

3. Лебедев Б.К., Дуккардт А.Н. Разбиение на основе комбинированных генетических процедур // Известия ТРТУ. – 2006. – № 8 (63). – С. 46-51.
4. Дуккардт А.Н. Решение задачи разбиения на основе процедуры «Выбивания» // Известия ТРТУ. – 2006. – № 6 (61). – С. 63-66.
5. Лежебоков А.А. Генетический алгоритм решения задачи размещения элементов СБИС с учетом временных задержек // Сборник научных трудов международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте». Т. 1. – М.: Физматлит, 2007. – С. 429-436.
6. Лежебоков А.А., Гладков Л.А. Результаты компьютерного моделирования решения задачи размещения элементов СБИС с учетом временных задержек // III Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем– 2008» (МЭС-2008), 2008 / Сборник научных трудов под общ. ред. А.Л. Стемпковского. – М.: ИППМ РАН, 2008. – С. 130-136.
7. Курейчик В.М., Биоинспирированный поиск с использованием сценарного подхода. // Известия ЮФУ. Технические науки. – 2010. – № 7 (108).– С. 7-12.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.О. Чернышев.

Лежебоков Андрей Анатольевич

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: legebokov@gmail.com.

347928, г. Таганрог, Некрасовский, 44.

Тел.: 88634371651.

Кафедра систем автоматизированного проектирования, к.т.н., ассистент.

Дуккардт Александр Николаевич

E-mail: aduckardt@gmail.com

Кафедра систем автоматизированного проектирования, к.т.н., ассистент.

Lezhebokov Andrey Anatolyevich

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: legebokov@gmail.com.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +78634371651.

The Department of Computer Aided Design; Cand. of Eng. Sc.; Assistant.

Duckardt Alexander Nikolaevich

E-mail: aduckardt@gmail.com

The Department of Computer Aided Design; Cand. of Eng. Sc.; Assistant.

УДК 681.3.016:658.512.2.011.5:519.711.3

О.С. Коваленко

ФОРМАЛИЗАЦИЯ ПРОБЛЕМЫ ОПТИМИЗАЦИИ РАЗМЕЩЕНИЯ ДАННЫХ САПР В «ОБЛАКЕ»*

Рассмотрена формализованная постановка задачи оптимизации размещения данных САПР в «облаке» в соответствии с предложенным иерархическим представлением структуры сложной САПР. Определены и описаны основные этапы и проблемы решения данной задачи на различных уровнях организации системы, определены исходные данные задачи. На каждом этапе решения задачи оптимизации вводятся необходимые ограничения. Раз-

* Работа выполнена при частичной поддержке РФФИ (проект № 11-01-00122).

работана математическая модель размещения данных для взаимосвязи уровней функциональной и аппаратной конфигурации, размещения фрагментов данных. Выбраны критерии эффективности и выведена формула общего критерия эффективности для задачи оптимизации размещения данных в узлах «облака».

«Облачное» хранение данных; размещение данных; метод оптимизации.

O.S. Kovalenko

CAD SYSTEM'S DATA ACCOMMODATION IN «CLOUD» OPTIMIZATION PROBLEM FORMALIZATION

It is considered formalized statement of optimization of the accommodation data of the system of CAD in the «cloud» according to the offered hierarchical representation of the structure complex CAD system in this article. The basic stages and problems of the decision of the given problem at several of the system organization levels are defined and described. Input data are declared. Required constraints are introduced in each stage of the optimization problem decision. Mathematical model of the data accommodation is designed for the relationship between levels of functional and hardware configuration and data fragments accommodation. It is chosen efficiency criterions and deduced a formula complex efficiency criterion for the problem of the optimization of the data accommodation in «cloud» hosts.

«Cloud» data storage; accommodation of data; the method of optimization.

Введение. Общей проблемой внедрения облачных сервисов является потребность в компромиссе между функциональными возможностями и эксплуатационными расходами. В современных начальных сервисах данных обеспечиваются API, намного более ограниченные, чем в традиционных системах баз данных, с минималистским языком запросов и ограниченными гарантиями согласованности. Это затрудняет программирование приложений, но позволяет создавать более предсказуемые службы облачных сервисов, в которых обеспечиваются соглашения об уровне обслуживания, трудно достижимые для полнофункциональных сервисов данных на основе языка SQL. Требуется дополнительная работа и накопление опыта в нескольких направлениях для исследования непрерывного спектра подходов между ранними облачными сервисами данных и более развитыми с функциональной точки зрения, но, возможно, менее предсказуемыми альтернативами [1].

