

ПРИМЕНЕНИЕ СИСТЕМЫ X-COM ДЛЯ ИССЛЕДОВАНИЯ РАЗВИТИЯ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКОГО КОМПЛЕКСА С УЧЁТОМ ТРЕБОВАНИЙ ЭНЕРГЕТИЧЕСКОЙ БЕЗОПАСНОСТИ

А.В. Еделев

Одним из важнейших направлений обеспечения приемлемого уровня жизни общества является обеспечение энергетической безопасности (ЭБ), которую по сути можно характеризовать как состояние защищенности ее граждан, общества, экономики страны от дефицита в обеспечении их обоснованных потребностей в энергии [1].

Актуальность исследований проблем обеспечения ЭБ в настоящее время весьма высока и определяется это значительным количеством угроз процессу нормального топливно- и энергоснабжения, таких как изношенность оборудования, социальная напряженность, высокая концентрация энергетических мощностей и т.д.

Разработанная в отделе живучести и безопасности систем энергетики ИСЭМ СО РАН методика поиска рациональных путей развития топливно-энергетического комплекса (ТЭК) с позиций обеспечения ЭБ на основе методов комбинаторного моделирования [2,3] позволяет:

1. Формировать варианты развития ТЭК и оценивать их допустимость по ресурсным, финансовым и другим ограничениям.
2. Сопоставлять варианты по различным критериям с целью выбора из них наиболее эффективных.
3. Исследовать влияние угроз ЭБ, выявлять рациональный путь развития ТЭК с позиций обеспечения ЭБ.

Множество вариантов развития ТЭК представлено в виде направленного графа, вершины которого будут соответствовать состояниям ТЭК в опорные годы, а дуги - переходам из одного состояния в другое.

В начале исследования создаётся так называемый базовый граф, содержащий по одному узлу для каждого опорного года. Эти узлы содержат основную информацию для расчётов.

Далее структура ТЭК разбивается на несколько составляющих частей, например, по территориальному признаку (по федеральным округам) или отраслевому признаку (по системам энергетики). Для каждой части экспертами строится свой граф изменений параметров по опорным годам.

Затем путем комбинирования информации об изменениях параметров различных частей ТЭК, принадлежащих одному временному разрезу, и наложении полученного набора изменений на базовое состояние, получается множество состояний ТЭК для определенного момента времени. Полученные состояния ТЭК соответствуют узлам графа развития ТЭК, которые затем связываются между собой дугами-переходами.

Все варианты развития ТЭК начинаются из начального узла - общего исходного состояния. С начального узла будет осуществляться проход по графу для оценки допустимости состояний ТЭК. Каждое состояние ТЭК оценивается на допустимость трехступенчатой системой ограничений.

Для реализации ограничений в виде логических условий были использованы списки пар *несовместных* узлов. Парой несовместных узлов называется пара узлов, принадлежащих различным графам развития, сочетание которых в возможном состоянии ТЭК по каким-либо причинам невозможно или не имеет смысла. Проверка отсутствия таких узлов является первой ступенью оценки допустимости состояний ТЭК.

Балансовая экономико-математическая модель ТЭК России является второй ступенью для оценки допустимости состояний ТЭК. Она представляет собой в математическом смысле классическую задачу линейного программирования, а в содержательном (энергоэкономическом) смысле базируется на традиционной территориально-производственной модели ТЭК с блоками электроэнергетики, тепло-, газо- и углеснабжения, а также нефтепереработки – мазутоснабжения [4].

Если модель относительно рассматриваемого состояния ТЭК имеет решение, то данное состояние могло бы считаться допустимым. Однако для того, чтобы варианты (траектории) развития ТЭК соответствовали требованиям ЭБ, в составляющих их состояниях необходимо оценить уровень ЭБ при помощи аппарата индикаторов ЭБ.

Аппарат индикаторов ЭБ является третьей ступенью для оценки допустимости состояний ТЭК. На этом этапе производится сравнение значения индикатора и его порогового значения. На текущий момент реализовано два индикатора: доля собственных источников в балансе котельно-печного топлива (КПТ) и доля доминирующего ресурса в структуре КПТ [1].

