

МЕТОДЫ ОРГАНИЗАЦИИ ГРИД-ОБОЛОЧЕК СИСТЕМНОГО СЛОЯ В ТЕХНОЛОГИИ CAE BEANS

Г.И. Радченко

Представлена технология построения иерархии проблемно-ориентированных грид-оболочек CAEBeans над инженерными пакетами. Дано описание структурной организации оболочек CAEBeans в виде четырех слоев: концептуального, логического, физического, системного. Рассмотрены методы организации оболочек CAEBeans системного слоя.

Ключевые слова: CAE, CAEBean, грид-оболочка, сервисно-ориентированная архитектура, системный слой, брокер ресурсов

Введение

Одним из наиболее перспективных методов сокращения сроков разработки сложной технологической продукции является применение программных систем класса CAE (Computer Aided Engineering), позволяющих эффективно осуществлять компьютерное моделирование разрабатываемых образцов. Такой подход позволяет значительно повысить точность анализа проектных вариантов продукции и проводить виртуальные эксперименты, которые в реальности выполнить затруднительно или вовсе невозможно.

Каждый CAE-пакет представляет собой программный комплекс со сложным технологическим циклом, который может включать в себя следующие этапы: подготовка геометрии модели исследуемой области; построение вычислительной сетки; определение физики протекающих процессов; решение поставленной задачи моделирования; визуализация и анализ полученных результатов. В настоящее время на первое место выходит точность компьютерного моделирования и достоверность получаемых результатов. Это обуславливает постоянный рост размерностей сеток, используемых для проведения вычислительных экспериментов. На сегодняшний день размерность сеток, используемых в задачах компьютерного моделирования, может превышать десятки миллионов элементов [1]. В связи с этим постоянно возрастает вычислительная сложность, и требуются значительные вычислительные ресурсы для проведения экспериментов компьютерного моделирования.

Практически все современные CAE-пакеты имеют параллельные реализации для многопроцессорных систем, в том числе и для систем с кластерной архитектурой. Современный опыт использования суперкомпьютерных систем показывает, что максимальная эффективность использования вычислительных ресурсов может быть достигнута при объединении таких систем в вычислительные грид-сегменты [2, 3]. Таким образом, важной темой исследования является интеграция CAE в грид-среды.

При реальном внедрении CAE-систем в распределенную вычислительную среду возникает целый ряд проблем. В общем случае базовые компоненты CAE-пакетов, реализующие отдельные этапы технологического цикла постановки и решения задач, не поддерживают возможности удаленной постановки задачи и предоставления результата удаленному пользователю. Также необходимо решить задачи прозрачного использования удаленных грид-ресурсов, в том числе автоматизацию поиска, выбора и использования вычислительных ресурсов, требуемых для решения конкретных задач.

В качестве перспективного подхода к решению задач внедрения современных САЕ-систем в распределенные вычислительные среды, предлагается применение технологии CAEBeans [4, 5]. В основе технологии CAEBeans лежит обеспечение сервисно-ориентированного предоставления программных ресурсов базовых компонентов САЕ-систем и формирование деревьев проблемно-ориентированных оболочек, инкапсулирующих процедуру постановки и решения определенного класса задач. Таким образом, процедура постановки и решения задачи с применением технологии CAEBeans сводится к указанию пользователем значений параметров в терминах той проблемной области, в рамках которой проводится решение задачи. Технология CAEBeans автоматизирует декомпозицию задач на типовые подзадачи; поиск вычислительных ресурсов, обеспечивающих оптимальное решение поставленной задачи в грид-среде; постановку задач соответствующим базовым компонентам САЕ-систем; мониторинг хода решения задач; подготовку и предоставление результатов решения задачи пользователю.

Статья организована следующим образом. В разделе 1 приводится описание основных концепций технологии CAEBeans, раскрывается понятие дерева проблемно-ориентированных оболочек, дается определение системы CAEBeans. В разделе 2 рассматриваются основные особенности слоистой архитектуры системы CAEBeans, раскрываются основные функциональные особенности концептуального, логического и физического слоев CAEBeans. Раздел 3 посвящен детальному анализу архитектуры и основных функциональных особенностей системного слоя CAEBeans. В разделе 4 приведены результаты разработки оболочек CAEBeans. В заключении суммируются основные результаты и описаны направления дальнейшей работы.

