структуры программы. Имеется возможность определить условия эффективной работы комплекса, а также максимальное значение числа процессоров для подобных структур.

Заключение

Предлагаемый программный комплекс содержит три модели типовых многопроцессорных систем. Он используется при выполнении лабораторных работ по дисциплине «Высокопроизводительные вычислительные системы» студентами направлений 230100 и 231000. Имитационные модели разработаны с использованием универсальных сред (Delphi и C++). Они являются упрощенными и воспроизводят основные элементы структур и режимов функционирования объектов, что обеспечивает простоту усвоения материала и позволяет определять наиболее оптимальные параметры структур и режимов. Важной особенностью моделей является визуализация исследуемых процессов. Она обеспечивает максимальную наглядность и оптимальный режим обучения.

Литература

1. Таненбаум, Э. Архитектура компьютера: пер. с англ. / Э. Таненбаум.- Изд. 5-е.- СПб., 2010. - 848 с.

2. Хамахер, К. Организация ЭВМ: пер. с англ. / К. Хамахер, З.Вранешич, С. Заки; Сер.: Классика computer science.- Изд. 5-е.; - Спб: Питер, 2003г. - 845 с.

3. Орлов С.П. Организация компьютерных систем: учебное пособие/С.П. Орлов, Н.В. Ефимушкина. – Самара: Самар. гос. техн. ун-т, 2011. – 188 с.

Е.Е. Ивашко, А.С. Румянцев, А.Л. Чухарев, А.С. Головин

ЗАДАЧА ПОВЫШЕНИЯ ЭНЕРГОЭФФЕКТИВНОСТИ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

(Федеральное государственное бюджетное учреждение науки
Институт прикладных математических исследований
Карельского научного центра Российской академии наук)

Значительная часть современных бизнес-процессов происходит с использованием информационных технологий. При этом обработка данных, используемых или порожденных такими процессами, происходит в центрах обработки данных. Согласно определению Ассоциации производителей в области телекоммуникаций (TIA), центр обработки данных (далее ЦОД) — это здание (или его часть), включая оборудование для обработки данных, размещенное в машинном зале, вспомогательную инженерную инфраструктуру (управление электропитанием, обеспечение бесперебойного и резервного питания), коммуникационное оборудование, системы создания необходимых условий окружающей среды (охлаждение, осушение, очистка воздуха), системы пожаротушения, средства обеспечения безопасности [1].

Большая мощность вычислительного и вспомогательного оборудования приводит к значительному объему потребляемой электрической энергии, стоимость которой составляет около 40% общих эксплуатационных затрат ЦОД, что является сдерживающим фактором для широкого их применения. В то же время, значительная часть ЦОД является не загруженными задачами (в связи с резервом мощности, заложенным на стадии создания ЦОД), а нагрузка во времени имеет нестационарный характер. Это позволяет проводить исследования по экономии энергозатрат.

Современное оборудование ЦОД, включая сервера и системы хранения данных, имеет ряд режимов функционирования, среди которых, как правило, присутствуют так называемые режимы пониженного энергопотребления. Для описания состояний энергопотребления устройств применяется спецификация ACPI, поддерживаемая большинством производителей оборудования [2]. Стандарт ACPI позволяет согласованно производить действия по снижению энергопотребления как на уровне базовой (аппаратной) системы управления оборудованием BIOS, так и на уровне операционной системы (ОС). Работа вычислительного оборудования рассматривается в ACPI как переход между режимами, каждый из которых характеризуется уровнем энергопотребления и степенью готовности к работе [3]. Режимы работы подразделяются на глобальные и локальные. Глобальный режим характеризует степень готовности системы вцелом. Режим G0 соответствует режиму готовности, однако в этом режиме отдельные устройства могут переходить в режим снижения энергопотребления. В частности, доступны режимы C0, C1, C2 и C3 для центрального процессора (ЦП) системы. Режим G1 представляет собой режим сна. В этом режиме вычисления приостанавливаются. Стандарт ACPI различает 4 режима сна (S1,S2,S3,S4), каждый из которых характеризуется величиной энергопотребления и скоростью возвращения в режим G0 (больший номер режима соответствует меньшему энергопотреблению и большему времени выхода в режим готовности). Кроме того, значительная часть современных центральных процессоров поддерживает методы снижения частоты процессора. Важно отметить также, что энергопотребление вспомогательного оборудования (такого как источники бесперебойного питания, системы охлаждения) зависит от энергопотребления вычислительной инфраструктуры. Это означает, что наибольшего прироста экономии можно достигнуть путем снижения энергопотребления вычислительных ресурсов.

