

Опыт разработки программного обеспечения системы биллинга ресурсов теплоснабжающего предприятия*

Д. В. Волков, П. Ю. ШЕРНАС

*Специальное конструкторско-технологическое бюро “Наука” КНЦ СО РАН,
Красноярск, Россия*

e-mail: vdv@kis.krasnoyarsk.ru

Представлен опыт реализации проекта разработки программного обеспечения системы коммерческого учёта тепла и расчётов с потребителями. Определены факторы сложности разработки. Приведены рекомендации по обеспечению производительности и надёжности программного обеспечения.

Ключевые слова: программное обеспечение биллинга, система коммерческого учёта тепла, балансовый метод расчёта, надёжность программного обеспечения биллинга, производительность программного обеспечения биллинга.

Введение

В последние годы правительством, муниципальными властями, бизнес-сообществом предпринимаются значительные усилия по модернизации систем ресурсоснабжения (поставки воды, тепла, электроэнергии). Совершенствуется законодательство, идет массовая установка приборов учёта, внедряются новые технические и технологические решения для повышения энергоэффективности.

Актуальной задачей в комплексе задач модернизации является создание автоматизированных систем коммерческого учёта тепловой энергии и других коммунальных ресурсов. Основа такой системы — это программное обеспечение (ПО) биллинга и выстроенные на его основе деловые процессы автоматизированного учёта ресурсов и расчётов с потребителями. Программное обеспечение биллинга должно обладать целым рядом уникальных свойств, однако типовой методики его создания нет. Для его разработки и эффективного применения требуется решение ряда научно-технических задач. Авторы в течение ряда лет выполняли пилотный проект по созданию и опытной эксплуатации программного обеспечения биллинговой системы теплоснабжающего предприятия, и полученный ими опыт может быть интересен.

1. Постановка задачи

Программное обеспечение [1] предназначено для автоматизации деятельности теплоснабжающих (ресурсоснабжающих) предприятий, выполняющих учёт ресурсов и услуг, начисления и расчёты с потребителями. Основные группы функций ПО:

*Работа выполнена в рамках проекта 14.1 программы Президиума РАН.

- описание конфигурации и характеристик сети поставки ресурсов;
- управление подключениями объектов снабжения;
- работа с приборами учёта (ПУ);
- расчёты с физическими и юридическими лицами.

Одна из базовых расчётных функций ПО — распределение объёмов потребления ресурса для составных объектов снабжения балансовым методом [2]. Сеть снабжения моделируется графом в виде дерева. Расчёт по алгоритму распределения осуществляется путём полного обхода дерева с выполнением следующих функций:

- автоматическое определение состава сложного объекта снабжения на основании данных о топологии сети поставки, структуре объектов снабжения и точках их присоединения к сети;

- выбор параметров распределения (нагрузки, площади, жители, процент, объём потребления для общедомовых приборов учёта (ОДПУ) и т. д.), в качестве которых могут выступать сведения, содержащиеся в договорах абонентов, в характеристиках объектов снабжения и т. д., в том числе при расчёте могут быть использованы несколько параметров распределения;

- расчёт суммы параметров распределения составного объекта (нагрузки, площади и т. д.) с учётом информации допуска, подключения/отключения объекта, нарушения режима потребления по ПУ и т. д. и сумм корректирующих объёмов потребления, добавляемых или вычитаемых из общего распределения (например, сумма потребления по приборам учёта, объём утечек, тепловых потерь и т. д.).

В общем случае распределение суммарного объёма сводится к следующей формуле:

$$Q_i = \left(P_i / \sum_{j=1,n} P_j \right) (Q + G_m).$$

Та же формула, выраженная через коэффициент распределения, имеет вид

$$K = (Q + G_m) / \sum_{j=1,n} P_j, \quad Q_i = P_i K,$$

где Q_i — распределённый объём i -го объекта снабжения; K — коэффициент распределения; P_i — параметр распределения (нагрузки, площади и т. д.) i -го объекта снабжения; n — число элементарных объектов снабжения в составном объекте снабжения; Q — общий объём потребления составного объекта снабжения; G_m — сумма дополнительных корректирующих объёмов (потребление ПУ, утечки, объёмы промывки и т. д.), в общем случае может быть как положительной, так и отрицательной величиной.

В соответствии с топологией сети распределение может быть многоуровневым (распределённый объём (Q_i) является исходным общим (Q) для дальнейшего распределения). Количество уровней может быть произвольным. Например, при расчёте по отпуску головного ПУ учитывается в качестве вычитаемых результатов отпуск по приборам на отдельных направлениях (например, в населенных пунктах, микрорайонах и т. д.). Далее, на следующем этапе распределения, учитываются показания ПУ, установленных на многоквартирном жилом доме, объёмы которого распределяются между отдельными квартирами и помещениями согласно нагрузке с учётом показаний индивидуальных приборов учёта.