1. Основные концепции технологии CAEBeans

1.1. Дерево проблемных оболочек CAEBeans

Технология CAEBeans ориентирована на инкапсуляцию и предоставление ресурсов САЕ-пакетов в рамках гетерогенных высокопроизводительных сред, что позволяет организовать эффективное использование САЕ-пакетов широким кругом пользователей, посредством автоматизации процесса решения типовых задач. В основе данного подхода лежит процесс формирования классификации типовых задач, для решения которых используется САЕ пакет, и построения дерева проблемно-ориентированных оболочек CAEBeans над САЕ пакетом на основе этой классификации.

При постановке задачи посредством проблемной оболочки CAEBean пользователю предоставляется возможность оперировать терминами той проблемной области, в рамках которой проводится решение задачи. Каждая конкретная задача может быть описана некоторым индивидуальным набором значений входных параметров, необходимых для реализации алгоритма решения данного класса задач. Область допустимых значений каждого входного параметра может быть строго фиксирована определенным множеством значений, задаваться неявно в зависимости от значений других входных параметров или быть определена как-либо еще [5]. В качестве примера таких параметров можно привести: температуру жидкости, протекающей в системе труб при моделировании трубопровода [4]; шаг и профиль резьбы при моделировании резьбового соединения труб [6]; скорость поступательного движения и скорость вращения трубы при моделировании процесса закалки [7].

Корневой элемент дерева проблемных оболочек представляется проблемной оболочкой, соответствующей наиболее обобщенному описанию определенного класса решаемых задач. Процедуру формирования дерева проблемно-ориентированных оболочек можно описать как движение от корневого элемента к листьям. Технология CAEBeans позволяет дочерним проблемно-ориентированным оболочкам конкретизировать классы задач, решаемые их ро-

дательской оболочкой, путем выделения и инкапсуляции групп параметров, значения которых фиксируются на данном уровне абстракции.

Первоначально для решения конкретной инженерной задачи, предлагается использовать проблемную оболочку с листовой вершины соответствующего дерева проблемных оболочек. Данная оболочка имеет максимально простой пользовательский интерфейс и позволяет пользователю указать значения ограниченного набора возможных параметров задачи. Однако область применения данной оболочки значительно ограничена и допускает решение узкого класса задач. В случае, если пользователю необходимо охватить более широкий класс задач, он осуществляет переход к родительской оболочке, тем самым переходя на более высокий уровень абстракции. Данная оболочка позволяет пользователю указать большее количество параметров и расширяет класс задач, который может быть решен посредством проблемной оболочки предыдущего уровня абстракции. Таким образом, пользователь может прийти до корневого CAEBean, который предлагает максимальные возможности по решению данного класса задач.

1.2. Состав системы CAEBeans

Системой CAEBeans будем называть совокупность программных средств, данных и аппаратных ресурсов, ориентированных на поддержку технологии CAEBeans. Программные средства, входящие в систему CAEBeans, должны обеспечить техническую поддержку процессов создания, модификации и использования оболочек CAEBeans. Возможно выделить следующие основные программные продукты, входящие в состав системы CAEBeans.

- **CAEBeans Constructor** – интегрированная среда разработки и управления оболочками CAEBeans. CAEBeans Constructor представляет собой систему разработки оболочек CAEBeans всех слоев, формирования деревьев проблемных оболочек CAEBeans,.
- **Брокер ресурсов** – автоматизированная система регистрации, анализа и предоставления ресурсов распределенной вычислительной среды. Подробное описание функциональных возможностей и структуре брокера ресурсов приведено в п. 3.3.
- **База данных CAEBeans** – система хранения и обработки оболочек CAEBeans. Поддерживает хранение деревьев проблемных оболочек, готовых системных, компонентных и потоковых оболочек.

Набор данных, входящих в состав системы CAEBeans, включает в себя: отдельные экземпляры готовых оболочек CAEBeans, предназначенные для решения типовых инженерных задач; шаблоны создания оболочек CAEBeans различных слоев; наборы знаний о CAE-пакетах и поддерживаемых ими интерфейсах автоматизации постановки заданий. В набор аппаратных ресурсов входят все рабочие станции, сервера и суперкомпьютеры, обеспечивающие аппаратную поддержку системы CAEBeans.

1.3. Основные требования, предъявляемые к программной поддержке технологии CAEBeans

Для обеспечения поддержки CAE-пакетов в грид-средах предлагается разработать систему, удовлетворяющую следующим требованиям.

1. **Адаптируемость к конкретному инженерному пакету.** Разрабатываемая система не должна базироваться на функциональных возможностях конкретного инженерного пакета или набора компонентов. Система должна обеспечивать возможность интеграции различных компонентов различных CAE-пакетов в процессе решения конкретных задач и обеспечивать адаптируемость к конкретному инженерному пакету.

