

Васильев А.В., Сарданашвили С.А.

(РГУ нефти и газа имени И.М. Губкина)

**АРХИТЕКТУРНЫЕ РЕШЕНИЯ ИНТЕГРАЦИИ ПОДСИСТЕМЫ
МОДЕЛИРОВАНИЯ РЕЖИМОВ В ЕДИНОЕ ИНФОРМАЦИОННОЕ
ПРОСТРАНСТВО АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ ОАО «ГАЗПРОМ»**

Создание единого информационного и вычислительного пространства является на ближайшие годы стратегической задачей модернизации автоматизированной системы диспетчерского управления (АСДУ) ОАО «Газпром». Реализация данной задачи требует разработки принципиально новой архитектуры вычислительной среды решения режимно-технологических задач ДУ в виде распределённой иерархической системы, соответствующей многоуровневому диспетчерскому управлению. В данной статье рассматриваются соответствующие архитектурные решения построения подсистемы моделирования режимов систем газоснабжения программно-вычислительного комплекса ПВК «Веста», которые позволят интегрировать его в единое информационное и вычислительное пространство системы диспетчерского управления ЕСТ России, создаваемое в ОАО «Газпром».

Введение

В настоящее время ведётся разработка модернизированной автоматизированной системы диспетчерского управления (АСДУ) единой системы газоснабжения (ЕСТ) России. На первом этапе реализации модернизированная АСДУ ЕСТ создаётся как территориально-распределённая иерархическая двухуровневая система. Первому уровню соответствует верхний уровень иерархии диспетчерского управления (ДУ) – центральный производственный диспетчерский департамент (ЦПДД), а второму – производственно-диспетчерские службы (ПДС) эксплуатирующих организаций ОАО «Газпром». Ядром модернизированной АСДУ является единое информационное пространство (ЕИП), которое должно стать единым источником данных для принятия управляющих решений на всех иерархических уровнях ДУ ЕСТ. Для ЕИП разрабатывается единая модель данных с использованием единой нормативно-справочной, оперативной диспетчерско-технологической, учётной режимной, отчётной балансовой информации, поддерживаемой в актуальном состоянии и используемой на всех уровнях диспетчерского управления. Повышается степень автоматизации процессов сбора, обработки, хранения и предоставления информации в ЕИП за счёт

интеграции с используемыми автоматизированными системами различного функционального назначения.

Одной из ключевых функциональных подсистем модернизированной АСДУ ЕСГ является подсистема моделирования и поддержки принятия диспетчерских решений, позволяющая обрабатывать огромные объёмы информации ЕИП на основе моделирования технологических процессов и режимов работы объектов систем газоснабжения, анализа оперативных и ретроспективных данных, формирования сводок и отчётов, консолидированных как по дочерним обществам, так и по ЕСГ в целом. На сегодняшний день эта функциональная подсистема АСДУ ЕСГ представлена разнородными, слабо приспособленными к интеграции с ЕИП, программно-вычислительными комплексами систем поддержки принятия диспетчерских решений (ПВК СППДР), используемыми при принятии решений на всех уровнях ДУ[1].

Неотъемлемой частью каждого ПВК СППДР является подсистема моделирования, или вычислительная подсистема, вычислительный модуль. Архитектура построения вычислительной подсистемы, организация её взаимодействия с программно-вычислительным комплексом (ПВК) во многом определяет потенциал развития, обновления комплекса в целом, удобство его сопровождения, степень эффективности использования вычислительных ресурсов, а значит, масштаб и сложность решаемых задач.

Применяемые в настоящее время архитектуры построения подсистем моделирования имеют ряд существенных недостатков: ориентация на последовательные вычисления, монолитная, или монолитно-блочная организация, использование закрытых протоколов, ориентация на однопользовательский режим работы [2].

Ориентация на последовательные вычисления означает невозможность полного использования потенциала современной вычислительной техники, поскольку все современные аппаратные архитектуры ориентированы на использование параллелизма. Недостаток производительности существенно затрудняет решение наиболее вычислительно сложных и актуальных задач.

