_n^{BYX}) + C_2 N_{PP}, \quad (1)$$

где Q_n^{BX} — число вошедших пакетов заданий, ед.; Q_n^{BYX} — обслуженных, ед.; C_1 — штраф за отказ в обслуживании заданий, у.е./ед.; C_2 — штраф за простой процессоров, у.е./ед.; N_{PP} — количество

простаивающих процессоров (загрузка которых менее $1/K$). Необходимо обеспечить безусловную минимизацию наибольшего предельного значения средних текущих потерь (1)

$$\Phi_n = \frac{1}{n} \sum_{t=1}^n \xi_t. \quad (2)$$

Для выполнения безусловной параметрической оптимизации по (2) в модели распределенной среды используется проекционный алгоритм стохастической аппроксимации с небинарными потерями

$$[2]: \quad p_{n+1} = \pi_{\varepsilon_{n+1}}^N \left\{ p_n - \gamma_n \frac{\xi_n - \Delta}{e^T(x_n) p_n} e(x_n) \right\}, \quad \text{где } \gamma_n \text{ — длина шага; } \Delta \text{ — числовой параметр; } \pi_{\varepsilon_n}^N \text{ —}$$

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.

Рассматривается разработанная имитационная модель *GRID*-системы на основе описанных в литературе существующих систем [4, 5]. Архитектура аппаратного решения *GRID*-системы изображена на рисунке 1. *GRID*-система является одновременно как исполняющей системой, так и набором программных интерфейсов (*API*). Исполняющую часть реализуют программные модули клиентов, которые запускают приложения в среде. Они позволяют обернуть готовую программу и выполнять ее параллельно, если распараллеливание ведется функционально или по данным, что должно встречаться в подавляющем большинстве научных и инженерных задач. Из рисунка видно, что система представляет собой отдельные объединения рабочих узлов, каждое со своим сервером. Узел, производящий запуск параллельного задания, показан серым цветом (слева). Он ведет обмен данными только с серверами, не обращаясь напрямую к рабочим узлам, что позволяет скрыть аппаратную архитектуру системы и абстрагироваться от ее состава. На каждом сервере установлен программный контейнер с функционирующими в нем программными сервисами. Таким образом, программное средство в *GRID*-среде позволяет использовать неограниченное количество рабочих узлов, объединенных серверами, для параллельного исполнения приложений в *GRID*-среде. Имитационная модель, разработанная в программной среде *AnyLogic*, показана на рисунке 2. Разработанная имитационная модель включает в себя разработанный блок принятия решений на основе проекционного алгоритма стохастической аппроксимации. Имеется реализация основных блоков *GRID*-среды и наглядное представление текущих результатов функционирования модели. Результаты моделирования показаны на рисунке 3. Просматривается зависимость средних текущих потерь от характера заявок (пакетов запросов) — «дорогие» или «дешевые».

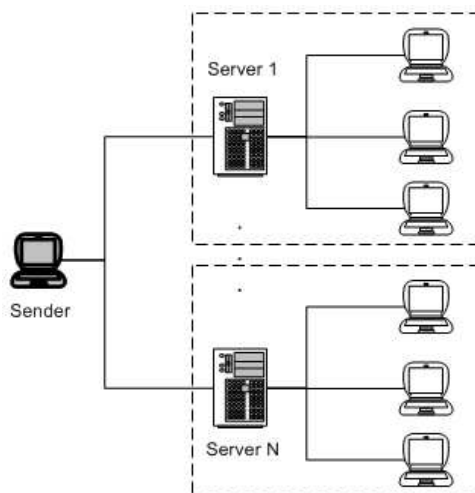


Рисунок 1 — Архитектура *GRID*-системы (из [5])