2. **Адаптируемость к конкретной проблемной области.** Разрабатываемая система не должна быть узкоспециализированной в конкретной области компьютерного моделирования. Должны существовать возможности адаптации данной системы как для задач из определенной проблемной области, так и для таких задач, решение которых требует применения знаний из нескольких проблемных областей (мультифизических задач).

3. **Независимость от программно-аппаратных платформ.** На сегодняшний день существует широкий спектр операционных систем, на которых могут функционировать служебные и/или пользовательские компоненты инженерных систем. Разрабатываемая технология должна иметь возможность функционировать на большинстве таких операционных систем и архитектур.

4. **Поддержка сервисно-ориентированного подхода в грид.** Необходимо обеспечить возможность предоставления удаленного доступа к ресурсам различных инженерных пакетов, а также обеспечить возможность организации автоматического взаимодействия этих ресурсов в процессе решения задач. Это влечет за собой требование поддержки современных стандартов распределенных вычислительных систем и сервисно-ориентированного подхода предоставления ресурсов.

2. Архитектура системы CAEBeans

2.1. Слои системы CAEBeans

Архитектура системы CAEBeans формируется из четырех слоев: концептуального, логического, физического и системного. Обобщенная схема слоев системы CAEBeans приведена на рис. 1.

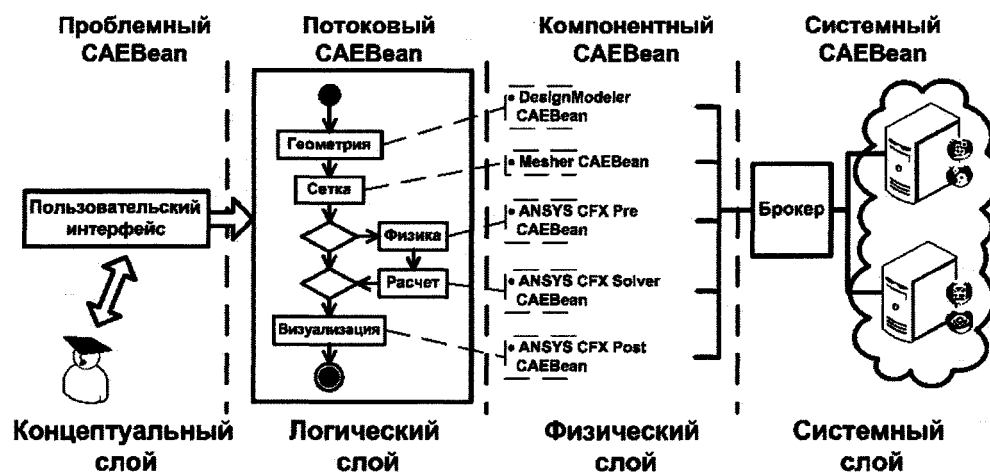


Рис. 1. Обобщенная схема слоев системы CAEBeans

2.2. Концептуальный слой CAEBeans

Концептуальный слой системы формируется на основе оболочек CAEBeans, которые мы будем называть проблемными. Пользовательский интерфейс, предоставляемый *проблемным CAEBean*, является основным средством взаимодействия пользователя с системой CAEBeans. Посредством пользовательского интерфейса *проблемного CAEBean*, ориентированного на решение определенного класса инженерных задач, пользователь может произвести постановку инженерной задачи; проследить за ходом решения поставленной задачи; получить требуемые результаты решения. Внешний вид пользовательского интерфейса *про-*

блемного CAEBean может значительно варьироваться в зависимости от класса задач, решаемых данной оболочкой.

Процесс постановки задачи посредством *проблемного CAEBean* сводится к указанию тем или иным образом значений входных параметров, характеризующих задачи соответствующего класса. Таким образом обеспечивается проблемно-ориентированная постановка задачи, независимая от функциональных особенностей программных продуктов, на основе которых будет производиться ее решение. *Проблемный CAEBean* скрывает от пользователя «лишнюю» функциональность САЕ-пакета, распределенный характер вычислительной среды, структуру аппаратных, программных и лицензионных ресурсов.

Также технология CAEBeans предусматривает возможность формирования дерева *проблемных CAEBeans*, которая подробно рассмотрена в п. 1.1. Это позволяет организовать подстройку *проблемных CAEBeans* под возможную конкретизацию базовой общей задачи в соответствии с требованиями конечных пользователей.

