

НОВАЯ ВЕРСИЯ АВТОМАТИЗИРОВАННОЙ ТЕХНОЛОГИИ АСООИ НА ВЫДЕЛЕННОМ СЕРВЕРЕ ГИДРОМЕТЦЕНТРА РОССИИ

Ю.А. Степанов, И.И. Жабина

*Гидрометеорологический научно-исследовательский
центр Российской Федерации
shabina@tcom.ru*

На протяжении многих лет в Гидрометцентре России разрабатывались и успешно эксплуатировались технологии АСООИ (автоматизированные системы оперативной обработки информации). При переходе от одной аппаратной платформы к другой, более современной, велись работы по адаптации, переработке и развитию АСООИ. Соответственно, были последовательно разработаны и эксплуатировались технологии АСООИ-ЕС, АСООИ-CRAY, АСООИ-Хеон4 [1–6].

Одновременно и параллельно с ростом вычислительных мощностей новых платформ существенно развивались и росли количественно и качественно **предметные области Гидрометцентра России**. Это в свою очередь вызывало необходимость все большего разделения труда в области информационного обеспечения, организации вычислительных пользовательских ресурсов, автоматизации оперативного вычислительного процесса и автоматизации администрирования этих ресурсов. Такая комплексная интеграция информационных и вычислительных ресурсов для различных предметных областей достигнута на высоком качественном уровне в рамках многомашинной технологии АСООИ-Хеон4 [1].

Ядро технологии АСООИ состоит из следующих подтехнологий:

- 1) АСООИ–СУБД – информационная основа (сервер баз данных);
- 2) АСООИ–БнЗ – Банк задач, вычислительная основа (сервер приложений);

3) АСООИ-АСУ – автоматизированная система управления, управляющая основа (сервер управления);

4) АСООИ-ОС – управление ресурсами ЭВМ (системными, пользовательскими и др.).

Технология АСООИ-Хеон4 является комплексной, многоуровневой интегрированной средой, которая обеспечивает надежное и эффективное управление пользовательскими и оперативными ресурсами сервера. Внутреннее устройство этой технологии является очень сложным, но эта сложность и обеспечивает надежность и внешнюю простоту использования, эксплуатации, сопровождения и администрирования [1].

В [1] рассматривается технология АСООИ-Хеон4 с методологической и научной точки зрения. Здесь обосновываются парадигмы масштабируемости и трехзвенной архитектуры управления в применении к оперативной практике Гидрометцентра России. Эти парадигмы нашли свое воплощение в виде следующих тесно взаимосвязанных подтехнологий: АСООИ-БнЗ (унифицированный Банк задач), АСООИ-АСУ (унифицированная управляющая среда), АСООИ-ОС (унифицированное управление ресурсами). Трехзвенная архитектура АСООИ («данные» ↔ «программы» ↔ «управление программами и ресурсами») является связующим управляющим звеном над классической двухзвенной архитектурой («данные» ↔ «программы»), которая обеспечивается подтехнологией АСООИ-СУБД.

Там же рассматриваются с «высоты птичьего полета» подтехнологии АСООИ-АСУ, АСООИ-БнЗ, АСООИ-ОС как с качественной стороны дела, так и с количественной. Далее в настоящей статье эти технологии рассматриваются с практической стороны дела.

Для успешного функционирования трехзвенной архитектуры технологии АСООИ-Хеон4 на каждом из серверов необходима надлежащая организация внутренних ресурсов сервера. **Ресурсы сервера** можно разделить на следующие категории:

- 1) ресурсы оперативной версии АСООИ;
- 2) ресурсы администраторской версии АСООИ;
- 3) ресурсы администратора АСООИ;
- 4) ресурсы пользователей АСООИ.

Эти категории должны быть непересекающимися множествами как физически, так и логически. Внутри категории (4) **ресурсы**

пользователей должны быть взаимно независимыми и гарантированными. Каждый пользователь (разработчик) ведет разработку приложений собственного Банка задач внутри собственного ресурса (см. [1, рис. 12]). **Администратор технологии АСООИ** должен иметь возможность вести собственные разработки как в прикладных областях, так и в области АСУ и информационных технологий. Эти работы ведутся внутри категории (3).

Администраторская версия технологии АСООИ является **ключевой** в составе ядра технологии, так как является мостом между пользовательской частью АСООИ и оперативно-значимой частью АСООИ (см. [1, рис. 12]).

Администраторский Банк задач является объединением копий пользовательских персональных Банков задач. Разработанные пользователями приложения АСООИ тиражируются администратором в объединенный Банк задач. Администратор осуществляет следующие виды работ (см. [1, рис. 9–12, табл. 3]):

- заново создает «под ключ» приложение (задачу) АСООИ точно таким же способом, как это делал пользователь в своем персональном Банке задач. Этот способ называется унифицированным АРІ Банка задач и реализован в виде стандартизованного управляющего программного обеспечения, разработанного в отделе СИО в 2003–2009 гг. Задача АСООИ *автоматически генерируется* из пользовательских исходных текстов и файлов данных, используя настроечные внешние параметры АРІ БнЗ. Такая задача становится полностью пригодной для оперативной версии АСООИ;

- проверяет работоспособность задачи в реальных условиях, т.е. моделируя работу задачи в оперативном режиме. При этом используется как вышеуказанный АРІ Банка задач, так и автоматизированная система управления счетом сеансов АСООИ (АСУ АСООИ), разработанная в 2003–2009 гг. в отделе СИО.

- Тиражирует задачу в **оперативный Банк задач** (либо новую задачу, либо обновляя текущую версию задачи) и удаляет из нее компоненты, которые существенно использовались при разработке, но не нужны при оперативном использовании. Такими компонентами являются, в частности, исходные тексты (каталог «sources»), спулы листингов («spool»), журналы АСООИ («log»), временные файлы («tmp») и т.д.