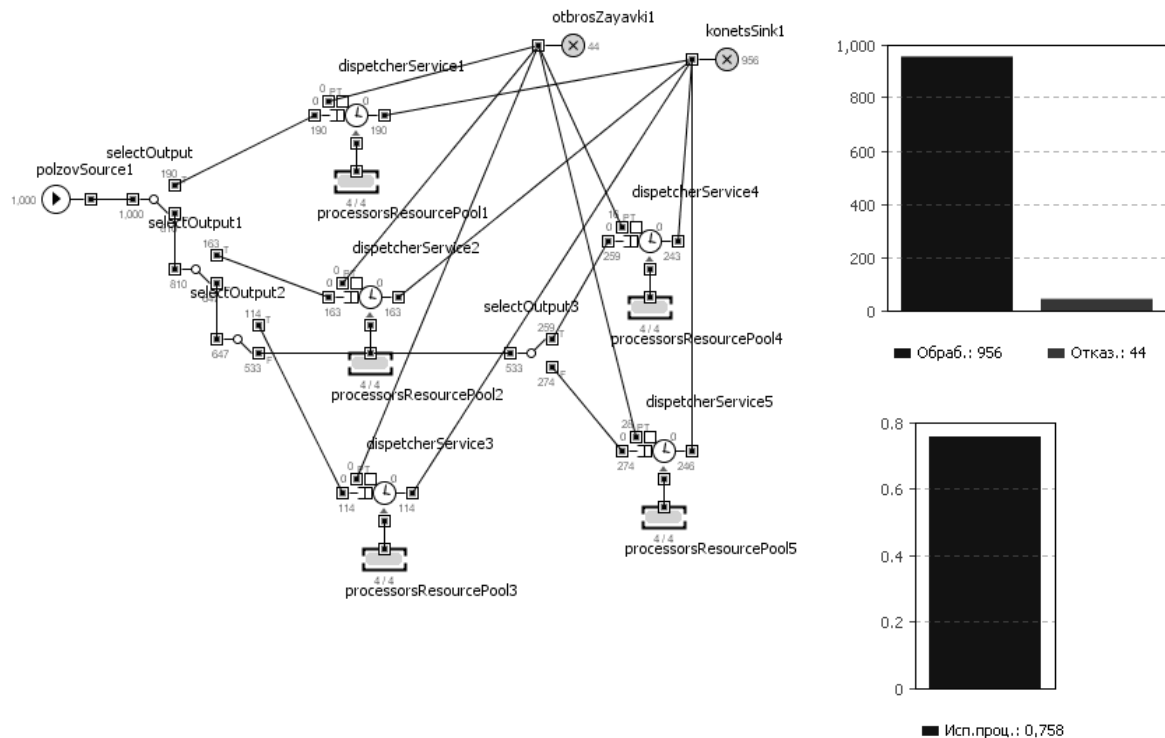


Рисунок 2 — Уточненная имитационная модель GRID-системы в среде AnyLogic

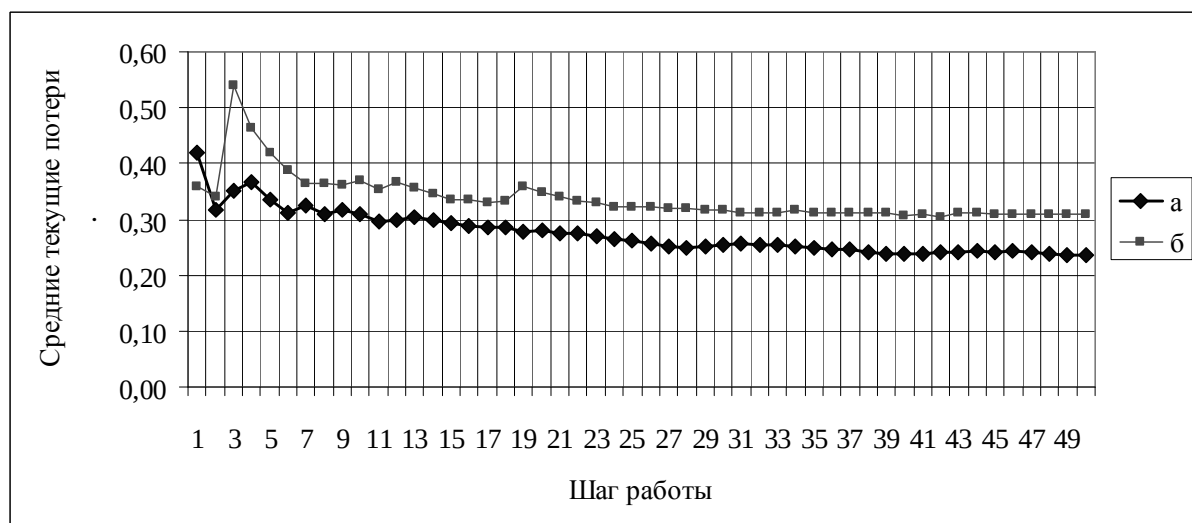


Рисунок 3 — Результаты моделирования для уточненной имитационной модели (а — «дорогие» заявки, б — «дорогие» процессоры)

Результатами данной работы явились разработанная имитационная модель и инструментальное средство информационного обеспечения принятия решений в условиях неопределенности по трудоёмкости работ. Предложенный комплекс программного обеспечения позволяет улучшить сходимость при калибровке имитационной модели, ввести понятийное описание произвольной распределенной среды с позиций системного подхода, получить выражения для функции штрафа, исключить противоречия, связанные с неопределенностью входных данных. Перспективой дальнейших исследований станет дальнейшая детализация инструментального средства, в том числе имитационной модели.

Бibliографический список использованной литературы

1. Ткаченко К.С. Задача диспетчеризации программ для многопроцессорных вычислительных систем / К.С. Ткаченко // Автоматизация: проблемы, идеи, решения: материалы міжнар. наук.-техн. конф., Севастополь, 5–9 вересня 2011 р. — Севастополь, 2011. — С. 266–267.

2. Назин А.В. Адаптивный выбор вариантов. Рекуррентные алгоритмы / А.В. Назин, А.С. Позняк. — М.: Наука, 1986. — 288 с.
3. Топорков В.В. Модели распределенных вычислений / В.В. Топорков. — М.: Физматлит, 2004. — 320 с.