2. Факторы сложности задачи

При разработке модели данных, архитектуры бизнес-логики работы ПО необходимо принимать во внимание следующие факторы, характеризующие рассматриваемый тип автоматизированных систем:

- структура сети снабжения — сложная, динамически изменяется во времени, характеризуется значительным числом параметров, часть из которых может изменяться под воздействием внешних и/или внутренних событий;

- множество используемых ПО алгоритмов расчёта открыто и постоянно пополняется, условия применения каждого расчётного алгоритма зависят от множества параметров, в том числе от времени, и могут изменяться;

- исходные данные для расчётов в любом (в том числе в прошлом) периоде могут быть изменены в любой момент времени, и возможен запрос на перерасчёт на любую глубину, вплоть до момента запуска ПО;

- постоянные изменения спецификаций алгоритмов и правил расчётов в новых периодах из-за изменения законодательства и условий применения ПО с необходимостью сохранения работоспособности алгоритмов и правил выполнения расчётов, действующих в прошлых периодах, начиная с момента эксплуатации ПО;

- необходимость обеспечения надёжности работы ПО в условиях внесения постоянных изменений в данные, алгоритмы, условия их применения и проч. при ограничениях на сроки внесения изменений и на время для тестирования.

Опыт разработки и эксплуатации ПО биллинга позволяет выделить наиболее критичные задачи, это обеспечение гибкости, производительности и надёжности ПО, а также создание эффективного механизма управления изменениями.

Обеспечение гибкости. В условиях непрерывной модернизации инфраструктур снабжения теплом и другими коммунальными ресурсами идет постоянное изменение алгоритмов, правил, условий расчёта на уровне как законодательства, так и практики его применения участниками расчётов. Это выводит на первый план обеспечение гибкости — восприимчивости ПО к постоянным изменениям. Особенностью биллинговых систем ресурсоснабжения является то, что необходимо не просто реализовать новые алгоритмы, а обеспечить при перерасчётах работу всех алгоритмов, поддерживаемых системой с момента ее создания. То есть спецификация функций системы является функцией времени. Изменения законодательства, внешних и внутренних условий работы не должны повлечь значительного перепроектирования ПО, а принятые решения не должны ограничивать возможные схемы его применения. Это означает, что модель данных, описывающая область автоматизации, должна быть как можно более детальной и отвечать критерию нормальных форм данных высокого порядка, расчёт должен быть подневным с анализом всех событий, меняющих конфигурацию сети, и всех параметров, влияющих на расчёт.

Обеспечение производительности. Полный перебор параметров, влияющих на расчёт, каждый день обеспечивает гибкость ПО к изменениям условий применения, но может сделать неприемлемыми показатели производительности — время выполнения расчёта, время подготовки отчёта. Исследуя возможности повышения показателей производительности на опытном образце ПО, авторы пришли к следующим выводам.

Для улучшения показателей производительности расчётов желательно:

- разделить базу данных (БД) на оперативную и архивную и встроить механизм, который позволяет анализировать глубину запроса на перерасчёт и в зависимости от глубины обращаться к соответствующей базе;

— осуществлять распараллеливание вычислительного процесса на несколько виртуальных машин при выполнении трудоемких расчётов.

Для повышения производительности при построении отчётов целесообразно использовать концепцию организации хранилища вычисляемых показателей. Суть концепции:

— определяется система часто используемых в отчётах предварительно вычисляемых показателей;

— организуется вычисление и запись показателей в хранилище данных в процессе выполнения расчётов;

— при формировании predetermined отчётов данные извлекаются из хранилища показателей.

Использование предложенного механизма на порядок ускоряет время формирования отчётов.

Обеспечение надёжности. Наиболее проблемным местом в работе ПО биллинга в условиях постоянных изменений правил и условий расчётов является надёжность. Результаты расчёта отражают финансовые обязательства участников ресурсоснабжения, и ошибки в расчётах воспринимаются сторонами очень болезненно. Источники ошибок: ошибки программистов при внесении изменений в код, неправильные действия пользователей, ошибки ввода данных из-за человеческого фактора, неверно специфицированный или интерпретированный запрос на изменения и проч. Необходимость постоянного внесения изменений в реально работающую систему в условиях ограниченного времени и ресурсов делает неприемлемым использование традиционной каскадной модели разработки и традиционных техник глубокого тестирования.

Осложняет процесс и то обстоятельство, что при одновременном изменении алгоритмов и данных трудно сформировать критерий безошибочности работы ПО и алгоритмизировать проверку соответствия ему ПО по результатам испытаний. Часто ошибка выявляется только в процессе детальной ручной проверки экземпляра расчёта. В качестве одного из путей преодоления указанных сложностей авторами предложены и опробованы два механизма: сценарное тестирование и тестирование на эталонных данных.