По сравнению с традиционными системами, достижение высокого уровня управляемости, которая является особенно важным качеством, в облачных средах осложняется тремя факторами: ограниченным человеческим вмешательством, значительным разбросом диапазона рабочих нагрузок и разнообразием совместно используемых инфраструктур. В подавляющем большинстве случаев администрирование платформ должно будет в основном производиться в автоматическом режиме. Со временем может значительно измениться рабочая нагрузка даже у одного и того же потребителя. При этом возможности настройки сервисов зависят от способа «виртуализации» совместно используемой инфраструктуры. В каждом из решений «виртуализации» обеспечивается свой подход к контролю поддерживаемых рабочих нагрузок и используемых платформ. При наличии этих вариантов требуется пересмотреть традиционные роли и распределение ответственности для многоуровневого управления ресурсами. Потребность в управляемости делает более срочной разработку технологий самоуправления баз данных, которые исследовались в последнее десятилетие. Для обеспечения жизнеспособности этих систем потребуются адаптивные онлайн-методы [1,2].

Архитектура системы при использовании облачных вычислений является логическим продолжением и обоснованным аппаратно-программным решением для построения единого информационного пространства, объединяющего потоки дан-

ных. Все функциональные элементы перенесены внутрь облака и скрыты за кольцом интерфейсов доступных сервисов, а само облако при этом является интеллектуальным источником распределения данных и построения сложной архитектуры системы [2].

Создание сложных компьютерных интеллектуальных аналитических систем для САПР, функционирующих в «облаке» предполагает, прежде всего, подготовку «облачной» САПР и БД к работе такими системами. Работа «облачной» САПР требует, в свою очередь, поддержания необходимой функциональности распределенных по серверам «облака» программных модулей САПР и фрагментов БД САПР. Для решения этих задач создается «облачная» информационная система, которая обеспечивает своевременное поступление исходных данных и команд, необходимых системе и пользователю в процессе работы, передачу результатов работы, а также размещение данных среды САПР в информационной системе. В настоящее время актуальной является постановка и решение задачи оптимизации размещения данных среды САПР в «облачной» информационной системе автоматизированного проектирования [1–7].

Решение любой оптимизационной задачи предполагает следующие этапы:

- ◆ постановка проблемы;
- ◆ конструирование концептуальной модели;
- ◆ выбор критерия эффективности;
- ◆ построение математической модели;
- ◆ выбор алгоритма оптимизации;
- ◆ реализация;
- ◆ проверка модели;
- ◆ анализ полученных результатов.

На первом этапе для формализации процедуры нахождения оптимального размещения данных в «облаке» описание структуры САПР можно представить в виде иерархии нескольких уровней (рис. 1) [1–4,7]:

- ◆ уровень аппаратной конфигурации, который представлен серверами и другим оборудованием «облака», а также набором характеристик, таких как объем оперативной памяти, скорость передачи данных, частота процессора, объем видеопамати и т.д.;
- ◆ уровень функциональной конфигурации, означающий характер распределения программного обеспечения и программных модулей САПР по узлам вычислительной сети внутри «облака», т.е. элементам аппаратной конфигурации, при этом каждый программный модуль характеризуется составом данных, необходимых для его функционирования и правильной работы приложений;
- ◆ уровень информационной модели, на котором представлен состав данных, необходимых для функционирования системы в целом и отдельных приложений и программных модулей в частности, а также БД.

Из рисунка (см. рис. 1) видно, что на различных элементах аппаратной конфигурации, т.е. серверах «облака», может быть размещен различный набор функциональных модулей, которые требуют в свою очередь различного состава данных уровня информационной модели (это показано изображением блоков, имеющих различную степень окраски, в том случае, когда данные не требуются, блоки изображены белым фоновым цветом).

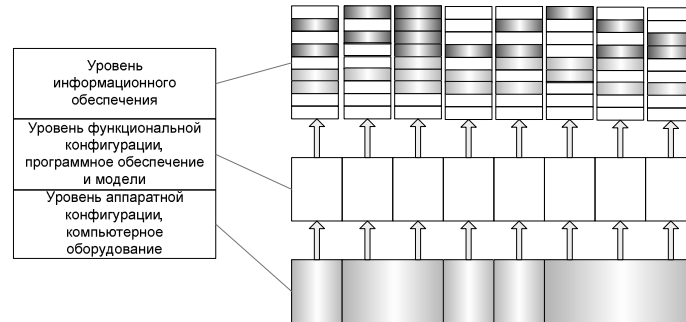


Рис. 1. Структура САПР в виде иерархии уровней [3]