4. Жижимов О.Л. Модель распределенной информационной системы Сибирского отделения РАН на базе протокола Z39.50 [Электронный ресурс]. / О.Л. Жижимов, Н.А. Мазов // Электронные библиотеки, 1999, Том 2, Выпуск 2. — Режим доступа: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/1999/part2/zhizhimov> Tuesday, September 27 2011 10:30.
5. Дунаев А.В. Моделирование параллельных вычислительных процессов в среде Грид на примере Intel Grid Programming Environment [Электронный ресурс] / А.В. Дунаев, А.В. Ларченко, А.В. Бухановский // Международная научная конференция «Параллельные вычислительные технологии» (ПАВТ'2008). — Режим доступа: <http://www.ict.edu.ru/vconf/files/11503.pdf> Tuesday, September 27 2011 10:35.
6. 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.

Поступила в редакцию 21.10.2011 г.

Ткаченко К.С. Адаптивне управління розподіленням середовищем на базі імітаційної моделі GRID-системи

Розглядається адаптивне управління розподіленням середовищем на базі імітаційної моделі GRID-системи для забезпечення безумовної мінімізації найбільшого граничного значення середніх поточних втрат. Наведено необхідні рисунки.

Ключові слова: розподілено середовище, імітаційна модель, адаптивне управління.

Tkachenko K.S. Adaptive control of distributed environment based on simulation model of GRID-system

Considered the adaptive control of a distributed environment based on a simulation model of GRID-system for unconstrained minimization of the maximum average current limit losses. Necessary figures are given.

Keywords: distributed environment, simulation model, adaptive control.