Остаются только три компонента: исполнимый модуль (каталог «bin»), файлы данных («datasets») и управляющее ПО АРІ Банка задач («lib»). Это и есть чисто оперативная задача АСООИ;

- проверяет работоспособность обновленной задачи в реальных условиях оперативной эксплуатации, т.е. внутри ресурсов оперативной версии АСООИ;

- если необходимо, администратор вносит изменения в **Управляющую базу данных АСООИ-АСУ** (реестр задач, реестр сеансов, реестр пользователей, сценарий автоматического запуска сеансов на счет и другие таблицы), т.е. изменяет **конфигурацию оперативного счета АСООИ**. Проверяет (моделирует) работоспособность обновленной конфигурации в администраторской версии АСООИ (см. [1, рис. 11]). Затем «зеркально» тиражирует *всю* подтехнология АСООИ-АСУ в **оперативную версию АСООИ-АСУ** и снова проверяет работоспособность и целостность АСУ (см. [1, рис. 12]). Тиражирование является инкрементным и гарантированным, с учетом всех изменений атрибутов файлов и каталогов и многоуровневой иерархии управляющего ПО. Для надежного тиражирования дисковых объектов (полного или частичного) используются мощные средства, входящие в состав ОС Linux;

- все вышеуказанные работы ведутся на основном сервере Хеон4b и называются **базовым администрированием технологии АСООИ-Хеон4**. На резервном сервере Хеон4a структура ресурсов должна быть (и является) аналогичной;

- все модификации подтехнологий АСООИ-БнЗ и АСООИ-АСУ, внедренные в оперативную версию АСООИ, администратор тиражирует надежными средствами сетевой синхронизации на сервер Хеон4a (см. [1, рис. 13]). Однако при переносе этих работ на другие, более современные Linux-серверы, возникают существенные вопросы совместимости ПО во всех подтехнологиях АСООИ. Эти вопросы рассматриваются в конце настоящей статьи.

Оперативная версия технологии АСООИ функционирует в полностью автоматическом режиме и обеспечивает регулярный регламентированный счет всех задач, входящих в состав **оперативного Банка задач АСООИ**, в соответствии с **управляющей базой данных АСООИ-АСУ**. Оперативная версия внешне создает впечатление статической, зафиксированной технологии. Однако ее развитие (как количественное, так

и качественное) является достаточно динамичным и обеспечивается вышеуказанными работами администратора АСООИ и пользователями АСООИ. Подробное описание этих работ приводится в производственной документации.

Для организации и гибкого обслуживания всех перечисленных ресурсов (пользовательских, оперативных, администраторских, системных) администратор АСООИ должен иметь возможность непосредственно работать на уровне операционной системы, т.е. осуществлять функции **системного администратора**. Все работы, связанные с системными и аппаратными ресурсами сервера, выделены в отдельную, очень важную подтехнологию, АСООИ-ОС.

Администратор АСООИ-ОС должен иметь прямой доступ ко всем видам ресурсов АСООИ. Для каждого пользователя администратором изначально создается и конфигурируется дисковый ресурс, не зависящий от других пользователей. Внутри дискового ресурса конфигурируется стандартизованная управляющая среда для разработки приложений АСООИ и создается **персональный Банк задач** пользователя. Устанавливается **дисковая квота** для пользователя. С этого момента пользователь может вести разработку «под ключ» любого числа задач, используя API Банка задач. При этом пользователь полностью избавлен от необходимости что-либо знать о системных ресурсах сервера и об устройстве и ресурсах технологии АСООИ. Поскольку способ разработки задач АСООИ унифицирован и внешне является очень простым, то пользователь теперь не должен использовать и поддерживать свое управляющее программное обеспечение, т.е. свои разнообразные скрипты. Вся «скриптовая часть» теперь обеспечивается API Банка задач.

Для каждой новой задачи администратор АСООИ создает «заготовку» – стандартный шаблон – это корневого каталога задачи и подкаталоги *sources*, *datasets*, *lib*. В каталог *lib* заносится стандартное управляющее программное обеспечение, т.е. API Банка задач. Это следующие shell-программы (скрипты):

AppConfigProg.1.sh

Launch.1.sh

LaunchSimple.1.sh

В этих скриптах можно «прошить» постоянные настройки (например задать опции компилятора, список дополнительных библиотек,

способ архивации и ротации листингов, включить/отключить компиляцию или выполнение задачи и много другого, «продвинутого»). Обычно это 3–4 индивидуальные настройки. Обязательно указывается мнемоническое имя задачи, например *oasyn*. Такие настройки называются **управляющими параметрами**, или **управляющими флагами АСООИ**, или просто **флагами**. В программе *Launch.1.sh* присутствуют все управляющие параметры API Банка задач (почти 100 штук). Программа *LaunchSimple.1.sh* является программной «оберткой» для *Launch.1.sh* и содержит в себе около дюжины самых практичных для пользователя параметров.

Программа *AppConfigProg.1.sh* содержит программный код, который автоматически выполняется перед запуском исполнимого модуля задачи. Здесь пользователь может добавить свой программный код (т.е. скриптовую часть, написанную на языке KornShell). На практике такая возможность используется редко, для специфических задач.

Для более специфического и тонкого управления задачей можно использовать (опционально) следующие predetermined администратором скрипты:

AppPreCompile.1.sh

AppPreRuntime.1.sh

AppPostRuntime.1.sh

В этих скриптах пользователь также может задействовать собственный программный код KornShell. Фактически это есть «plugin» модули. Схематический порядок (алгоритм) работы API Банка задач представлен на рис. 1.

Главная управляющая программа *appcplg.23.sh* является **многофункциональной**, т.е. состоит из многих **функциональных подсистем**, которые обеспечивают гибкий оперативный вычислительный процесс технологии АСООИ. Эта программа является **централизованной** и располагается внутри ресурсов администратора. **Множество управляющих параметров** (флагов АСООИ) позволяет динамически настраивать, конфигурировать, видоизменять поведение каждой задачи АСООИ в ходе вычислительного процесса технологии АСООИ (а также в ходе процесса разработки). Подробно эта технология API Банка задач описана в документации.