При постановке пользователем задачи на решение, соответствующий *проблемный CAEBean* формирует полный дескриптор задачи, представляющий собой множество пар «параметр – значение». Данные параметры могут иметь определенную семантику как с точки зрения проблемной области задачи, так и с точки зрения особенностей процесса моделирования и анализа. При формировании полного дескриптора задачи не делается различий между тем, указывал ли пользователь значение данного параметра явно, или же оно было тем или иным способом вычислено соответствующим *проблемным CAEBean*. При этом набор параметров, входящих в полный дескриптор задачи, одинаков для всех *проблемных CAEBean*, входящих в одно дерево проблемных оболочек. Сформированный полный дескриптор задачи передается в *поточный CAEBean*, представляющий логический слой системы CAEBeans.

2.3. Логический слой CAEBeans

Логический слой системы CAEBeans представлен так называемым *поточным CAEBean*, реализующим логический план решения определенного класса задач компьютерного моделирования. Логический план решения задачи представляет собой ориентированный граф, в вершинах которого могут находиться блоки двух типов: подзадачи, выполняемые отдельными базовыми компонентами (сущность базового компонента подробно рассматривается в п.3.1), и специальные служебные операции управления потоком решения задачи (такие, как ветвление, распараллеливание и т.п.), что обеспечивает возможности гибкого исполнения логического плана (рис. 2).

При получении полного дескриптора задачи, *поточный CAEBean* распределяет массив переданных параметров по блокам операций логического плана решения задачи (пример распределения набора параметров приведен на рис. 2). Если параметр управляет потоком решения задачи, его значение передается в один из блоков управления, где производится его анализ и принимается решение о дальнейшем ходе потока решения задачи. Иначе, значение параметра передается в соответствующий блок подзадачи, где производится его дальнейшая интерпретация.

Решение каждой подзадачи инкапсулируется отдельным *компонентным CAEBean*, представляющим физический слой системы CAEBeans. В процессе решения задачи *поточный CAEBean*, «руководствуясь» планом решения задачи, передает параметры подзадачи соответствующим *компонентным CAEBean* и получает от них результаты решения соответствующих подзадач.

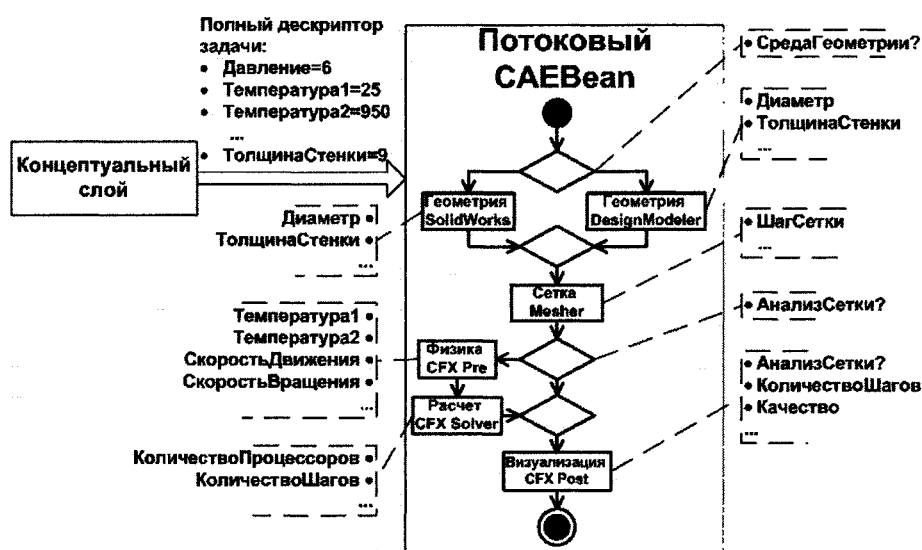


Рис. 2. Пример логического плана решения задачи

2.4. Физический слой CAEBeans

Компонентные оболочки CAEBean, составляющие физический слой системы CAEBeans, инкапсулируют процесс постановки и решения конкретной подзадачи компьютерного моделирования посредством определенного базового компонента. Каждый отдельный базовый компонент обладает уникальным интерфейсом автоматизации постановки и решения заданий. В качестве примера таких интерфейсов можно привести: лог-файлы и макросы, поддерживаемые САЕ-пакетом ANSYS Mechanical; файлы сценариев на языке Python, поддерживаемые САЕ-пакетом Abaqus; текстовый файл CCL (Command Language File – Файл языка команд) поддерживаемый решателем ANSYS CFX-Solver. В связи с тем, что каждый конкретный базовый компонент обладает уникальным интерфейсом автоматизации постановки задач, невозможно создать универсальную систему, которая позволяла бы сформировать постановку задания для любого компонента по данному проблемно-ориентированному описанию задачи.