Монолитность вычислительного модуля влечёт за собой сложность его развития, обновления, сопровождения, не позволяет без существенной переработки перейти к использованию параллельных вычислений.

Использование закрытых, нестандартных протоколов для взаимодействия ПВК с расчётным модулем затрудняет создание «лёгких», «тонких» клиентов, способных использовать вычислительную подсистему для проведения необходимых расчётов.

Использование закрытых форматов представления решений затрудняет интеграцию с другими системами, сравнение результатов работы с другими вычислительными комплексами.

Ориентация на однопользовательский режим работы не соответствует принципам создания модернизированной АСДУ ЕСТ, распределённой многоуровневой системы с иерархией, соответствующей иерархии диспетчерского управления и может стать затрудняющим обстоятельством для интеграции с ЕИП.

Вычислительные подсистемы комплексов являются закрытыми, затруднён обмен данными между ПВК, а тем более использование одним ПВК вычислительных методов, реализованных в подсистеме моделирования другого ПВК. Это влечёт за собой дублирование реализации сходных вычислительных задач в различных ПВК, увеличивает трудоёмкость разработки, затрудняет согласование результатов их работы.

Таким образом, актуальной задачей является разработка программной архитектуры вычислительной подсистемы ПВК СППДР, ориентированной на:

- 1) эффективную организацию вычислительного процесса с использованием параллельных вычислений,
- 2) модульность, расширяемость, гибкость, развиваемость,
- 3) открытость, использование стандартных протоколов сетевого взаимодействия, форматов передачи данных,
- 4) соответствие распределённой многоуровневой иерархии модернизированной АСДУ,
- 5) возможность предоставления единых вычислительных сервисов для ПВК всех уровней по запросу.

На основе перечисленных принципов, выработанных в результате анализа применения различных программно-вычислительных комплексов ПВК (Астра, САМПАГ, Gannezi, Simone) решения задач моделирования режимов трубопроводного транспорта газа, может быть построена современная, эффективная, приспособленная к развитию, подсистема моделирования, выходящая за рамки вычислительного модуля для конкретного ПВК. Такая подсистема может по открытым протоколам предоставлять единые вычислительные сервисы для различных ПВК всех уровней и

послужить основой разработки необходимого дополнения ЕИП – Единого Вычислительного Пространства.

В данной статье рассматривается программная архитектура, построенная по перечисленным принципам, которые реализуются в настоящее время в ПВК «Веста», разрабатываемом коллективом сотрудников института инновационных образовательных проектов и проблем управления РГУ нефти и газа имени И.М. Губкина, и внедряемом в различных газотранспортных и газодобывающих эксплуатирующих организациях ОАО «Газпром» в качестве компьютерного тренажера диспетчерских служб.

Эффективная организация вычислительного процесса с использованием различных типов вычислительных ресурсов

На сегодняшний день используются существенно различающиеся типы вычислительных ресурсов, среди них выделяются три наиболее распространённых [3]:

- 1) многоядерные центральные процессоры общего назначения,
- 2) массивно-параллельные сопроцессоры,
- 3) распределённые системы.

Каждому типу вычислительных ресурсов соответствует множество алгоритмов, которое может быть эффективно выполнено с его использованием.

Применительно к рассматриваемой задаче моделирования режимов систем газоснабжения можно утверждать следующее. Центральный процессор может быть использован для организации функционирования ядра подсистемы, общего управления вычислительным процессом, моделирования сложных, составных технологических объектов, таких как компрессорный цех, компрессорная станция, газотранспортная (ГТС) или трубопроводная система (ТС).

Массивно-параллельные сопроцессоры могут быть эффективно использованы для:

- выполнения оптимизированных библиотек численных методов, например, решения систем уравнений с разреженными матрицами высокой размерности,
- SIMD-моделирования (Single Instruction – Multiple Data) многочисленных простых технологических объектов, таких как трубопроводы, краны и другие локальные объекты.