В качестве исходных данных задачи размещения программных модулей САПР в «облаке» служит следующая информация:

- ◆ $PC_i, i = \overline{1, n}$ – количество серверов «облака», при этом каждый из них может характеризоваться быстродействием центрального и графического процессора, объемом оперативной памяти, возможностями сетевой подсистемы;
- ◆ $M_j, j = \overline{1, p}$ – набор программного обеспечения, модулей и компонент системы проектирования, которые располагаются в узлах вычислительной сети («облака»), т.е. элементах аппаратной конфигурации;
- ◆ $D_k, k = \overline{1, m}$ – набор структурированных данных определенных типов и категорий на уровне информационного обеспечения, которые представляют наполненные соответствующей информацией фрагменты БД.

Не менее важной компонентой исходной информации о системе сложной САПР являются выбранные проектные решения для организации межмашинного сетевого обмена, а также характеристики каналов связи и возможности доступных сервисов специализированного оборудования [3].

Для взаимосвязи уровней функциональной и аппаратной конфигурации вводится матрица распределения программных функциональных модулей и компонент САПР по элементам вычислительной сети «облака» $MC_{ij}, i = \overline{1, n}, j = \overline{1, p}$.

Элемент матрицы MC_{ij} может принимать следующие значения:

$$MC_{ij} = \begin{cases} 1, & \text{если } j\text{-ая компонента находится на } i\text{-ом сервере,} \\ 0, & \text{в противном случае.} \end{cases}$$

При этом подразумевается факт физического расположения определенной программы или элемента системы проектирования на определенном сервере «облака».

Возможны ситуации, когда на сервере располагаются сразу несколько таких модулей, что позволяет экономить аппаратные средства и осуществлять процесс проектирования исходя из конкретных целей и задач. Например, это возможно при наличии нескольких мониторов или средств поддержки проектирования в составе программного обеспечения САПР, не использующих средства визуализации и прямого взаимодействия с пользователями системы. Очевидно, что на каждый физический сервер, входящий в состав «облака», должна быть помещена хотя бы одна компонента САПР. В математическом виде данное условие может быть записано как

$$\sum_{j=1}^p MC_{ij} \geq 1, i = \overline{1, n}.$$

Ограничением на данном этапе формализации может являться существование единственной копии уникального сервера или специализированного модуля программного обеспечения в комплексной системе САПР, в том случае, если наличие таких ограничений не снимается спецификой и задачами функционирования конкретного процесса проектирования. Например, для повышения надежности работы системы или при работе с одним проектом несколькими операторами одновременно:

$$\sum_{j=1}^n MC_{ij} = 1, \quad i = \overline{1, p}.$$

Состав данных уровня информационной модели каждого отдельного физического элемента аппаратной конфигурации определяется совокупным составом данных всех компонент уровня функциональной конфигурации, расположенных на данном сервере.

Для этого необходимо ввести матрицу определения потребности данных для каждого программного модуля и пользователя системы. Соотношения матрицы соответствуют факту потребностей определенного элемента или самостоятельного компонента программного обеспечения уровня функциональной конфигурации данных и объектов проектирования, представленных уровнем информационного обеспечения процесса проектирования, т.е. матрицей вида MD_{jk} , $j = \overline{1, p}$, $k = \overline{1, m}$.

Элемент матрицы MD_{jk} может принимать следующие значения:

$$MD_{jk} = \begin{cases} 1, & \text{если } j\text{-му пользователю (модулю) нужны } k\text{-ые данные,} \\ 0, & \text{в противном случае.} \end{cases}$$

Из данного представления видно, что одному элементу уровня функциональной конфигурации могут соответствовать несколько типов объектов.

При этом каждый существующий на начало проектирования фрагмент данных должен быть необходим, по крайней мере, одному модулю или элементу программного обеспечения, в противном случае сама процедура его размещения в общем информационном пространстве становится бессмысленной.

Перечисленные выше утверждения и описания исходных составляющих так называемой базы данных конфигурирования системы могут служить исходными для процедур и модулей, позволяющих строить оптимальные стратегии распределения фрагментов информационной модели сложной САПР в «облаке».

Следующим этапом формализации данной задачи может стать создание на основе полученных исходных данных начальной матрицы размещения фрагментов данных. В этом случае делается предположение о том, что данные должны быть распределены с наиболее возможной максимальной потребностью для каждого физического узла системы, т.е. необходимо ввести аналог матрицы MD , но уже для уровня аппаратной конфигурации системы. Форма записи имеет вид C_{ik} , $i = \overline{1, n}$, $k = \overline{1, m}$.