Отметим некоторые современные программные разработки, позволяющие достигать указанного снижения энергопотребления. Система Intelligent Power Software Suite компании Eaton [4] позволяет управлять электрическими устройствами в режиме реального времени. Платформа Avocent (Emerson) Trellis [5] компании Avocent (подразделения крупнейшего поставщика электрооборудования Emerson) представляет собой программно-аппаратный комплекс управления инфраструктурой ЦОД. Trellis позволяет проводить вычислительные эксперименты по изменению конфигураций ЦОД, вести непрерывный мониторинг сети датчиков функционирования оборудования, управлять энерго-

эффективностью ЦОД, а также управлять распределением времени обслуживающего персонала. Программная система Prognose [6] компании Romonet позволяет проводить прогнозирование поведения ЦОД при измененных режимах функционирования, таких как измененная конфигурация оборудования, повышенная вычислительная нагрузка.

Основным сдерживающим фактором для получения указанной экономии является необходимость поддержания заданного уровня качества обслуживания. Действительно, в случае отсутствия указанных ограничений целесообразно перевести всю инфраструктуру ЦОДа в максимальный режим сна, а в случае необходимости оказания услуг использовать минимальное число серверов (один процессор одного сервера). Подобные схемы используются в некоторых менеджерах расписаний задач операционной системы Linux [7]. Однако большинство коммерческих ЦОДов указывают некоторые характеристики доступности оборудования для пользователей в договоре на обслуживание, что не позволяет применить такие «грубые» методы. Таким образом, единственным способом для повышения экономии электроэнергии без нанесения значительного ущерба качеству обслуживания является применение методов прогнозирования нагрузки ЦОДа. При этом в случае построения достаточно надежного прогноза, можно заранее готовить требуемое «будущим» задачам количество ресурсов, минимизируя риск нарушения условий договора.

Вычислительный кластер (ВК) является примером ЦОД. Особенность ВК состоит в том, что множество вычислительных ресурсов (процессоры, оперативная память, системы хранения, быстрая коммуникационная сеть) воспринимаются пользователями ВК как единый аппаратный ресурс. Доступ к ВК осуществляется удаленно через локальную сеть организации, либо глобальную сеть Интернет в режиме разделения нагрузки между пользователями с использованием менеджера очередей. Работа с кластером ведется в режиме сессии, в течение которой пользователь осуществляет постановку заданий, затем может прервать сессию в ожидании результата. Отметим, что ВК являются лидерами по энергопотреблению среди ЦОДов.

В [8] предложена модель прогнозирования нагрузки ЦОД с разделяемым между n пользователями $\{u_1, \ldots, u_n\}$ доступом. Обозначим $\tau_i(u), i \geq 1$ время начала сессии доступа и $\sigma_i(u)$ продолжительность сессии с номером i для пользователя u. Активные в момент времени t сессии пользователя u (т.е. такие, что $\tau_i(u) \leq t \leq \tau_i(u) + \sigma_i(u)$) порождают поток заявок, каждая из которых характеризуется временем прихода $t_j(u)$, требуемым временем обслуживания $S_j(u)$ (которое может быть заранее неизвестно, либо известно приближенно) и, возможно, дополнительными параметрами (ограничениями на ресурсы). Предположим, что каждое устройство имеет несколько режимов, C_0 (режим обычной работы), $C_1, \ldots, C_k$ (РПЭ, больший индекс соответствует большей экономии). Режиму C_i соответствует величина минимального времени использования этого режима t_i (t_i связано с временем ухода и возвращения из состояния C_i). Если в момент времени T в системе имеется свободный ресурс из K устройств, а в момент i в соответствии с прогнозом ожидается уменьшение свободного ресурса уст-

ройств (например, в результате прихода новой заявки) на $\dot{N}$ единиц, то на основании прогноза $\dot{t}, \dot{N}$ необходимо построить вектор $\kappa = (\kappa_0, \ldots, \kappa_k) \in \mathbb{Z}_+^{k+1}$, такой что $\kappa_0 + \cdots + \kappa_k = K$, в котором κ_i есть количество устройств, которое необходимо перевести в состояние C_i, а также вектор $\bar{t} = (\dot{t}_0, \ldots, \dot{t}_k), 0 \leqslant \dot{t}_i \leqslant \dot{t}$, состоящий из моментов выхода из состояний $\{C_i\}$. Целесообразно производить построение прогноза в момент освобождения ресурсов (окончания обслуживания заявки) и корректировать прогноз в следующие ключевые моменты: приход или уход очередной заявки, начало или окончание сессии пользователя.