Распределённые системы могут быть использованы для решения самых сложных вычислительных задач, обладающих высокой размерностью, вычислительной сложностью, например, для:

- численного моделирования стационарных и нестационарных режимов региональных систем газоснабжения с высокой степенью детализации состава технологических объектов,
- решения задач идентификации эмпирических параметров моделей и адаптации расчетных режимов к фактическим режимам on-line;
- задач квазиоптимального планирования стационарных режимов,
- задач прогнозирования нестационарных режимов.

Анализ применимости различных типов вычислительных ресурсов к задаче поиска режима системы газоснабжения

Основой численного моделирования стационарных и нестационарных режимов разветвленных, закольцованных газотранспортных и распределительных систем являются алгоритмы балансирования потоков на математическом графе, описывающем соответствующие технологические системы газоснабжения. При этом с точки зрения программной архитектуры, не имеет значения то, какие конкретные алгоритмы и вычислительные методы применяются в качестве численной модели. Поэтому для иллюстрации возможностей предложенной ниже архитектуры ПВК, приведен пример разработанного распределённого модуля, выполняющего численное балансирование потоков газа на расчётном графе газотранспортной системы и интегрированного с интерфейсом ПВК «Веста».

В самом общем виде задача моделирования стационарных и нестационарных режимов системы газоснабжения заключается в следующем.

Пусть дуги графа рассматриваемой системы газоснабжения представляют моделируемые локальные объекты: трубы, компрессорные цеха, байпасные краны, краны-регуляторы, краны редуцирования давления и так далее. Узлы графа являются точками сопряжения входов-выходов моделируемых объектов, расчетных или заданных краевых режимных параметров моделей газовых потоков этих объектов. Вычислительная модель объекта должна численно связывать входные и выходные параметры режима объекта. Задача заключается в определении множества значений давлений и температур газа в узлах графа, обеспечивающих баланс входящих и выходящих потоков в каждом узле, при соблюдении всех технологических ограничений.

Модель нестационарных режимов отличается от модели стационарных режимов тем, что балансирование потоков газа в узлах графа осуществляется на множестве

временных слоев и тем, что модель режима дуги графа в общем виде представлена минимум на двух слоях координаты времени.

Решение данной задачи может быть реализовано методом узловых давлений [4], который, представляет собой итерационную процедуру, включающую в себя моделирование множества объектов, численное вычисление производных, формирование и решение разреженной СЛАУ (Системы Линейных Алгебраических Уравнений).

Для выполнения алгоритма могут быть использованы следующие ресурсы:

- 1) ядро центрального процессора для общей организации итерационного вычислительного процесса,
- 2) сопроцессор для быстрого решения разреженных СЛАУ,
- 3) сопроцессор для моделирования множества однотипных простых объектов, например, трубопроводов,
- 4) несколько ядер центрального процессора для моделирования множества более сложных объектов, таких как компрессорные цеха.

Таким образом, для решения описанной задачи может быть эффективно использовано несколько вычислительных ресурсов разного типа, типы и состав которых могут зависеть от состава, типов и количества технологических объектов в схеме, становящихся известными только во время выполнения алгоритма. Более подробно вопрос использовании гетерогенных параллельных вычислений в методе узловых давлений рассматривается в [5].

Клиент-серверная архитектура среды моделирования

Для того чтобы обеспечить доступ различных ПВК, функционирующих в ЕИП, к вычислительным сервисам, предоставляемым средой моделирования, хорошо подходит широко применяемый клиент-серверный подход. В данном случае клиенты – это ПВК, а сервер – среда моделирования, выполняющая следующие функции:

- 1) предоставление клиентам расчётных сервисов по определённым интерфейсам и заданным протоколам;
- 2) обеспечение возможности динамического выделения вычислительных ресурсов по запросу во время выполнения вычислительной задачи;
- 3) балансирование вычислительной нагрузки по ресурсам сервера с организацией выполнения задач на вычислительных ресурсах наиболее подходящего типа;
- 4) обеспечение монопольного доступа к ресурсам и возможности выполнения клиентом последовательности операций с сохранением состояния.

Для организации выполнения перечисленных функций сервером предлагается следующий подход.