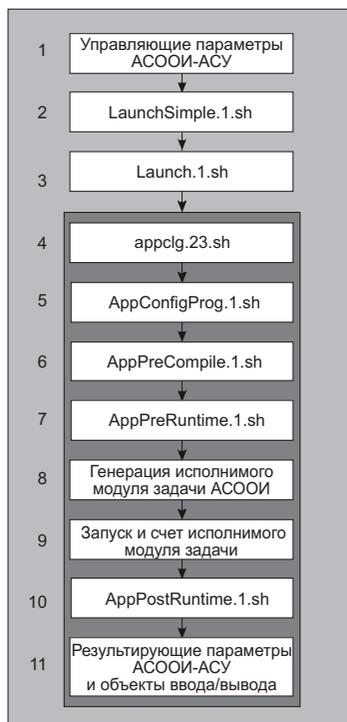


Рис. 1. Общая схема работы API Банка задач АСООИ.

В каталог *sources* пользователь заносит свои исходные тексты прикладной программы. Обычно это фортранные тексты. В головную Фортран-программу вносятся некоторые небольшие стандартизованные дополнения, реализующие взаимодействие Фортрана с shell-средой API БнЗ и обеспечивающие как передачу в Фортран-среду нескольких обязательных входных параметров АСООИ, так и получение из Фортран-среды нескольких выходных параметров. Эти входные параметры (7 штук) обеспечивают внутри программы минимальную стандартизованную привязку к информационным и управляющим ресурсам АСООИ. В каталог *datasets* пользователь заносит, если необходимо, файлы данных, используемые прикладной программой в процессе исполнения (т.е. в режиме *runtime*).

Процесс разработки любым пользователем АСООИ любой своей задачи АСООИ-БнЗ осуществляется следующим **стандартизованным унифицированным способом**:

1) модифицировать исходные тексты и файлы данных в каталогах *sources* и *datasets*;

2) в каталоге *lib* запустить на счет программу *LaunchSimple.1.sh*; необходимые управляющие параметры можно либо модифицировать непосредственно внутри программы, либо задать при запуске в командной строке, например: `dd=7 mm=11 hh=0 LaunchSimple.1.sh`;

3) посмотреть результаты работы прикладной программы и результаты работы управляющей программы (не обязательно) в каталоге *spool*. Этот каталог содержит, в частности, **листинги задачи**. Предыдущие версии листингов автоматически архивируются и сжимаются;

4) посмотреть, если интересно, каталоги *bin* (исполнимые модули), *log* (журналы событий АСООИ), *tmp* (временные файлы), *admin* (полезная информация о дисковых ресурсах задачи). Все эти каталоги, в том числе *spool*, создаются и обслуживаются управляющей программой *appclg.23.sh* автоматически;

5) повторить (продолжить) разработку, начиная с п. 1, если результат работы задачи неудовлетворителен.

Таким образом, в процессе разработки приложения АСООИ пользователь использует всего три простых команды Linux (*vi*, *cd*, *ls*) и одну команду АСООИ *LaunchSimple.1.sh*, т.е. процесс практически полностью автоматизирован, а уровень «рутины» практически нулевой для всех пользователей технологии АСООИ.

Каждая **задача АСООИ является унифицированным объектом**. Поведение такого объекта определяется как множеством **внешних управляющих параметров** (флагов) АСООИ, так и **внутренними объектами ввода/вывода**. Хранилищем таких объектов-задач является Банк задач. Структура Банка задач также унифицирована. Каждый разработчик имеет собственный Банк задач и наполняет его своими объектами-задачами. Объекты-задачи могут быть переносимы между Банками задач различных пользователей, в том числе и в объединенный Банк задач администратора АСООИ. При таком переносе остаются неизменными как поведение объектов-задач, так и их внутренние и внешние функциональные возможности (т.е. атрибуты задач).

Каждый Банк задач является **составным объектом**, компонентами (атрибутами) которого являются объекты-задачи. В совокупности с управляющей базой данных АСУ, администраторский Банк задач является фактически **объектно-ориентированной базой данных**, для которой применимы стандартные операции:

- добавить (INSERT) объект-задачу;
- исключить (DELETE) объект-задачу;
- модифицировать (UPDATE) объект-задачу;
- выбрать (SELECT) объект-задачу;
- выполнить, исполнить (RUN) объект-задачу.

Пользователь АСОИИ может внутри персонального Банка задач добавлять, исключать, модифицировать и исполнять свои объекты-задачи в «персональном» режиме, т.е. в режиме разработки, неоперативном, в соответствии с вышеизложенным порядком работ. Администратор АСОИИ в своем Банке задач может осуществлять те же действия, что и пользователь АСОИИ, а также выбирать и исполнять **группы объектов-задач** как в статическом, так и в динамическом режиме. Для этого задействуются следующие компоненты подтехнологии АСОИИ-АСУ:

1) **Управляющая база данных АСОИИ-АСУ** (реляционная, много-табличная);

2) **Управляющее программное обеспечение АСУ**, реализующее SQL-функционал SELECT для Банка задач;

3) **Объекты-сеансы**, определяющие группы задач и последовательность их исполнения; основными атрибутами объекта-сеанса являются:

а) **список (последовательность) имен объектов-задач**; список может формироваться статически или динамически; статические списки являются таблицами в управляющей базе данных; динамические списки задаются при запуске объекта-сеанса;

б) **комплект атрибутов задач**, т.е. комплект управляющих флагов АСОИИ, передаваемый в каждую задачу сеанса в процессе запуска на исполнение; этот комплект атрибутов задает одинаковое поведение каждого объекта-задачи внутри сеанса; например атрибуты

Flag_BackupRotate=8, Flag_BackupType=3

означают, что каждая задача в ходе своего выполнения будет осуществлять 8-дневную циклическую ротацию своих спулов и использовать полную систему именования ротируемых файлов;

в) **комплект атрибутов сеанса**, т.е. комплект управляющих флагов АСОИИ, контролирующих работу самого объекта-сеанса, но не объектов-задач; например атрибут Flag_TypeOfLaunch=AUTO означает автоматический режим запуска сеанса и статический список задач;