Рассмотрим систему, в которой очередь пуста. Тогда в момент построения прогноза T пользователь $u \in U$ является активным, если найдется индекс i такой, что $\tau_i(u) \leqslant T \leqslant \tau_i(u) + \sigma_i(u)$, в противном случае пользователь неактивен. Для активного пользователя u прогнозируется возможное время до прихода ближайшей заявки от данного пользователя, параметры заявки, а также длительность активной сессии. Для каждого неактивного пользователя прогнозируется длительность периода неактивности пользователя, а также время прихода ближайшей заявки и параметры заявки. При этом предполагается отсутствие сессий без постановки задачи в очередь. Тогда прогнозируемые значения $\dot{t}, \dot{N}$ могут быть вычислены как значения функционалов

$$\dot{t} = \varphi(\dot{t}_1(U), \ldots, \dot{t}_n(U)), \dot{N} = \nu(\dot{N}_1(u), \ldots, \dot{N}_n(u)),$$

где $\dot{t}_i(U)$ есть вектор ближайших прогнозируемых приходов заявок пользователей U после момента времени T, а $\dot{N}$ есть вектор прогнозируемых уменьшений ресурсов. В простейшем случае можно полагать $\varphi(\cdot) = \nu(\cdot) = \min(\cdot)$. Функционал $\nu(\cdot)$ можно определить аналогично. Отличие случая с непустой очередью заключается в том, что кроме вычисления прогнозируемых векторов $\dot{t}_i(U), 1 \leqslant i \leqslant n$ приходов ближайших заявок и размеров уменьшения ресурса $\dot{N}_i(U)$, необходимо также прогнозировать ближайшее время завершения вычисления очередной заявки (среди заявок, уже занимающих ресурсы).

Предложенная методика может быть использована ретроспективно для оценки возможного эффекта от внедрения системы по имеющимся лог-файлам параметров сессий и заявок, а также для подсчета максимально возможной экономии.

Литература

1. Стандарт TIA-942 [Электронный ресурс]. URL: http://www.in-systems.ru/upload/iblock/23a/tia942.pdf (время доступа: 20.08.2013)

2. ACPI - Advanced Configuration and Power Interface [Электронный ресурс]. URL: http://www.acpi.info/ (время доступа: 15.02.2013).

3. ACPI Overview [Электронный ресурс]. URL: http://www.acpi.info/presentations/ACPI_ Overview.pdf (время доступа: 15.02.2013).

4. Intelligent Power Software Suite [Электронный ресурс]. URL: http://powerquality.eaton.com/Products-Services/Power-Management/Software-Drivers/Intelligent-Power-Suite.asp (время доступа: 20.08.2013)

5. The Trellis Dynamic Infrastructure Optimization Platform [Электронный ресурс]. URL: http://www.emersonnetworkpower.com/en-US/trellis/Documents/Brochures/TRELLIS-BRO-EN.pdf (время доступа: 22.08.2013)

6. Romonet: The Holy Grail of Data Center TCO [Электронный ресурс]. URL: http://www.romonet.com (время доступа: 22.08.2013)

7. Srinivasan V. et al. Energy-aware task and interrupt management in linux // Ottawa Linux Symposium. 2008.

8. Ивашко Е. Е., Румянцев А. С., Чухарев А. Л. Задача прогнозирования нагрузки для повышения энергетической эффективности вычислительного кластера // Параллельные вычислительные технологии (ПаВТ'2013): труды международной научной конференции (г. Челябинск, 1-5 апреля 2013 г.). Челябинск, Издательский центр ЮУрГУ, 2013. С. 363-370.

В.В. Козлов

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ НАТИВНЫЕ СЕРВЕРНЫЕ WEB ТЕХНОЛОГИИ

(Самарский государственный архитектурно-строительный университет)

Распространенные WEB технологии (PHP, ASPX, NodeJs, Python, …) являются скриптовыми и удовлетворительно работают при низких нагрузках на сервер. При повышении нагрузки требуется горизонтальное масштабирование серверного оборудования, то есть создания серверных ферм, что влечет за собой значительное удорожание поддержки таких сайтов. Таким образом, можно выделить основные недостатки скриптовых WEB языков:

1) низкая производительность на один сервер – предопределено динамической структурой скриптовых языков программирования;
2) снижение эффективности кеширования контента и значительное увеличение трудоемкости разработки при горизонтальном масштабировании системы (увеличении количества серверов).

Технология Java выходит за рамки скриптовых языков. Однако использование Java локализовано в так называемом Enterprise секторе для корпоративных приложений, а самым массовым WEB языком, в настоящее время, является PHP.

Одними из наиболее крупных потребителей и движущей силой PHP технологии являются FaceBook и Vkontakte (Россия). Однако, даже перед этими компаниями достаточно давно остро встал вопрос недостаточной производительности применяемых ими технологий.

Рассмотрим историю решения вопроса производительности информационных систем FaceBook [1] (состояние вопроса на 2009 год). В 2010 компания опубликовала технологию HipHop [2], суть которой состояла в трансляции языка PHP в код на C++ с последующей его компиляцией, что позволило поднять