Серверная среда моделирования, или «сервер» представляет собой вычислительный кластер, состоящий из нескольких взаимосвязанных «узлов», каждый из которых содержит вычислительные ресурсы различных типов (рис. 1). Для простоты будем рассматривать два типа вычислительных ресурсов – CPU и Tesla, как пример центрального процессора и сопроцессора, соответственно.

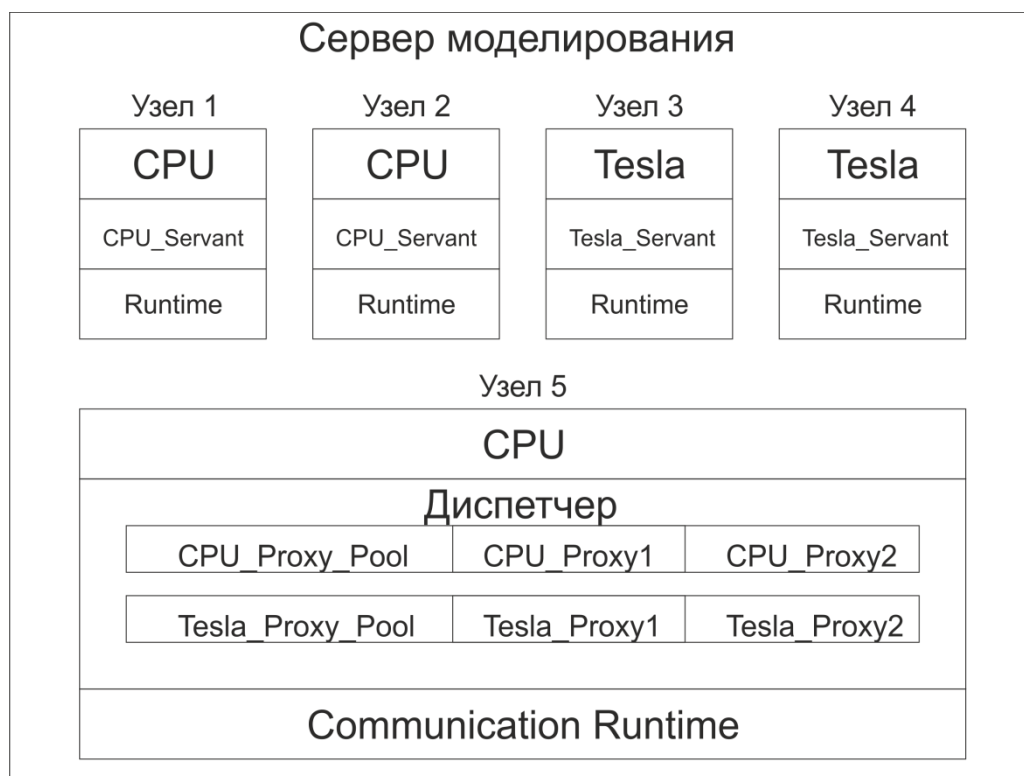


Рисунок 1. Схема сервера моделирования.

Каждому типу вычислительных ресурсов сопоставляется программный интерфейс, содержащий перечень функций, которые могут быть эффективно выполнены ресурсом данного типа. Далее создаётся класс, реализующий этот интерфейс, и один или несколько экземпляров данного класса, каждый из которых выполняется в процессе, запущенном на вычислительном ресурсе соответствующего типа. Объект класса, выполняющийся на типизированном вычислительном ресурсе на одном из узлов сервера, назовём «сёрвантом» (Servant) соответствующего типа.

Далее вводится понятие прокси-объекта на сёрвант. Прокси-объект («прокси») – это локальный представитель некоторого конкретного сёрванта, который может быть передан клиенту. Прокси реализует тот же интерфейс, что и сёрвант, дополненный

возможностью асинхронного вызова методов. При вызове методов прокси-объекта, сам он не выполняет вычислений, а передаёт сообщение сёрванту, который выполняет вычисления на сервере и возвращает прокси-объекту результат. Прокси выглядят для клиентов как локальные объекты, обеспечивающие семантику синхронного и асинхронного вызова удалённых процедур.