4) **Управляющее программное обеспечение АСУ**, предназначенное для контролируемого запуска и исполнения объектов-сеансов; это ПО является **интегрирующим, связующим** для следующих компонентов:

- Банка задач и всех объектов-задач;
- управляющей базы данных;
- ПО доступа к управляющей БД;
- всех объектов-сеансов;

5) **Управляющее программное обеспечение АСУ**, предназначенное для надежного и эффективного доступа пользователей и администратора к результатам исполнения сеансов АСОИИ; эти результаты находятся в **иерархических спулах АСОИИ**, которые содержат десятки тысяч постоянно изменяющихся файлов; поэтому процесс доступа должен быть (и является) автоматизированным и унифицированным; зная имя задачи, пользователь одной командой (например *lr1.oasyn*) получает таблицу листингов своей задачи (*oasyn*) за 8 суток в хронологическом порядке и выбирает любой из файлов, загружая его в текстовый редактор vi для последующего просмотра.

Выполнение объекта-сеанса в унифицированной управляющей среде АСОИИ-АСУ является **составной транзакцией**, состоящей из заданной (статической или динамической) последовательности объектов-задач (одиночных транзакций). При этом результаты работы задач (листинги, журналы событий и др.) автоматически перенаправляются в **централизованные спулы АСОИИ**. Доступ к спулам, поиск и выборка файлов из них автоматизированы и предельно упрощены для пользователей АСОИИ. Общая схема работы объекта-сеанса приведена в [1, рис. 10]. Результаты работы самих сеансов располагаются в собственных спулах и доступны администратору также с помощью унифицированных средств АСУ (пользователям эти спулы недоступны). Общая схема многоуровневой организации ресурсов и программного

обеспечения подтехнологий АСООИ-АСУ, АСООИ-БнЗ, АСООИ-ОС представлена в [1, рис. 12].

Такая организация управления ресурсами соответствует **объектно-ориентированному подходу** (парадигма ООП). Объект-задача, объект-сеанс, объект-Банк_задач и другие типы объектов, будучи созданы (разработаны) один раз, могут быть использованы в различных контекстах (в пользовательской среде, в администраторской среде, в оперативной среде) многократно. Эти объекты могут являться шаблонами для создания других однотипных объектов. Задавая различные значения атрибутов объекта (управляющих параметров АСУ), можно изменять (настраивать) функционирование объекта в зависимости от области применения. Таким образом, к объектам технологии АСООИ применимы такие концепции ООП, как **наследование, полиморфизм, инкапсуляция**. Это позволяет гибко и надежно управлять ресурсами АСООИ и обеспечивать их масштабируемость и развитие.

Также в рамках одного сервера объекты АСООИ являются локально портируемыми между различными ресурсами (пользовательскими, администраторскими, оперативными), поскольку объекты АСООИ привязаны к программно-аппаратной архитектуре конкретного сервера Хеон4b (тип и разрядность ЦП, версия ОС Linux, версии компиляторов и др.). Однако между серверами различных архитектур (платформ) объекты АСООИ не являются портируемыми (переносимыми, интероперабельными). Поэтому для того, чтобы переносить без изменений объекты АСООИ между серверами различных платформ, технологии АСООИ-БнЗ и АСООИ-АСУ должны быть портируемыми.

Необходимость разработки новых версий АСООИ-АСУ и АСООИ-БнЗ на современной серверной платформе

Новая версия технологии АСООИ, разрабатываемая на сервере Хеон5345 Гидрометцентра России, должна быть в целом преемственной к существующей технологии АСООИ-Хеон4, рассмотренной выше. Эта версия должна сохранить парадигму трехзвенной архитектуры управления АСООИ, изложенную в [1]. Ядро технологии состоит из следующих подтехнологий:

- АСООИ-БнЗ;
- АСООИ-АСУ;
- АСООИ-ОС;
- АСООИ-СУБД.

Подтехнология АСООИ-СУБД принципиально не изменяется с точки зрения программного обеспечения и администрирования.

Первые три подтехнологии концептуально должны остаться прежними в соответствии с вышеизложенными аспектами. Но их реализация должна быть существенно модернизирована. В первую очередь эта связка трех подтехнологий должна быть максимально портируемой (переносимой) между Linux-серверами различных платформ. Во-вторых, необходимо еще более упростить для пользователей процесс разработки приложений Банка задач АСООИ, используя современные подходы, в том числе объектно-ориентированное проектирование. В-третьих, надо апробировать такое модернизированное ядро АСООИ сначала на небольшом количестве задач, т.е. проверить как практическую портируемость ядра, так и удобство работы пользователей и администраторов.

Организация программно-аппаратных ресурсов сервера Хеон5345

Сервер Хеон5345 Гидрометцентра России имеет следующие базовые характеристики:

ЦП:	два 4-ядерных Хеон5345;
разрядность:	64;
диски:	8x70 ГиБ, RAID5, общий полезный объем ~390ГиБ;
LAN:	2 порта Ethernet 100/1000;
ОС:	RHEL5;
компиляторы:	IFC-10.1.017, IFC-11.1.073

Новый пользовательский интерфейс Банка задач АСООИ на сервере Хеон5345. Полная автоматизация разработки задач

В существующей в настоящее время версии API Банка задач все управляющее программное обеспечение было разработано и реализовано следующими средствами:

- интерпретатор Shell, являющийся встроенным в ОС Linux языком программирования;
- компиляторы Фортрана различных версий;
- множество утилит, входящих в стандартный комплект любой ОС Linux (например команды *grep(1)*, *awk(1)*, *tar(1)*, *bzip(1)*, *xargs(1)* и многие другие).

Компиляторы Фортрана предназначены для непосредственно компиляции и сборки прикладной (пользовательской) части задачи АСООИ. **Утилиты ОС Linux** используются для автоматизации многих управляющих аспектов разработки и администрирования задач. **Интерпретатор Shell** используется как связующая программная среда для указанных компонентов. Вся содержательная часть API Банка задач написана на языке интерпретатора KornShell с использованием множества утилит Linux.