Компонентный CAEBean обладает информацией о методах взаимодействия с конкретным базовым компонентом. В зависимости от интерфейса автоматизированной постановки задач, поддерживаемого базовым компонентом, разрабатывается специальный генератор постановки данной подзадачи. В процессе решения задачи, компонентному CAEBean из потокового CAEBean передается набор значений параметров, на основе которого генерируется файл постановки задания.

Другой важной функцией компонентных CAEBean является инкапсуляция взаимодействия с брокером ресурсов и вычислительной грид-средой (процесс взаимодействия компонентного CAEBean с грид-средой подробно рассмотрен в пп. 3.3.1, 3.3.2). Компонентный CAEBean соединяется с системным CAEBean, представляющим конкретный базовый компонент в грид-среде, и передает ему файл постановки задания.

Внедрение физического слоя в систему CAEBeans позволяет обеспечить универсальность применения совместимых базовых компонентов для решения одинаковых подзадач. Универсальность подхода обеспечивается тем, что проблемно-ориентированная постановка подзадачи не зависит от интерфейса конечного базового компонента. Таким образом, возможность применения того или иного базового компонента для решения определенной подзадачи зависит только от возможности реализации компонентного CAEBean, транслирующего постановку подзадачи в соответствующую компонентно-ориентированную форму.

3. Системный слой CAEBeans

3.1. Базовый компонент

С точки зрения системной реализации, большинство современных CAE-систем может быть представлено как набор относительно функционально-независимых компонентов, каждый из которых отвечает за собственную ступень технологического цикла решения задачи компьютерного моделирования. Можно привести следующие типичные блоки, реализуемые в различных современных инженерных системах: построение геометрии; формирование сетки; определение физики; численное моделирование поставленной задачи; визуализация и анализ результатов решения. В качестве примеров таких компонентов, можно привести генератор сеток ANSYS CFX Mecher, решатель LSDyna или пост-процессор ANSYS CFX Post.

С точки зрения CAEBeans, такие компоненты (*базовые компоненты*) – это изолированные приложения, которые представляются как «черный ящик», реализующий некий специфический интерфейс [8]. Для каждого *базового компонента* существует уникальный способ указать набор входных параметров, запустить процесс решения подзадачи, получить промежуточные результаты решения или информацию о ходе решения, получить конечный результат.

3.2. Системный CAEBean

Системным CAEBean называется оболочка, инкапсулирующая функциональные возможности конкретного базового компонента и обеспечивающая сервисно-ориентированный подход к постановке задач и получению результатов (рис. 3). В общем случае, *базовые компоненты* не поддерживают возможности удаленной постановки задачи и предоставления результата удаленному пользователю. *Системный CAEBean*, реализованный для конкретного базового компонента, является его «представителем» в вычислительной грид-среде. Реализация соответствующего *системного CAEBean* полностью зависит от поддерживаемых данным базовым компонентом методов автоматизации постановки задач и получения результата.



Рис. 3. Схема представления системного CAEBean

Исследования, проведенные в работах [8,9], показали, что практически во всех случаях современными инженерными системами поддерживается широкий спектр различных методов управления процессом постановки и решения задач.

3.3. Структура и функциональные особенности системного слоя CAEBeans

3.3.1. Структура системного слоя CAEBeans

Для решения подзадачи, поставленной на физическом слое *компонентным CAEBean*, необходимо обеспечить ее проекцию на набор ресурсов, предоставляемых системным слоем.

Для организации этого процесса используется *брокер ресурсов* – автоматизированная система регистрации, анализа и предоставления ресурсов распределенной вычислительной среды (рис. 4). Брокер представляет собой промежуточное программное обеспечение, обеспечивающее оптимальный выбор и виртуализацию доступа к наборам ресурсов в гетерогенной вычислительной среде [2].

Возможно выделить следующие основные задачи *брокера ресурсов*: обработка каталога ресурсов грид-среды; анализ запросов на предоставление ресурсов, поступающих от *компонентных CAEBean*; сбор и предоставление информации об актуальном состоянии грид-среды.