Элемент матрицы C_{ik} может принимать следующие значения:

$$C_{ik} = \begin{cases} 1, & \text{если на } i\text{-ый сервер помещен } k\text{-ый тип данных,} \\ 0, & \text{в противном случае.} \end{cases}$$

Выполнение такого размещения в действительности должно согласовываться с рядом ограничений, например таких, как объем оперативной памяти каждого компьютера, отведенный для хранения и работы с фрагментами данных. Эти ограничения должны выбираться и согласовываться с особенностями функционирования

ния элементов САПР, так как одним из основных показателей отдельных фрагментов данных является их высокая динамичность, постоянное изменение объема данных и количества экземпляров. В связи с этим могут возникнуть дополнительные трудности и особенности разработки алгоритмов и моделей оптимизации.

Отсюда следует, что фактически данные должны быть помещены, по крайней мере, на один физический узел, и, в то же время, при необходимости находиться в реплицированном виде на других узлах. В отдельных случаях какой-то узел может не содержать фрагментов данных, как по экономическим соображениям, так и вследствие получившейся стратегии размещения данных.

После выполнения процедуры создания матрицы C необходимым условием является оценка принятого решения. Данная оценка может производиться расчетом времени, необходимого для такта функционирования системы. Это обусловлено тем, что одной из основных особенностей систем реального времени являются жесткие временные ограничения, которые необходимо соблюдать при проектировании распределенной информационной системы.

Данная процедура может быть выполнена расчетом временной матрицы для соотношения программного модуля (потребителя определенного типа объекта проектирования) и собственно данных (источника информации – объект проектирования) с учетом физического размещения данных, принятого матрицей C , т.е. вычисляется матрица $T_{jk}, j = \overline{1, p}, k = \overline{1, m}$.

Элементы данной матрицы принимают значения в соответствии со следующими вариантами:

- ◆ модулю $M_j, j = \overline{1, p}$ необходимы для своего функционирования данные $D_k, k = \overline{1, m}$, при этом они размещены на том же физическом узле конфигурации, что и модуль, тогда время T_{jk} вычисляется с учетом работы непосредственно в оперативной памяти компьютера;
- ◆ модулю $M_j, j = \overline{1, p}$ необходимы для своего функционирования данные $D_k, k = \overline{1, m}$, при этом они размещены на другом физическом узле конфигурации, тогда время T_{jk} вычисляется с учетом работы сетевой подсистемы;
- ◆ модулю $M_j, j = \overline{1, p}$ не нужны для своего функционирования данные $D_k, k = \overline{1, m}$, тогда время T_{jk} принимается равным 0.

Существенной проблемой при этом будет оценка времени получения данных сетевой подсистемой при различном количестве информации, текущей загрузке сети, используемом протоколе обмена и методе реализации алгоритмов работы с данными. Таким образом, с учетом введенных условных предположений можно рассчитать суммарное значение времени. Другим немаловажным показателем является суммарная потребность в памяти для данного варианта размещения данных. Критерием эффективности может являться оценка, складывающаяся из двух составляющих:

- ◆ минимальное отношение значения времени доступа к распределенному информационному ресурсу САПР на отдельном узле и (или) «облаке» в целом к эталонному значению времени доступа;
- ◆ минимальное отношение значения потребности в используемой памяти для хранения структур и объектов САПР на отдельном узле и (или) «облаке» в целом к эталонному значению потребности в используемой памяти.

В общем виде критерий эффективности F' для функции времени можно представить следующим образом:

$$F^1 = \sum_{j=1}^p \sum_{k=1}^m \frac{T_{jk}}{T_{\ominus}} \rightarrow \min ,$$

где T_{\ominus} – эталонное время доступа.

Для критерия эффективности используемой памяти F^v , выражение принимает вид

$$F^v = \sum_{i=1}^n \frac{V_i}{V_{\ominus}} \rightarrow \min ,$$

где V_i , $i = \overline{1, n}$, – суммарное значение памяти, занимаемой объектами САПР на отдельном узле и (или) «облаке» в целом, V_{\ominus} – эталонное значение памяти.

Тогда общий критерий эффективности F принятого размещения данных в системе

$$F = w_1 F^1 + w_2 F^v \rightarrow \min ,$$

где w_1, w_2 – аналоги весовых коэффициентов, позволяющих дифференцировать важность того или иного критерия в общем представлении, $w_1 + w_2 = 1$, $w_1 = \overline{0, 0,5}$, $w_2 = \overline{0,5, 1}$.