Язык программирования KornShell является достаточно гибким и мощным для реализации весьма сложных управляющих алгоритмов и имеет такие сильные стороны:

- является интерпретируемым языком, т.е. не содержит стадии компиляции/сборки;
 - содержит все необходимые стандартные управляющие конструкции (if-then-else, case, for, while; арифметику, функции, простые массивы, ввод/вывод, перехват прерываний и др.);
 - содержит мощные средства управления процессами (каналы для межпроцессного взаимодействия, перенаправление ввода/вывода, хронометраж процессов, асинхронные процессы и ожидание завершения, шаблоны имен файлов и др.);
 - входит в состав всех современных дистрибутивов Linux.
- Основные недостатки интерпретатора KornShell:
- отсутствие как строгой, так и динамической типизации;
 - отсутствие встроенных средств обработки строковых и других типов данных;
 - отсутствие модульной архитектуры;
 - некоторые существенные несовместимости с интерпретатором BASH.

Поэтому при разработке АСУ в предыдущие годы пришлось сначала разработать на языке KornShell специальный макроязык,

который дополнял и расширял возможности интерпретатора. После этого были разработаны на KornShell библиотека функций и библиотека подпрограмм, основанные на строгой типизации переменных. Это позволило создать на языке KornShell модульное управляющее программное обеспечение, как это принято в процедурных языках программирования (C++, Си, Фортран и т.п.). Получилось очень надежное многофункциональное управляющее ПО, но очень сложное внутри.

При переносе этого ПО на сервер Xeon5345 и на другие Linux-машины выявились некоторые существенные несовместимости работы KornShell, а также компиляторов Фортрана и некоторых утилит ОС Linux. Весь комплект программного обеспечения АСООИ-БнЗ и АСООИ-АСУ (в том числе и компилируемая фортранная часть) оказался платформозависимым, т.е. не полностью портируемым.

В новой разрабатываемой версии АСООИ для сервера Xeon5345 производится поэтапный переход (и полная переработка) управляющего ПО АСООИ на новую систему программирования. Для этого был выбран современный кроссплатформенный объектно-ориентированный интерпретирующий **язык программирования Python**. Переход на язык Python позволяет осуществить следующие фундаментальные возможности:

- 1) перевести подтехнологии АСООИ-БнЗ и АСООИ-АСУ на объектно-ориентированные рельсы, что обеспечит высокую управляемость, надежность, эффективность всей технологии АСООИ;
- 2) обеспечить портируемость этих подтехнологий на различные серверные платформы;
- 3) обеспечить автоматизацию разработки задач АСООИ для пользователей и администраторов АСООИ;
- 4) обеспечить универсальный способ администрирования персональных Банков задач пользователей АСООИ;
- 5) обеспечить независимый от пользователей вычислительный процесс, используя объекты-задачи и объекты-сеансы АСООИ.

Переработка управляющего программного обеспечения с языка **KornShell** на язык **Python** осуществляется поэтапно. Это ПО состоит из функциональных подсистем, реализованных в виде компонентов очень большой Shell-программы *appcplg.23.sh*, Shell-подпрограмм и

большого количества Shell-функций. Переработка на Python осуществляется покомпонентно таким способом, чтобы новый (переработанный) компонент был оформлен в виде модуля языка Python. Это обеспечит модульную и портируемую архитектуру нового ПО.

Для примера рассмотрим новую версию **системы журналирования событий АСООИ**.

Объекты-журналы появляются путем создания экземпляра класса *C_Asoi_Logging_Setup*, определяемого в модуле *M_Asoi_Logging_Setup*. В любом месте Python-программы можно создать необходимый объект-журнал, открыть файлы журнала, осуществить запись необходимых сообщений в эти файлы, закрыть файлы журнала и удалить объект-журнал. Такие объекты-журналы являются независимыми друг от друга, но могут одновременно обрабатывать одни и те же файлы журналов на диске в асинхронном режиме.

Объект-журнал можно встроить в объект-задачу. Тогда каждая задача будет иметь собственную независимую систему журналирования событий. Множество задач АСООИ может выполняться как последовательно, так и параллельно; при этом журналы АСООИ будут целостными, и хронология поступления сообщений в эти журналы будет соответствовать действительности. Это обеспечивается интерпретатором Python независимо от платформы.

Аналогично системе журналирования событий АСООИ на языке Python разработаны такие функциональные возможности (подсистемы), как:

- работа с датой/временем;
- работа с конфигурационными файлами;
- работа с опциями командной строки;
- запуск и контроль процессов (в т.ч. управление Shell-скриптами);
- работа со средой ОС;
- системное программирование, заменяющее многие утилиты Linux;
- другие, более специфические подсистемы.

Каждая из таких подсистем реализована в виде соответствующего **Python-модуля**, содержащего один или несколько классов. **Классы** используются для динамического создания управляющих объектов внутри объектов-задач и объектов-сеансов.

Разработанные Python-модули для новой Python-среды АСООИ-АСУ и АСООИ-БнЗ позволяют реализовать новые возможности технологии АСООИ:

- модульная архитектура ПО;
- портируемость управляющего ПО;
- определение новых типов данных (т.е. новых классов) АСООИ;
- создание объектов новых типов и их динамическое использование в ходе выполнения основной программы;
- объекты АСООИ, в том числе и одного типа, могут создаваться в любом количестве и работать одновременно и независимо друг от друга (в том числе асинхронно и в параллельной среде);
- объект-задача АСООИ создает и использует управляющие объекты АСООИ независимо от других объектов-задач АСООИ;
- объект-сеанс АСООИ создает и использует как управляющие объекты АСООИ, так и объекты-задачи независимо от других объектов-сеансов;
- объекты АСООИ новых типов имеют идентичное поведение и свойства на различных платформах Linux/Python.