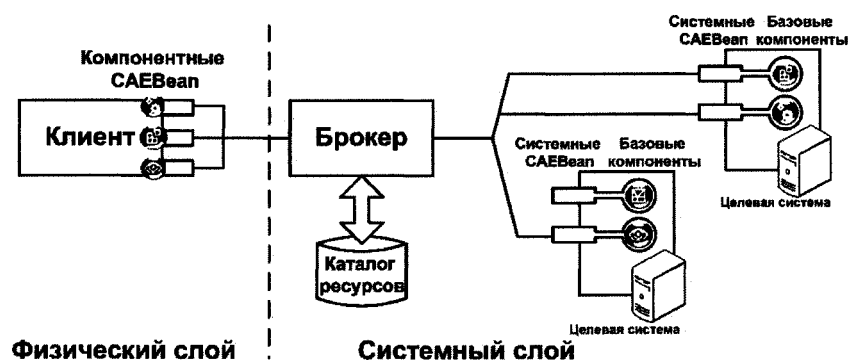


Рис. 4. Архитектура системного слоя

В процессе формирования гетерогенной вычислительной среды, на каждом отдельном вычислителе создается набор целевых систем. Каждая целевая система характеризуется определенным набором свойств, описывающих физические и логические ресурсы, предоставляемые целевой системой. Физические ресурсы определяются такими параметрами, как операционная система, объем доступной оперативной памяти, объем доступного дискового пространства, тип и количество процессоров и т.п. Логические ресурсы определяются набором и характеристиками предоставляемых *системных CAEBean*. Это могут быть такие характеристики как тип CAEBean, версия, параметры запуска, набор доступных модулей и т.п. Данная информация передается в каталог ресурсов во время инициализации целевой системы и автоматически обновляется при изменении состояния вычислителя.

Компонентные CAEBean обладают определенной информацией о том, какие значения характеристик вычислительной среды позволяют оптимизировать процесс решения соответствующей подзадачи. Данная информация может формироваться из различных источников.

1. В ходе постановки задачи посредством *проблемного CAEBean* пользователь может указать определенные критерии, которым должен удовлетворять процесс решения задачи (например, максимально допустимое время решения задачи). Данные требования транслируются *компонентному CAEBean*, который обладает достаточными знаниями о ходе решения соответствующей подзадачи и может перевести данные требования в запрос конкретных физических характеристик вычислительной среды.

2. Разработчиком *компонентного CAEBean* может быть заложен определенный набор требований (пожеланий) к вычислительной среде, обеспечение которых позволит оптимизировать процесс решения данного типа подзадач (минимальный объем оперативной памяти, ограничение максимального количества процессоров и т.п.)

Таким образом, *компонентный CAEBean* формирует запрос к брокеру ресурсов, в котором описывает характеристики вычислительной среды, необходимой для решения данной подзадачи.

3.3.2. Алгоритм предоставления ресурсов грид-среды

В процедуре проекции подзадачи с физического на системный слой брокер ресурсов выступает в качестве промежуточного звена, обеспечивающего поиск и предоставление ресурсов, оптимальных для решения данной подзадачи (рис. 5).

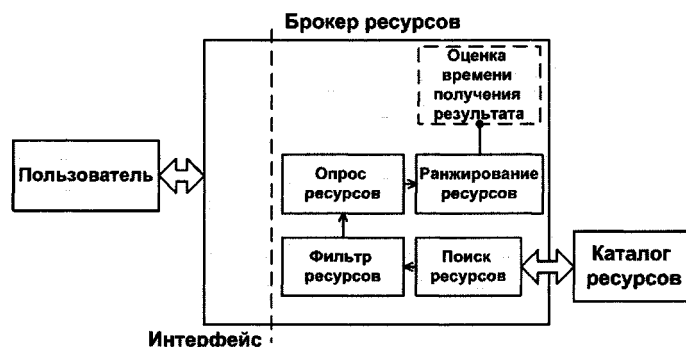


Рис. 5. Внутренняя структура брокера ресурсов

Рассмотрим более подробно алгоритм выделения и использования ресурсов грид-среды с участием брокера ресурсов.