При этом взаимодействие с сетевой подсистемой не ограничивается только получением необходимых данных при условии того, что программному модулю M_j , $j = \overline{1, p}$ необходимы для своего функционирования данные D_k , $k = \overline{1, m}$ и они физически находятся на другом узле, а также необходимо для поддержания всего информационного пространства в адекватном состоянии на каждый момент времени. Например, при изменении параметров объектов САПР данные должны быть обновлены не только в локальных структурах определенного сервера состава аппаратной конфигурации, но и доставлены всем остальным реплицированным компонентам, находящимся в распределенных структурах. Своевременное выполнение таких операций позволяет поддерживать информационное пространство САПР и наполнение процесса проектирования для всех участников.

Из этого следует, что чрезмерное дублирование элементов программных модулей может отразиться возрастающими потребностями дополнительного обновления отдельных структур, интенсивно меняющих свои параметры в процессе работы системы. В то же время наличие статичных структур большого объема также не позволяет тратить оперативную и дисковую память для их хранения на многих узлах, и предпочтительнее возложить функции работы с ними на сетевую подсистему, используя их как элементы удаленных сервисов данных.

Тогда задачей моделей оптимизации является определение эффективного размещения данных с учетом наличия дублирования в тех случаях, где это не повлечет за собой значительных затрат на повторное обновление отдельных структур и создание одиночных экземпляров данных для работы с ними через удаленные интерфейсы для сокращения лишнего сетевого трафика. Выполнение подобных условий позволит создать максимально сбалансированную стратегию размещения данных проектирования в «облачной» САПР с учетом введенных ограничений.

В отличие от рассмотренной глобальной проблемы оптимизации размещения в рамках всей распределенной структуры «облачной» САПР также можно выделить локальную задачу оптимизации топологии структур для хранения данных программных модулей на отдельном узле системы. В данном случае при использовании современных технологий программирования и разработки приложений производится формирование динамических структур типа списков и деревьев в оперативной памяти компьютера. При этом при организации последующего доступа к данным необходимо учитывать дополнительные временные затраты на обход и поиск нуж-

ных элементов. Для уменьшения времени поиска необходимой информации в этих структурах целесообразно предложить оптимизацию топологии дерева данных проектирования, которая учитывала бы факторы использования запросов. Это может достигаться путем дополнительных разбиений на отдельные поддеревья, группировки данных в соответствии с требованиями потребителей, введением специальных указателей с учетом специфики работы с определенными типами данных. Такое решение позволит минимизировать технологическую сторону в скорости получения информации пользователями и программными модулями уже на конкретном физическом элементе функциональной конфигурации САПР.

Введение математических постановок задачи на различных этапах функционирования системы с учетом того, что данные операции могут входить как функции в САПР и выполняться как фоновый процесс во время процесса проектирования, позволит разработать модели, алгоритмы и соответствующие компоненты программной реализации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Коваленко О.С.* Обзор состояний, проблем и перспектив хранения и анализа данных в «облаке» // Информатика, вычислительная техника и инженерное образование. – Таганрог: Изд-во ТТИ ЮФУ, в печати.
2. *Коваленко О.С.* Обзор проблем и состояний облачных вычислений // Информатика, вычислительная техника и инженерное образование. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – № 1 (3).
3. *Янюшкин В.В.* Формализация проблемы оптимизации размещения данных модельного мира в распределенной информационной системе тренажерно-моделирующего комплекса // Известия вузов. Северо-Кавказский регион. (Технические науки). – 2008. – № 4. – С. 25-28
4. *Коваленко О.С.* Обзор проблем и перспектив анализа данных // Информатика, вычислительная техника и инженерное образование. – Таганрог: Изд-во ТТИ ЮФУ, 2010. – № 2.
5. *Островский Г.М., Волин Ю.М.* Технические системы в условиях неопределенности: анализ гибкости и оптимизация: Учебное пособие. – М.: Бином, 2008.
6. *Штойер Р.* Многокритериальная оптимизация. Теория, вычисления и приложения. – М.: Радио и связь, 1992.
7. *Янюшкин В.В.* Математические модели оптимизации распределенных информационных систем тренажерно-моделирующих комплексов: Дисс. ... канд. тех. наук. – Новочеркасск, 2010.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.О. Чернышев.

Коваленко Олеся Сергеевна

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге
E-mail: kovalenko.olesja@gmail.com.
347928, г. Таганрог, пер. Некрасовский, 44.
Тел.: 88634360793.

Кафедра систем автоматизированного проектирования; аспирант.

Kovalenko Olesya Sergeevna

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”
E-mail: kovalenko.olesja@gmail.com.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: +78634360793.
The Department of Computer Aided Design; Post-graduate Student.