Новая «гибридная» версия автоматизированной системы управления АСООИ-АСУ

Управляющая среда АРІ Банка задач является **многоуровневой** и многоуровневой. Общая схема такой организации АРІ представлена на рис. 2. Уровни взаимодействуют между собой программным способом, задействуя необходимые **функциональные подсистемы АРІ**. Объекты ввода/вывода, т.е. дисковые объекты, создаются и/или используются на разных уровнях по-разному. На схеме это обозначено кружочками и стрелками.

В действующей технологии АСООИ-Xeon4 АРІ реализован уровнями 1, 2, 3, 5. В новой версии технологии АСООИ АРІ представлен всеми пятью уровнями. Новый уровень 4 достаточно подробно рассмотрен в документации. По мере разработки новой Python-среды АСООИ-АСУ часть функциональных подсистем постепенно переводится из уровня 3 (среда интерпретатора Shell) в уровень 4 (среда интерпретатора Python). Уровень 3 будет постепенно «утончаться», а уровень 4 будет «утолщаться».

Уровень 1 (прикладное пользовательское ПО) не знает (и не должен знать) что-либо о вышележащих уровнях, кроме некоторых деталей уровня 2.

Уровень 2 обычно является предустановленным вместе с **уровнем 5**. Оба эти уровня можно назвать **системными уровнями**.

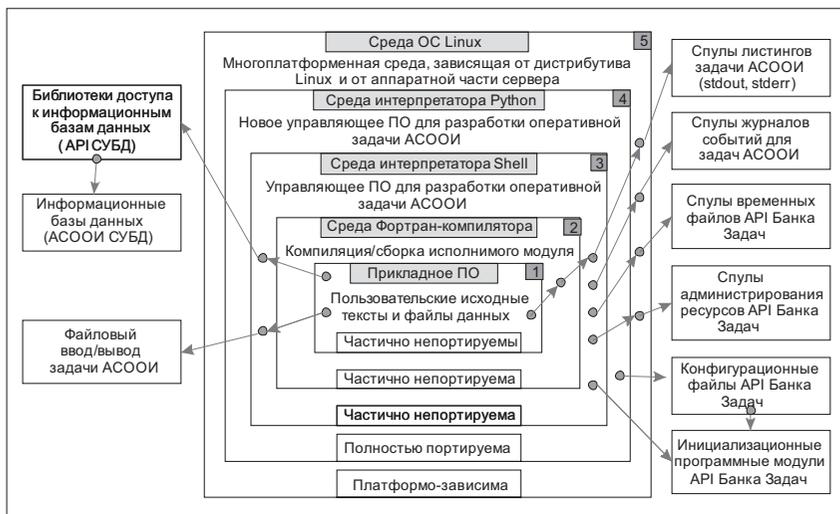


Рис. 2. Многоуровневая управляющая среда API Банка задач технологии АСООИ и объекты ввода/вывода.

Уровни 3 и 4 являются полностью собственной разработкой администраторов АСООИ и отдела СИО Гидрометцентра России. В новой реализации технологии АСООИ для сервера Xeon5345 оба эти уровня составляют **гибридную версию** API Банка задач, т.к. здесь совместно используются две среды разных интерпретаторов (Python и Shell).

Объект-задача АСООИ реализуется на уровнях 1, 2, 3, 4. Управление объектом-задачей осуществляется большим **комплексом управляющих флагов АСООИ**. Флаги можно передать в задачу несколькими способами в соответствии со **схемой приоритетов** на рис. 3. Все флаги по умолчанию «прошиты» в приоритетах П1 и П2. Любой из этих флагов может быть переопределен пользователем и/или администратором в приоритетах П3, П4, П5. Такая схема приоритетов позволяет

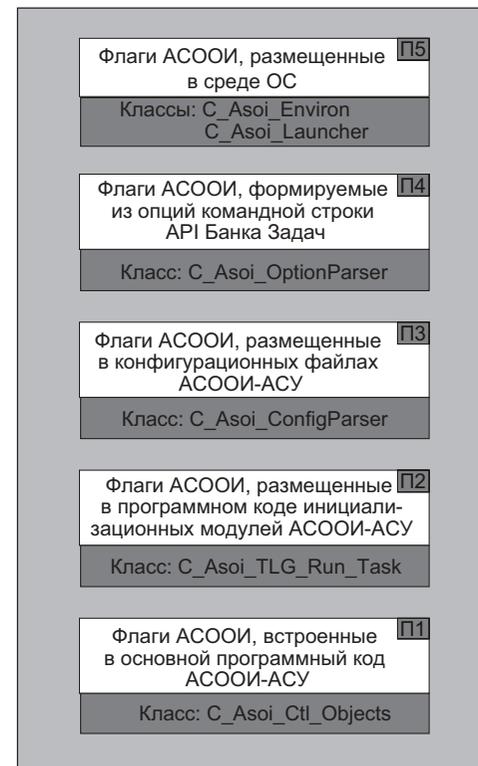


Рис. 3. Приоритеты использования и переопределения управляющих флагов АСООИ-АСУ.

управлять задачей внешним образом в разных контекстах. Например, в режиме разработки можно задать одни значения каких-либо флагов, в режиме администраторской версии АСООИ – другие значения тех же самых флагов, а в режиме оперативной версии АСООИ – третьи значения (например отключить все режимы компиляции/сборки модуля). В этих трех режимах идентичные объекты-задачи будут иметь различное поведение.

Из рис. 2 видно, что большая часть объектов ввода/вывода формируется и обрабатывается на уровне 3 (среда API Shell) с использованием системных средств уровня 5 (среда ОС). Объекты ввода/вывода

уровня 1 перехватываются уровнем 3 и применяются к соответствующим дисковым объектам. Таким образом, управляющие уровни 2, 3 (это, собственно, и есть API Банка задач) преобразуют непертируемые компоненты уровней 1, 2, 5 в портируемое представление, т.е. реализуют на практике **парадигму Middleware** (связующая среда). Непортируемые элементы уровня 3 (Shell) постепенно перерабатываются в портируемые аналоги и переносятся в уровень 4 (Python).