1. *Компонентный CAEBeap* производит запрос к *брокеру* на предоставление ресурса.
2. *Брокер* производит запрос каталогу ресурсов, формируя список узлов, характеристики которых удовлетворяют заявленным обязательным требованиям и правам пользователя, поставившего задачу.
3. *Брокер* производит опрос узлов из выбранного списка, для уточнения их доступности и определения значений их динамических характеристик (состояние очереди задач, доступный объем оперативной памяти и т.п.).
4. *Брокер* оценивает общее время получения результата (TTD – Total Time to Delivery) [10] для каждого из анализируемых узлов и выбирает узел, с наименьшим значением данного параметра.
5. *Брокер* передает адрес предоставляемого ресурса *компонентному CAEBeap*.
6. *Компонентный CAEBeap* производит захват предоставленного ресурса, а *брокер* маркирует данный ресурс как занятый.
7. Дальнейший процесс постановки, решения и получения результатов происходит напрямую, между *компонентным CAEBeap* и предоставленным ресурсом без участия *брокера*.
8. После получения результатов вычисления *компонентный CAEBeap* производит освобождение ресурса и извещение *брокера* об его освобождении.

3.3.3. Оценка общего времени получения результата

Одной из наиболее важных задач брокера ресурсов является оценка общего времени получения результата (TTD – Total Time to Delivery) при постановке задачи пользователем конкретному вычислительному ресурсу. В дальнейшем, на основе этой оценки, определяется ресурс, обеспечивающий оптимальное время исполнения поставленной задачи. При оценке TTD необходимо учитывать, что от постановки до получения конечных результатов задача, как правило, проходит следующие стадии решения [10].

1. **Постановка.** Производится передача исходных файлов и сценариев решения задачи выбранному ресурсу.
2. **Ожидание.** Это время, необходимое для запуска задачи. Состоит из времени ожидания задачи в очереди задач выбранного ресурса, времени **постановки** задачи и т.п.

3. **Исполнение.** Проведение вычислений на выбранном ресурсе.

4. **Завершение.** Передача файлов результатов в требуемое хранилище.

Необходимо отметить, что время ожидания обычно определяется как максимум из времени на постановку задачи и времени на все остальные операции, необходимые для того, чтобы запустить задачу на исполнение. Это связано с тем, что процесс передачи исходных файлов может происходить независимо от всех остальных процессов подготовки задачи. Таким образом, TTD может быть оценено как сумма оценок времени на проведение операций ожидания, исполнения и завершения.

При формировании данных оценок, брокер использует информацию о подзадаче, передаваемую соответствующим *компонентным CAEBean*. В процессе запроса вычислительных ресурсов *компонентный CAEBean* может предоставить брокеру такую информацию, как размер исходных данных; вычислительная сложность рассматриваемой подзадачи; масштабируемость подзадачи и т.п. На основе этих данных и информации о текущих характеристиках грид-среды брокер ресурсов может произвести оценку общего времени получения результата.

4. Реализация оболочек CAEBeans

В рамках исследования возможностей внешнего управления процессом постановки и решения задач в современных CAE-пакетах были реализованы прототипы проблемно-ориентированных оболочек над пакетами ANSYS Mechanical, ANSYS CFX, ABAQUS, DEFORM [9].

Важной частью работы является применение разрабатываемой технологии для решения конкретных производственных задач. В рамках контракта с Челябинским Трубопрокатным заводом созданы проблемно-ориентированные оболочки для решения задач, возникающих в процессе производства и эксплуатации изделий предприятия. По заказу предприятия решены задачи: «Разработка проблемно-ориентированных оболочек для моделирования воздействия деформирующих нагрузок на резьбовые соединения обсадных и насосно-компрессорных труб» [6] и «Разработка проблемно-ориентированных оболочек для моделирования овализации труб при закалке» [7]. Разработанные оболочки инкапсулируют процесс решения соответствующих задач в рамках функциональных возможностей пакетов ANSYS Mechanical и DEFORM. Данные оболочки позволяют производить моделирование поставленных задач средствами высокопроизводительного комплекса вычислительного центра «Infinity» Южно-Уральского государственного университета.

Заключение

В статье представлены основные результаты разработки технологии CAEBeans, ориентированной на сервисно-ориентированное предоставление ресурсов CAE-пакетов в рамках гетерогенных высокопроизводительных сред. Архитектура представленной системы может быть разделена на четыре слоя: концептуальный; логический; физический и системный. *Проблемные оболочки CAEBean*, формирующие концептуальный слой системы CAEBeans предоставляют пользовательский интерфейс решения конкретного класса инженерных задач. Логический слой системы формируется *потокowym CAEBean*, реализующим логический план решения конкретного класса инженерных задач. *Компонентные оболочки CAEBean*, составляющие физический слой системы CAEBeans, отвечают за формирование заданий базовым компонентам и взаимодействие с вычислительной грид-средой. Системный слой CAEBeans представлен вычислительной грид-средой, ресурсы которой используются для решения инженерных задач. Процесс взаимодействия пользовательской системы с грид-

ресурсами происходит посредством брокера ресурсов.