На рис. 1 показан пример создания простого графа развития ТЭК для 3 моментов времени. Самый верхний граф является базовым, ниже идут графы изменений значений добычи и экспорта газа. Созданный граф развития ТЭК состоит из 9 узлов. Проверку допустимости не прошло 4 узла. В них не обеспечивался требуемый уровень ЭБ. На последнем этапе был найден оптимальный (рациональный) путь развития ТЭК с позиций обеспечения ЭБ.

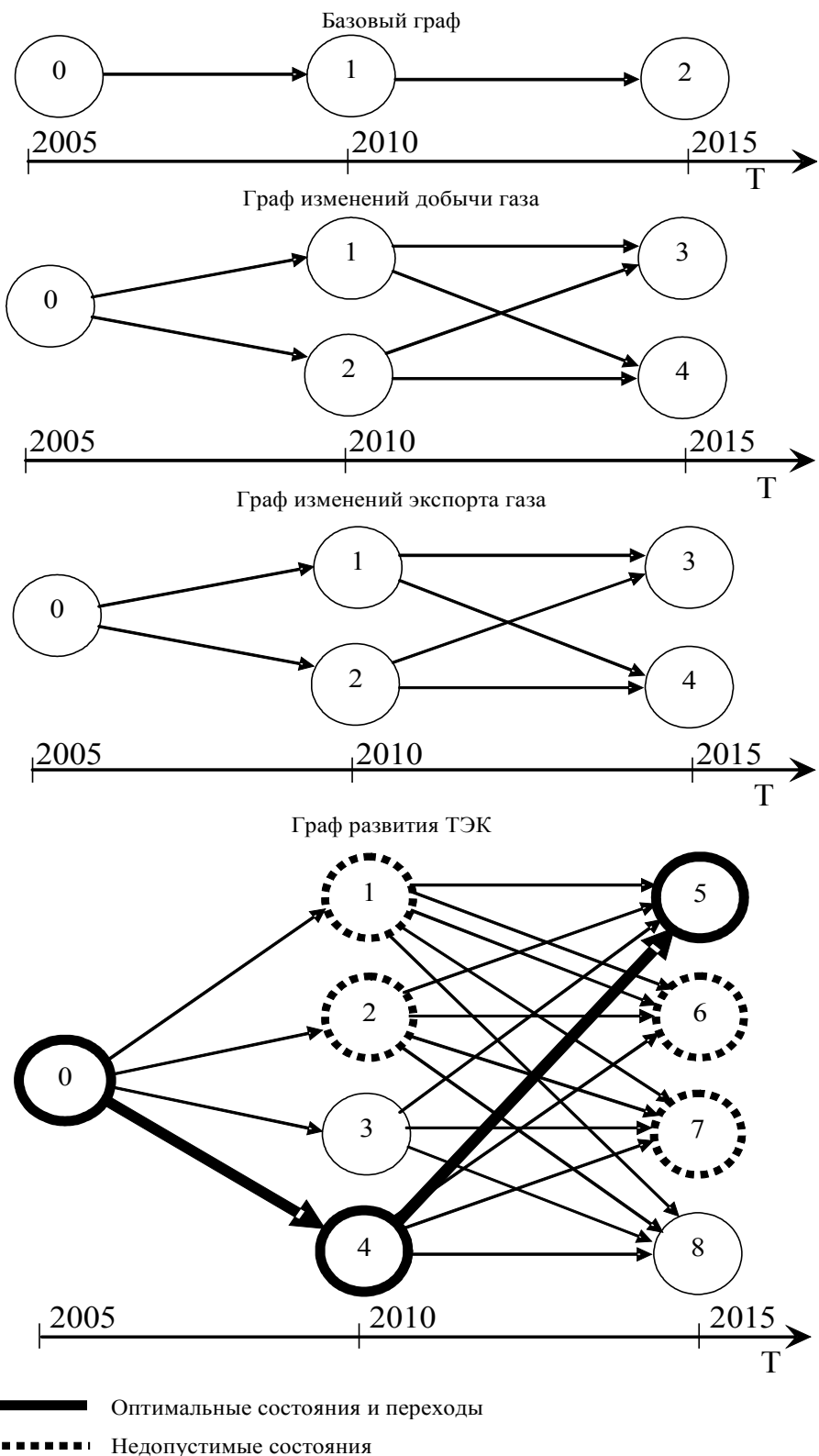


Рис. 1. Граф развития ТЭК.