Новая версия API Банка задач существенно упростила работу пользователя-разработчика задач АСООИ, рассмотренную выше и представленную на рис. 1. Теперь вместо трех «громоздких» Shell-скриптов

AppConfigProg.1.sh

Launch.1.sh

LaunchSimple.1.sh,

которые пользователь должен был иногда модифицировать вручную, используется одна маленькая Shell-программа («запускалка») **LaunchTask.3.sh** (которую вообще не надо модифицировать, а только запускать на выполнение в командной строке терминала) и простой текстовый конфигурационный файл **AppConfigFile.1.ctl**, в котором можно модифицировать и/или добавлять значения необходимых **флагов АСООИ**.

Очень важная особенность нового API состоит в том, что **имя задачи** можно нигде не указывать. Управляющее ПО автоматически определяет это имя (которое должно быть уникальным во всей сумме всех Банков задач) по факту наличия в текущем Банке задач. Это очень упрощает как процесс тиражирования и клонирования объектов-задач, так и создание новой задачи из типового шаблона.

Для управляющего уровня 3 главная Shell-программа *appclg.23.sh* была существенно переработана и оптимизирована для использования в связке с Python-средой уровня 4 и переименована в *appclg.25.sh* (версия 25).

Для управляющего уровня 4 **новое программное обеспечение API Банка задач** состоит из 20 Python-модулей. Все имена модулей начинаются с префикса «*M_Asoi_*». Все эти модули взаимодействуют посредством определяемых в них классов и глобальных объектов. Рассмотренная выше «запускалка» является главной программой, которая при запуске на выполнение загружает (импортирует) цепочечно всю

иерархию модулей, и этот программный код сразу исполняется интерпретатором Python. Запуск головной программы — это и есть параметризация, конфигурирование, создание и выполнение объекта-задачи в автоматизированном режиме.

Все модули, классы, функции и методы подробно **самодокументированы** синтаксическими средствами языка Python (Python docstrings). Весь комплект этой документации может быть автоматически сгенерирован модулем *pydoc*. Структура этой документации точно такая же, как у всех модулей стандартной библиотеки Python.

Общие сведения о портируемости и кроссплатформенности программного обеспечения АСООИ-БнЗ и АСООИ-АСУ

Проблема портируемости программного обеспечения актуальна в современной индустрии компьютерных и информационных технологий. **Портируемое ПО** имеет ряд важных преимуществ в сравнении с обычным ПО. Одним словом, портируемое ПО более надежно и эффективно при практическом использовании. В связи с ростом и разнообразием парка ЭВМ вопрос портируемости постепенно становится актуальным и в Гидрометцентре России.

Для технологии АСООИ-Хеон4 портируемость всегда была одним из ключевых аспектов. Однако не было реальной возможности для разработки портируемого ПО. С появлением в Гидрометцентре России серверов современной архитектуры появилась возможность практической разработки и апробации портируемого ПО подтехнологий АСООИ-БнЗ и АСООИ-АСУ. Разработка ведется на сервере Хеон5345. Апробация ведется на нескольких доступных серверах различной архитектуры. Далее кратко рассматриваются факторы, которые влияют на портируемость ПО технологии АСООИ.

Разнообразие аппаратного обеспечения. Различие аппаратных архитектур в целом не является критичным, поскольку управление аппаратурой берет на себя операционная система. Современные ОС Linux могут полностью скрыть архитектурные особенности сервера, т.е. критичным становится системное ПО.

Разнообразие системного ПО. Любой дистрибутив Linux содержит большое количество программных средств (ПС), как низкоуровневых,

так и высокоуровневых. Такими средствами, в частности, являются системные библиотеки (например *glibc*), языки программирования и компиляторы (например *gcc*), интерпретаторы (например *bash*, *ksh*, *python*, *perl*), утилиты пользовательского уровня (например *vi(1)*, *ls(1)*, *grep(1)*, *find(1)*, *bzip2(1)*), утилиты привилегированного уровня (например *fdisk(8)*, *mount(8)*, *useradd(8)*), высокоуровневые языки программирования (Фортран, Си++, Python, Perl), некоторые интегрированные среды IDE, расширенные средства администрирования.

Многие из этих ПС используются прикладными пользовательскими программами (ПП) неявным образом. Некоторые из этих ПС напрямую используются в ПП. Если какой-либо компонент ОС Linux по-разному работает на разных платформах, то это может стать критичным фактором для ПП. Например, если пользовательский скрипт использует возможности утилиты *find(1)*, но результаты работы этой утилиты слегка различаются, то скрипт не является портируемым. Если интерпретатор *ksh* даже слегка различается в зависимости от версии ОС, то пользовательское ПО, написанное на этом языке программирования, не является портируемым, и, по-видимому, либо должно быть переработано в сторону универсальности, либо существовать в нескольких системно-зависимых версиях.

Особенно критичными для портируемости могут быть низкоуровневые системные средства и библиотеки (например *gcc*, *libstdc* и т.п.), которые обычно задействуются неявным образом. Высокоуровневые языки программирования компилируемого типа (например Фортран) также могут быть частично несовместимы от версии к версии, что ведет за собой переработку прикладного ПО.

Несовместимость пользовательского программного кода. Приложения Банка задач АСОИИ обычно написаны на языке Фортран. На сервере Хеон4b установлен компилятор IFC7, на сервере Хеон5345 – компилятор IFC10. Практика последних лет показала, что очень часто фортранские тексты, отлаженные в IFC7, компилируются в IFC10 с существенными ошибками. Это означает, что пользовательское фортранное ПО зачастую не является портируемым и требует существенной переработки.

Разнообразие пользовательских библиотек. В технологиях АСОИИ-Хеон4 и АСОИИ-Хеон5345 используются четыре специализированные

библиотеки собственной разработки отдела СИО Гидрометцентра России. Эти библиотеки задействуются в пользовательских прикладных программах для взаимодействия с такими внешними объектами, как базы данных, сеть, среда АСОИИ-АСУ, среда ОС и др. (см. [1, рис. 12]). Эти библиотеки реализованы на языке Фортран и, следовательно, являются частично платформозависимыми.

Библиотека ***bankload.a*** обеспечивает API для локального доступа к базам данных полей метеоэлементов.