Можно выделить следующие основные этапы дальнейшего развития работы: программная реализация иерархии оболочек CAEBeans, спецификация стандартов разработки системы; разработка программных средств, обеспечивающих поддержку технологии CAEBeans: CAEBeans Constructor, брокер ресурсов грид-среды. В рамках апробации технологии CAEBeans, необходимо продолжать разработку оболочек CAEBeans для решения реальных задач инженерного анализа. В качестве аппаратной поддержки технологии CAEBeans предполагается задействовать вычислительные ресурсы суперкомпьютерных систем Южно-Уральского государственного университета: «Infinity» и «СКИФ Урал».

Работа проводилась при финансовой поддержке Федерального агентства по науке и инновациям (грант 2007-4-1.4-20-01-026), программы СКИФ-ГРИД (грант СГ-1/07) и Фонда содействия развитию малых форм предприятий в научно-технической сфере (грант 7434).

Литература

1. Бегунов, А.А. Применение результатов моделирования для оптимизации и управления технологическими процессами / А.А. Бегунов // Параллельные вычислительные технологии: тр. Междунар. науч. конф. (28 янв. – 1 февр. 2008 г., г. Санкт-Петербург). – СПб., 2008. – С. 31 – 38.
2. Foster, I. The Grid 2: Blueprint for a New Computing Infrastructure. Second edition / I. Foster, C. Kesselman. – San Francisco: Morgan Kaufmann, 2003.
3. Foster, I. The Anatomy of the Grid: Enabling Scalable Virtual Organizations / I. Foster, C. Kesselman, S. Tuecke // International J. of Supercomputer Applications and High Performance Computing. – Massachusetts, 2001. – Т. 15, № 3. – С. 200 – 222.
4. Радченко, Г.И. CAEBeans: иерархические системы структурированных проблемно-ориентированных оболочек над инженерными пакетами / Г.И. Радченко, Л.Б. Соколинский // Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ: тр. Всерос. науч. конф. (24-29 сентября 2007 г., г. Новороссийск). – Новороссийск, 2007. – С. 54 – 57.
5. Радченко, Г.И. BeanShells: интеграция CAE-пакетов в GPE / Г.И. Радченко, Л.Б. Соколинский, И.С. Кутепов // Параллельные вычислительные технологии (ПаВТ'2007): тр. Междунар. науч. конф. (Челябинск, 29 янв. – 2 февр. 2007 г.). – Челябинск, 2007. – Т.2. – С. 15.
6. Разработка проблемно-ориентированной Grid-оболочки для моделирования резьбовых соединений труб для нефтяных скважин в распределенных вычислительных средах / В.В. Лёушкин, Л.Б. Соколинский, К.А. Чайко, В.В. Юрков // Параллельные вычислительные технологии: тр. междунар. науч. конф. (28 янв. – 1 февр. 2008 г., г. Санкт-Петербург). – СПб., 2008. – С. 534.
7. Дорохов, В.А. Разработка проблемно-ориентированной GRID-оболочки для решения задачи оваллизации труб при закалке / В.А. Дорохов, А.Н. Маковецкий, Л.Б. Соколинский // Параллельные вычислительные технологии: тр. Междунар. науч. конф. (28 янв. – 1 февр. 2008 г., г. Санкт-Петербург). – СПб., 2008. – С. 520.
8. Радченко, Г.И. Разработка компонентно-ориентированных CAEBean-оболочек для пакета ANSYS CFX / Г.И. Радченко, Л.Б. Соколинский, А.В. Шамакина // Параллельные вычислительные технологии: тр. Междунар. науч. конф. (28 янв. – 1 февр. 2008 г., г. Санкт-Петербург). – СПб., 2008. – С. 438 – 443.

9. Методы организации программных интерфейсов к инженерным пакетам в среде GPE / Р.С. Насибулина, К.В. Репина, А.В. Шамакина и др. // Параллельные вычислительные технологии: тр. Междунар. науч. конф. (28 янв. – 1 февр. 2008 г., г. Санкт-Петербург). – СПб., 2008. – С. 537.
10. Elmroth, E. A Grid resource broker supporting advance reservations and benchmark-based resource selection / E. Elmroth, J. Tordsson // Stateof-the-art in Scientific Computing, Springer-Verlag, LNCS. – 2006. – № 3732. – С. 1061 – 1070.

Кафедра системного программирования,
Южно-Уральский государственный университет
gleb.radchenko@gmail.com

Поступила в редакцию 17 марта 2008 г.