Для исследований графов развития ТЭК с позиций обеспечения ЭБ был разработан программно-вычислительный комплекс (ПВК) “Корректива”. Ядром расчётного блока ПВК является библиотека, написанная на интерпретируемом языке программирования Lua[5], а библиотека для работы с графами, модуль комбинаторного моде-

лирования и решатель задач линейного программирования написаны на C++ и реализованы в виде динамически подгружаемых библиотек. Такое сочетание обеспечивает необходимую гибкость настройки и высокую скорость расчётов.

Эксперименты показали, что на обыкновенном персональном компьютере нельзя эффективно работать с графами, содержащими более 50 тысяч узлов из-за нестабильной работы интерпретатора Lua версии 5.1.3 при условии, что создание графа и его обсчёт должны проходить в один этап.

В настоящее время появилась потребность в рассмотрении графов развития ТЭК, содержащих до 1 миллиона узлов. Для обчёта графа такого размера требуется уже другой подход к организации вычислений. После создания графа каждый узел является полностью независимым от других с точки зрения проведения расчётов.

Следовательно, организовав с помощью распределённых вычислений обсчёт графа небольшими частями одновременно на нескольких компьютерах, можно сократить время вычислений и избежать проблем, связанных с нестабильной работой интерпретатора Lua при обчёте большого числа узлов.

Программное обеспечение (ПО) для организации распределённых вычислений должно отвечать следующим требованиям:

1. компактность и небольшой размер;
2. клиент-серверная архитектура;
3. простота настройки;
4. ориентация на вычислительную сеть, состоящую из нескольких персональных компьютеров.

Выполнение первого требования создаёт условия для интеграции системы (среды) распределённых вычислений в ПВК "Корректив". В то же время архитектура системы (среды) должна быть достаточно гибкой, чтобы вписаться в устоявшийся процесс исследования графов развития ТЭК с учётом требований ЭБ.

Клиент-серверная архитектура подразумевает, что ПО для организации распределённых вычислений и прикладная задача должны содержать серверную и клиентскую часть.

Были опробованы 3 небольшие системы, удовлетворяющие вышеприведённым требованиям:

1. Среда пакетно-ориентированных распределённых вычислений, разработанная под руководством Абасова Н.В. [6](ИСЭМ СО РАН);
2. ALua, событийно-управляемая среда для разработки распределённых приложений, созданная на основе языка Lua (Папский Католический Университет Рио-де-Жанейро, Бразилия)[7];
3. X-Com, инструментарий для организации распределённых вычислительных экспериментов (Научно-исследовательский вычислительный центр МГУ)[8].

Их основные характеристики приведены в табл. 1.

Таблица 1.

Программное обеспечение для организации распределённых вычислений

Название системы	Основные компоненты	Операционная система, взаимодействие сервера с клиентами	Интерфейс взаимодействия с серверной частью прикладной программой
Среда пакетно-ориентированных распределённых вычислений	Сервер управления, транспортные узлы и вычислители	Microsoft Windows, клиентская часть ожидает задания от серверной части	Исходные данные и результаты расчёта хранятся в виде файлов в двух каталогах на сервере.
ALua	Агент - интерпретатор Lua, обрабатывающий сообщения от интерфейса пользователя и других агентов в сети	Unix, клиентская часть ожидает задания от серверной части	Набор функций на языке Lua для отправки сообщений агентам, запуска и остановки прикладных задач.
X-Com	Сервер, вычислительный узел	Unix, клиентская часть посылает запросы к серверу на получение задания	Набор функций на языке C (интерфейс для языков C и C++). Обмен информацией через файлы (File API), расположенные в файловой системе сервера.