На одном из узлов сервера запускается «диспетчер» вычислительного сервера. Диспетчер поддерживает набор типизированных пулов, каждый из которых содержит прокси-объекты на ресурсы одного и того же типа. Когда на одном из узлов сервера запускается «сёрвант» некоторого типа, он сигнализирует о своём запуске диспетчеру и тот создаёт прокси-объект соответствующего типа, связывает его с этим «сёрвантом» и помещает в соответствующий пул.

Для выполнения некоторой операции клиент запрашивает у диспетчера прокси-объект соответствующего типа. Диспетчер изымает прокси из соответствующего пула и передаёт его клиенту. Клиент использует прокси-объект до тех пор, пока он ему необходим, затем «отпускает» его, вызывая служебный метод, и прокси возвращается в соответствующий пул диспетчера для дальнейшего использования другими клиентами.

Рассмотрим, как на основе предлагаемых архитектурных решений может быть организован вычислительный процесс, реализующий алгоритм поиска режима газотранспортной системы, не вдаваясь в особенности алгоритмической реализации каждого шага и процедуры в целом (рис.2).

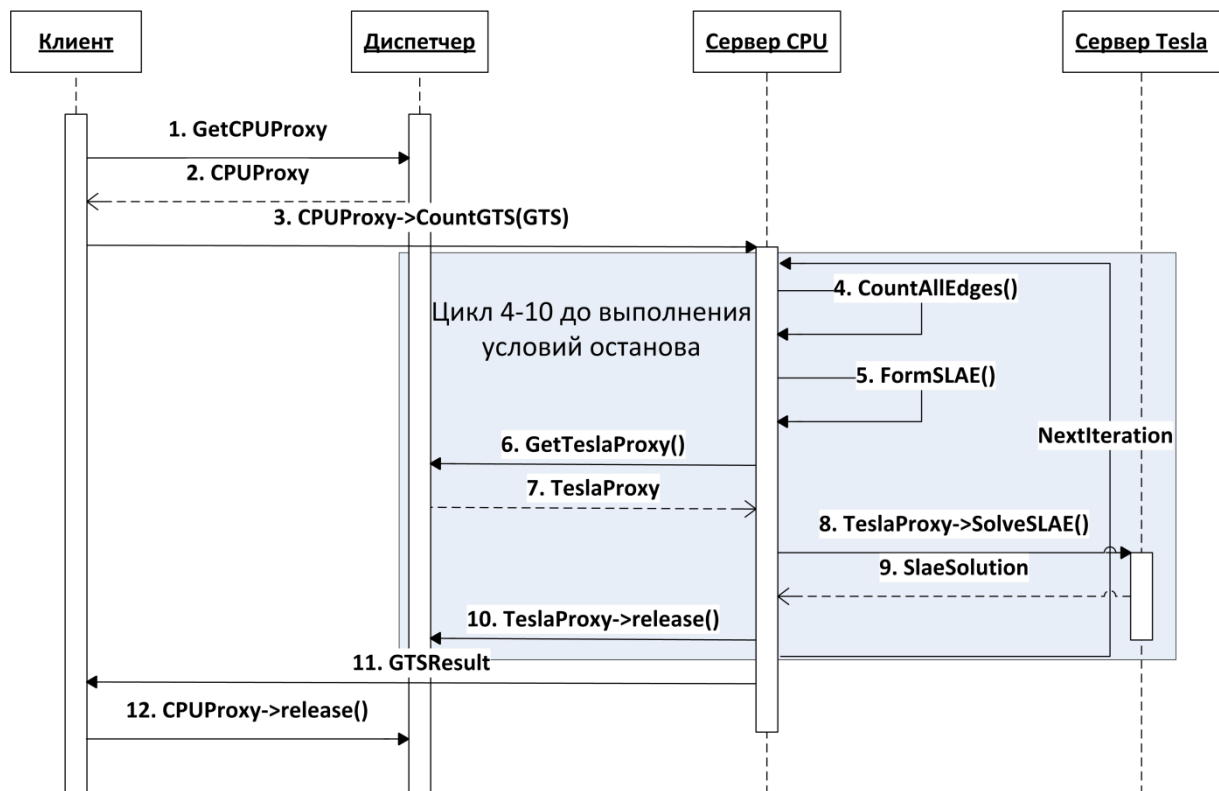


Рисунок 2. Взаимодействие компонент системы для решения задачи расчета параметров режима системы газоснабжения.