Суть сценарного тестирования заключается в следующем. В составе ПО реализован модуль создания и исполнения различных сценариев проведения расчётов на упрощённых наборах исходных данных. Подсистема позволяет пользователю без квалификации программиста создавать сценарии использования ПО, выполнять их как тесты, просматривать и запоминать правильные результаты тестов для последующих автоматических прогонов. К недостаткам сценарного подхода можно отнести затраты на обеспечение полноты всевозможных сценариев использования. К достоинствам — возможность комплексного тестирования ПО и возможность применения таких тестов уже на этапе разработки, в том числе для ускорения разработки за счёт формирования простого набора данных для отладки исходного кода.

Суть механизма тестирования на основе эталонных данных заключается в использовании в качестве исходных данных для проведения тестов эталонной копии реальной базы данных. После проведения расчётов, их “ручной” верификации просмотра результатов и выполнения отчётов имеется возможность запомнить результаты расчётов и отчётов как эталон. Впоследствии, при запуске новой версии, ПО автоматически “проигрывает” указанные сценарии и сравнивает их с эталонами расчётов и отчётов. В процессе эксплуатации ПО база сценариев и база эталонных исходных и результирующих данных могут постоянно пополняться, покрывая все новые потенциальные области возникновения ошибок. Данный подход позволяет в автоматизированном ре-

жиме находить большинство наиболее тяжелых ошибок в расчётах и отчётности при внесении изменений в код программы. Недостаток эталонного тестирования — относительная сложность первичного наполнения базы сценариев и поддержки эталонных баз при значительных изменениях функциональности биллинговой системы. Практическое применение показало высокую эффективность предложенных механизмов.

Важным элементом обеспечения надёжности является подсистема учёта и обработки инцидентов с ПО. Система реализует механизм регистрации инцидентов, классификации, формирования запроса на обработку инцидента, управления потоком работ по обработке запроса и формирования комплекса мероприятий по предотвращению инцидентов. Подсистема позволяет генерировать отчёты о показателях надёжности ПО, анализировать причины возникновения ошибок, формировать комплекс мероприятий по предотвращению ошибок ПО в будущем.

Управление изменениями. Обеспечение гибкости, производительности и надёжности необходимо, но недостаточно для организации эффективной промышленной эксплуатации ПО биллинга. В условиях постоянного изменения правил и условий расчётов необходимо реализовать механизм управления изменениями ПО. Он должен позволять всем участникам процессов расчётов осуществлять сквозной цикл работы с запросами на изменения: специфицировать запросы, вносить изменения в модель БД, в код ПО, в техническую документацию в соответствии с запросами, верифицировать внесённые изменения, генерировать и передавать заказчику обновление (новую версию) ПО с внесёнными изменениями.

Для управления изменениями предложено ввести в состав ПО в качестве обязательной компоненты систему обработки заявок (запросов) [3]. Каждый запрос на изменения оформляется в виде заявки на выполнение работ и регистрируется. С заявкой ассоциирован статус. Статус можно интерпретировать как лингвистическую переменную с данными, семантически значимыми для участников процесса (например, “ожидает исполнения”, “в работе”, “требует уточнения” и проч.). Регламент процесса предписывает каждому участнику выполнить определённые действия в зависимости от статуса заявки. Применение такой системы позволяет эффективно подключить всех участников расчётов к поддержке ее функциональности в актуальном состоянии с учётом требований бизнес-среды.

Заключение

Все предложенные решения были опробованы в проекте создания и опытной эксплуатации программного обеспечения биллинговой системы для ресурсоснабжающего предприятия Красноярского края, осуществляющего снабжение коммунальными ресурсами предприятия, организации и жителей города с населением порядка 120 тыс. человек.

Язык разработки — Java, используемая СУБД — MSSQL. Разработка ПО велась по методу очищаемого и наращиваемого прототипа. Встраивая в ПО предложенные в статье механизмы, удалось в 6–8 раз повысить производительность расчётов, в 10–12 раз снизить время выполнения основных отчётов, исключить массовые ошибки в расчётах, наносящие экономический и репутационный ущерб предприятию, уменьшить число некритичных ошибок, выявляемых в процессе практической эксплуатации, организовать удобное взаимодействие пользователей ПО с разработчиками в процессе внедрения и эксплуатации.

Список литературы

- [1] Волков Д.В., ШЕРНАС П.Ю. Программный комплекс для коммерческого учёта ресурсов, услуг и расчётов с потребителями “Шерпа” // Свидетельство о государственной регистрации программы для ЭВМ № 2012615921 от 28.06.2012.
- [2] Волков Д.В., ШЕРНАС П.Ю. Программный комплекс биллинговой системы коммерческого учёта ресурсов, услуг и расчётов с потребителями // Тр. XIII Всерос. науч.-практ. конф. “Энергоэффективность систем жизнеобеспечения города”. Красноярск, 2012. С. 100–103.
- [3] Волков Д.В., КОНСТАНТИНОВ П.Е., ШЕРНАС П.Ю. Система обработки заявок на выполнение работ: Свидетельство о государственной регистрации программы для ЭВМ № 2013616530 от 10.06.2013.

Поступила в редакцию 29 ноября 2013 г.