Среда ALua была наиболее привлекательным вариантом, так как целиком написана на Lua и её интеграция должна была пройти без особых проблем. Однако ALua не удалось заставить работать под Microsoft Windows из-за используемого ALua способа выполнения заданий. При получении задания процесс клиентской части создаёт свою копию и передаёт ей задание для выполнения, а сам продолжает ожидать получение нового задания. Для создания дочернего процесса путём копирования родительского используется системная функция fork, аналога которой в Microsoft Windows не существует.

Среда пакетно-ориентированных распределённых вычислений и система X-Com в экспериментах в целом показали себя одинаково. Операции по установке и обновлению клиентской части прикладной программы во время экспериментов в среде пакетно-ориентированных распределённых вычислений должны были производиться вручную, а в системе X-Com они осуществляются средствами самой системы. Основное преимущество X-Com в том, что система обладает более развитым интерфейсом взаимодействия с прикладной программой, поэтому выбор был сделан в пользу X-Com.

Через предназначенный для языков C и C++ интерфейс взаимодействия сервера X-Com с серверной частью прикладной программой можно использовать другие языки программирования[9]. Для реализации этого интерфейса на языке Lua была разработана библиотека luaхcom, которая при вызове сервером X-Com одной из функций интерфейса переадресует этот вызов соответствующей функции на Lua. Функции на Lua (табл. 2) должны быть определены в прикладной программе, имя файла которой изначально задаётся в файле настройки задачи X-Com.

Таблица 2.

Интерфейс взаимодействия сервера X-Com с серверной частью прикладной программой для языков C,

C++, Lua

Прототип функции на языках C и Lua	Описание функции
void initialize(char *arg) function initialize()	Инициализация серверной части прикладной программы. Параметр arg является названием файла прикладной программы на Lua.
char *getPortion(long N) function getPortion(N)	Запрос порции данных с номером N. Порция данных представляет собой текстовую строку языка C (null-terminated).
void addPortion(long N, char *data) function addPortion(N, data)	Обработка результата расчета порции данных номер N. В нее передается номер готовой порции данных и строка с результатами вычислений.
long getFirstPortionNumber() function getFirstPortionNumber()	Возвращает номер первой порции (это может быть любое целое число). Обычно порции нумеруются с единицы.
long getLastPortionNumber() function getLastPortionNumber()	Возвращает номер последней порции (любое целое число, большее, либо равное номеру первой порции). Если точно число порции не известно, то функция должна вернуть ноль.
int isFinished() function isFinished()	В случае если число порций не известно, то функция определяет, закончены ли вычисления. Она вызывается каждый раз после запроса очередной порции. Если число вычислительных порций известно точно, то функция всегда возвращает ноль.
double getPortionSizeGigaflop(long N) function getPortionSizeGigaflop(N)	Это необязательная функция, которая предсказывает вычислительную сложность порции данных с номером N. Она используется для оптимизации распределения заданий. Если предсказать вычислительную сложность произвольной порции затруднительно, то функция может для всех значений возвращать единицу.

Интерпретатор языка Lua не является потоко-безопасным (thread-safe), поэтому обращение к нему в библиотеке luaхcom производится с использованием критических секций. Это гарантирует, что одновременно только один поток X-Com получает доступ к интерпретатору языка Lua. Это, конечно, снижает производительность при частом вызове функций интерфейса, однако эта проблема может быть решена увеличением размеров порций данных.

При вызове функции void initialize(char *arg) библиотеки luaхcom выполняются следующие действия:

- выполняется инициализация интерпретатора Lua;
- загружается прикладная программа на Lua, имя файла которой передаётся в параметре arg;
- проверяется в загруженной программе наличие набора функций из табл. 2.

Клиентская часть прикладной программы в X-Com также называется вычислительной. Для передачи вычислительной части очередной порции данных используется специальный интерфейс, размещенный в файле с именем gctask. Этот файл представляет собой Perl-скрипт, в котором должны быть определены две функции (табл. 3) [9].