На сервере работают два сёрванта типа CPU и два сёрванта типа Tesla. Диспетчер управляет двумя пулами типизированных прокси, содержащих по два прокси типа «CPU» и «Tesla», связанные с соответствующими сёрвантами.

Последовательность действий, направленных на решение задачи, такова:

Шаги 1, 2. Клиент запрашивает у диспетчера CPUProxy. Диспетчер изымает Проху-объект из соответствующего пула и возвращает его клиенту.

Шаг 3. Клиент делает вызов CPUProxy->CountGTS(GTS), передавая исходные данные сёрванту для выполнения. Для клиента вызов выглядит как обычный локальный вызов.

Шаги 4, 5. Вызов передаётся сёрванту типа CPU через Communication runtime и он выполняет расчёт параметров режима всех дуг графа и формирует СЛАУ.

Шаги 6, 7. Для решения СЛАУ сёрвант CPU запрашивает у диспетчера прокси-объект TeslaProxy, связанный с сёрвантом «Tesla». Диспетчер изымает объект из пула «Tesla» и возвращает сёрванту CPU.

Шаг 8. Сёрвант CPU выполняет метод TelsProxy->SolveSLAE(SLAE).

Шаг 9. Сёрвант Tesla получает вызов от проху и выполняет решение СЛАУ с использованием оптимизированной библиотеки на CUDA и возвращает решение.

Шаг 10. Сёрвант CPU «отпускает» захваченный TeslaProху – теперь он возвращается в пул и может быть использован другими ресурсами.

Шаги 4-10 повторяются итерационно до нахождения искомого решения.

Шаг 11. Происходит возврат искомого решения клиенту.

Шаг 12. Клиент отпускает CPUProху и он возвращается в свой пул.

Приведённая упрощённая схема иллюстрирует принципы взаимодействия компонент системы для решения задачи и четыре важных особенности:

1. Компоненты системы могут выступать как в роли клиентов, так и серверов: в данном случае сервер CPU является сервером для клиента, и клиентом для сервера Tesla.

2. После получения прокси от диспетчера дальнейшее взаимодействие ведётся без участия диспетчера.

3. Ценный вычислительный ресурс остаётся в монопольном пользовании клиента до тех пор, пока он удерживает прокси на него. Это позволяет монопольно занимать ресурс только на то время, когда он непосредственно используется.

4. Монопольный характер владения ресурсом позволяет гарантировать, что изменить состояние ресурса может только захвативший его объект, и даёт возможность использовать операции, зависящие от состояния. Это свойство может быть очень ценно, когда задача характеризуется большим объём передаваемых данных, накладных расходов, а для проведения очередного расчёта достаточно передать лишь небольшое количество изменившихся параметров.

Дополнение клиент-серверного подхода использованием локальных ресурсов и переход к распределённой иерархии

В классическом клиент-серверном подходе сервер берёт на себя всю вычислительную нагрузку, клиенты освобождаются от вычислительной работы и только обеспечивают интерфейс пользователя. Однако клиенты могут обладать ценными вычислительными ресурсами, заранее отказываясь от использования которых было бы не рационально.

Предложенная архитектура может быть расширена с использованием распространённого подхода Peer-to-Peer (точка-точка) для использования ресурсов локальной вычислительной сети клиентов следующим образом. На локальных компьютерах запускаются сёрванты, соответствующие типам доступного аппаратного

обеспечения и локальный диспетчер. При загрузке, локальные сёрванты регистрируются в локальном диспетчере. Диспетчер поддерживает типизированные пулы прокси-объектов на локальные сёрванты. При необходимости получения прокси-объекта локальные клиенты обращаются к локальному диспетчеру, который в первую очередь пытается вернуть прокси на локальный объект. Если же это не удаётся, то он переадресует запрос основному, вышестоящему диспетчеру серверной среды моделирования.