Библиотека ***nablload.a*** обеспечивает API для локального доступа к базам данных наблюдений.

Библиотека ***remdbload.a*** обеспечивает API для удаленного доступа к базам данных АСОИИ внутри ЛВС Гидрометцентра России.

Библиотека ***ascpload.a*** является частью API Банка задач АСОИИ и обеспечивает взаимодействие среды Фортрана со средой ОС и со средой АСОИИ-АСУ. Библиотека также обеспечивает портируемость как файлов данных задач АСОИИ, так и всей структуры Банка задач. В режиме НИР эта библиотека не используется.

Вышеуказанные имена библиотек являются условными (мнемоническими). В новой версии технологии АСОИИ система именования и хранения библиотек (а также всех других объектов) является **самонастраиваемой и портируемой**.

Управляющее ПО АСОИИ-АСУ. Управляющее ПО технологии АСОИИ-Хеон4 реализовано в виде подтехнологии АСОИИ-АСУ и API Банка задач. Это ПО написано на языке программирования KornShell с использованием множества утилит Linux, поэтому оно не полностью портируемо. Для новой технологии АСОИИ-Хеон5345 разрабатывается портируемая версия, основанная на портируемом объектно-ориентированном интерпретаторе Python. Этот процесс подробно рассмотрен выше.

Поскольку версии Python 2.4 и 2.6 частично несовместимы сверху вниз, а на разных Linux-серверах установлена какая-либо одна версия, то новое ПО АСОИИ-АСУ разработано таким способом, чтобы оно стало портируемым между Python 2.4 и Python 2.6. Этот механизм универсальности кода является сложным, но вполне оправданным.

API Банка задач. API Банка задач является частью управляющего ПО АСОИИ-АСУ и предназначено как для полной автоматизации

разработки задач технологии АСООИ, так и для использования в оперативном вычислительном процессе в составе сеансов АСООИ. Поэтому проблемы портируемости АРІ Банка задач включают в себя все вышеперечисленные аспекты (системные, пользовательские, управляющие, библиотечные и др.).

В новой версии технологии АСООИ-Хеон5345 завершена разработка портируемого АРІ Банка задач, а также разрабатывается новая унифицированная и портируемая структура и организация всего программного обеспечения АСООИ на дисковых ресурсах любого Linux-сервера.

Предметные области и прикладное ПО, апробируемое в технологии АСООИ-Хеон5345. Новое управляющее ПО технологии АСООИ-Хеон5345 разрабатывается и апробируется одновременно по нескольким направлениям и на нескольких компьютерах различной программно-аппаратной архитектуры. Практическим результатом этой многоплановой работы должна явиться полная автоматизация различных предметных областей Гидрометцентра России на выделенном вычислительном ресурсе – сервере Хеон5345.

Апробация нового ПО осуществляется на новом Банке задач АСООИ, который постепенно расширяется по своему составу. Каждая задача тиражируется из администраторского Банка задач технологии АСООИ-Хеон4. Затем прежний АРІ БнЗ заменяется на новый АРІ и осуществляется тестирование и проверка совместимости и работоспособности. Затем моделируется счет сеансов АСООИ, содержащих эту задачу. Затем задача и сеансы добавляются в сценарий автозапуска АСООИ и осуществляется многократный – в течение суток – тестовый счет задачи в новой управляющей среде АСООИ-АСУ.

В настоящее время такую апробацию прошли такие предметные области, как глобальный ОА, ГСМА, региональный прогноз, мезомодели, авиационные прогнозы, морские прогнозы, ряд специализированных прогнозов, администрирование баз данных АСООИ. Ведутся работы над комплексом задач оценок прогнозов.

Одновременно осуществляется расширение состава и номенклатуры локальных баз данных АСООИ (для этих и других предметных областей).

В реальном режиме времени функционирует адаптированная для новой платформы технология приема и раскодирования данных наблюдений и продукции зарубежных центров.

По состоянию на 30.03.2011 г. существенная часть вышерассмотренных проблем портирования технологии АСООИ решена как в концептуальном, так и в алгоритмическом и программном аспектах. Эти новые разработки успешно реализованы и апробированы в новой версии АСООИ на сервере Хеон5345 (**технология АСООИ-Хеон5345**), и эта технология в перспективе может являться адекватной заменой существующей оперативной технологии АСООИ-Хеон4 Гидрометцентра России.

Список использованных источников

1. *Степанов Ю.А., Жабина И.И.* АСООИ-Хеон4 – многомашинная оперативная автоматизированная технология ГУ «Гидрометцентр России», предназначенная для информационного обеспечения, регламентированного счета и формирования продукции различных прогностических моделей // 80 лет Гидрометцентру России. – М.: Триада ЛТД, 2010. – С. 435–452.
2. *Жабина И.И., Пурина И.Э., Степанов Ю.А., Чекулаева Т.С.* Новые оперативные технологии обработки гидрометеорологических данных и управления прикладными программами с использованием суперЭВМ CRAY // 70 лет Гидрометцентру России. – СПб.: Гидрометеиздат, 1999. – С. 90–117.
3. *Кастин О.М.* Банк гидрометеорологических данных «ПРОГНОЗ». Общее описание БнД «ПРОГНОЗ-1060». – М., 1985. – 43 с.
4. *Кастин О.М., Жабина И.И., Степанов Ю.А.* Банк гидрометеорологических данных «ПРОГНОЗ». Средства и организация работы с базами данных БнД «ПРОГНОЗ». – М.: 1985. – 67 с.
5. *Пурин И.Э., Жабина И.И., Недачина А.Ю., Штырева Н.В.* Развитие информационных технологий в локальной сети CRAY // Труды Гидрометцентра России. – 2000. – Вып. 334. – С. 134–147.
6. *Фролов А.В.* Автоматизированная обработка оперативной метеорологической информации в ММЦ «Москва» с использованием суперЭВМ CRAY // Труды Гидрометцентра России. – 2000. – Вып. 334. – С. 3–18.

Поступила в редакцию 20.05.2011 г.