Таблица 3.

Интерфейс взаимодействия клиентской части X-Com с вычислительной частью прикладной программой

Прототип функции на языке Perl	Описание функции
sub gcsprepare	Предназначена для инициализации вычислительного модуля прикладной программы. Она не имеет аргументов и вызывается один раз, сразу после закачки архива с клиентской частью программы пользователя. Функция должна возвращать единицу, если инициализация прошла успешно и ноль, если в процессе инициализации произошли ошибки.
sub gctask	Подготавливает и запускает вычислительную часть прикладной программы. Функция имеет 5 аргументов: *0\$task имя задачи, которая сейчас решается *1\$taskarg аргумент для текущей порции данных *2\$portion номер текущей порции данных *3\$din имя файла с исходными данными (файл находится в текущем каталоге) *4\$dout имя файла, в который надо записать результат обработки данной порции (файл также должен находиться в текущем каталоге) Функция должна возвращать единицу, если расчет произведен успешно и ноль, если в процессе расчета произошла ошибка.

Для вычислительной части, написанной на Lua, первая функция просто возвращает единицу, а вторая функция вызывает исполняемый интерпретатор Lua и передает ему в качестве одного из параметров командной строки аргумент \$din. Стандартные потоки вывода и ошибок интерпретатора Lua перенаправляются в файл, имя которого содержится в аргументе \$dout.

Вместе с исходными данными должна передаваться некая строка с уникальной последовательностью символов. В стандартный поток вывода эта строка должна выводиться перед результатами обработки и после них. Таким способом результаты обработки отделяются от остальной информации, и серверная часть прикладной программы сможет легко их выделить при получении данных от вычислительной части.

Таким образом, реализованные интерфейсы позволяют создавать прикладные программы для системы X-Com на языке Lua, подобно как на C, C++ и других языках программирования. Согласно вышеописанному подходу, была создана и на текущий момент проходит отладку часть ПВК “Корректива” для обьема графа развития ТЭК на основе системы X-Com.

ЛИТЕРАТУРА:

1. Энергетическая безопасность России /В.В. Бушуев, Н.И. Воропай, А.М. Мастепанов, Ю.К. Шафраник и др. – Новосибирск: Наука. Сибирская издательская фирма РАН, 1998. -302 с.
2. Тимано-печорский ТЭК: Стратегии развития и пути повышения эффективности / Зоркальцев В.И., Иванова Е.Н., Калинина А.А., Кузиванов В.И. и др. –Сыктывкар. - КОМИ науч-ный центр УрО АН СССР. – 1991. –168с.
3. Региональный энергетический комплекс (особенности формирования, методы исследования) / А.Ф. Ануфриев, Г.М. Вишерская, В.И. Зоркальцев, А.А. Калинина и др. – Л.: Наука. – 1988. – 200с.
4. В.И. Зоркальцев Методы прогнозирования и анализа эффективности функционирования системы топливоснабжения. – М.: Наука, 1988. - 144 с
5. IERUSALIMSCHY, R.; FIGUEIREDO, L.; CELES, W. Lua - an extensible extension language. Software: Practice and Experience, v. 26, n. 6, p. 635–652, 1996.
6. Н.В. Абасов, Т.В. Бережных, А.П. Резников Долгосрочный прогноз природообусловленных факторов энергетики в информационно-прогностической системе ГИПСАР //Известия РАН, Энергетика, 2000, №6, С. 22-30.
7. URURAHY, C.; RODRIGUEZ, N.; IERUSALIMSCHY, R. ALua: Flexibility for parallel programming. Computer Languages, Elsevier Science Ltd., v. 28, n. 2, p. 155–180, 2002.
8. В.В. Воеводин, М.П. Филамофитский Суперкомпьютер на выходные //Открытые системы, 2003, №5, С. 43-48
9. Руководство пользователя системы X-Com <http://x-com.parallel.ru/guide.html>