Таким образом, пользователи могут использовать как локальные, так и «облачные» вычислительные ресурсы. Если удаётся использовать локальные ресурсы, то для пользователей работа может идти быстрее, а на облачный вычислительный сервер снижается нагрузка.

Идея использования локального диспетчера может быть развита и применена для создания многоуровневой иерархической распределённой вычислительной системы, соответствующей иерархии диспетчерского управления ОАО «Газпром». На каждом уровне выделяются локальные вычислительные ресурсы и локальный процесс-диспетчер, который распределяет типизированные вычислительные ресурсы. Процесс-диспетчер отдаёт предпочтение локальным вычислительным ресурсам, но может обратиться и к процессу-диспетчеру выше по иерархии.

Интеграция разработанной подсистемы с ПВК Веста

Расчетный модуль, построенный по предложенной архитектуре, может быть поэтапно интегрирован с базовым ПВК «Веста».

На первом этапе ПВК «Веста» общается со своим традиционным расчетным модулем «Calc» по закрытому протоколу.

На втором этапе создаётся программа-адаптер, которая подменяет расчетный модуль Calc. ПВК «Веста» общается с адаптером так, как будто это прежний расчетный модуль Calc и передаёт ему запросы на выполнение задач. Для выполнения этих запросов адаптер обращается к серверу по новому открытому протоколу. На сервере для реализации необходимых расчетных процедур используется экземпляр Calc.

Постепенно вычислительные процедуры Calc переписываются на основе параллельных, распределённых вычислений и предложенной архитектуры.

На заключительном этапе ПВК «Веста» переходит непосредственно на новый протокол взаимодействия с серверной средой моделирования.

Заключение

Для проверки применимости предложенной архитектуры с её помощью была реализована распределённая система решения задачи поиска режима на графе системы газоснабжения, упрощённая схема которой была рассмотрена выше.

Были разработаны компоненты решения разреженных СЛАУ и SIMD моделирования множества трубопроводов, составляющих ТС с использованием технологии CUDA. Эти компоненты были запущены на облачных серверах Amazon Web Services. Был также разработан агрегирующий компонент, использующий прокси-объекты для осуществления расчета режима ТС. Построенная система была интегрирована с интерфейсом ПВК Веста.

С использованием предложенной архитектуры планируется разработать полнофункциональную многопользовательскую подсистему решения режимно-технологических задач СППДР и интегрировать её с базовым ПВК «Веста». Использование предложенной архитектуры при разработке такой подсистемы позволит обеспечить гибкость, возможности использовать наиболее современные технологии параллельных и распределённых вычислений, решать наиболее сложные вычислительные задачи, эффективно используя доступный потенциал современной вычислительной техники. Принципы иерархичности, распределённости, открытости соответствуют принципам, по которым создаётся единое информационное и вычислительное пространство ОАО «Газпром», что позволит в дальнейшем интегрировать построенную распределённую среду моделирования в ЕИП.

Литература

1. Функциональные требования к модернизированной автоматизированной системе диспетчерского управления единой системы газоснабжения (АСДУ ЕСГ). – 2009.
2. *Швечков В.А.* Автоматизация и управление технологическими процессами и производствами. дис. канд. техн. наук: защищена 06.11.2007. – М., 2007. – 169 с.
3. *Herb Sutter.* The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software // Dr. Dobbs's Journal. – 2005. - №.30(3).
4. *Сарданашвили С. А.* Расчётные методы и алгоритмы (трубопроводный транспорт газа). М.:ФГУП Изд-во «Нефть и газ» РГУ нефти и газа им. И.М. Губкина, 2005.
5. *Васильев А.В. Сарданашвили С.А.* Программная архитектура для использования параллельных вычислений при численном балансировании потоков газа на

расчётном графе газотранспортной системы. // Автоматизация, телемеханизация и связь. – 2012. - №1.