2 Экспертная система выбора оптимального алгоритма планирования и ее практическое применение

2.1 Применимость выбора оптимального алгоритма в прикладных системах

Полученные в экспериментальные результаты показывают, что адекватное использование системы выбора оптимального алгоритма планирования способно существенно уменьшить вычислительные затраты, соответствующие этапу составления плана действий, что в некоторых случаях позволяет расширить функциональные возможности интеллектуального агента. При этом следует выделить ряд необходимых условий, определяющих применимость данного подхода в прикладных задачах, в частности:

1. Доступность набора алгоритмов планирования (реализованных локально или удаленно).
2. Существование формально определенного критерия оптимальности алгоритма.
3. Доступность системы выбора, поддерживающей имеющийся набор алгоритмов и заданный критерий оптимальности.
4. Наличие информации о среде функционирования и задаче планирования.

В дополнение к перечисленным условиям следует особо отметить классы систем, для которых целесообразно применение априорного выбора алгоритма:

– системы планирования в динамической среде: множества переменных, описывающих среду, и действий, модифицирующих значения переменных, меняются от задачи к задаче. Типичными представителями данного класса являются узкоспециализированные программные агенты, ориентированные на решение специфических задач в сложных средах;
– системы, требующие решения множества непохожих друг на друга задач планирования в статической или динамической среде. К данной группе относятся проблемно-независимые многофункциональные интеллектуальные программные системы.

Выделенные особенности позволяют провести предварительную оценку применимости и целесообразности использования предлагаемого подхода в конкретной агентной системе. Основное внимание уделено описанию практической реализации системы выбора оптимального алгоритма планирования, в частности прототипу экспертной системы, построенной на основании научно-обоснованных результатов, а также ее использованию для решения некоторых технических задач.

2.2 Анализ требований к системе выбора оптимального алгоритма планирования

Функциональная схема экспертной системы выбора локально оптимального алгоритма планирования характеризуются следующими особенностями:

- вход: информация о среде функционирования агента и условиях задачи планирования, предоставляемая в виде характеристических показателей.
- выход: рекомендуемый алгоритм решения, выбираемый в соответствии с определенным критерием оптимальности.

В процессе анализа реализованных интеллектуальных агентов и существующих тенденций развития агентных систем выдвинут ряд дополнительных технических и функциональных требований, направленных на обеспечение высокого уровня адаптивности системы выбора в сочетании с простотой интегрируемости с другими программными компонентами:

1. Независимость от вычислительной платформы (аппаратной архитектуры и операционной системы), обеспечивающая широкую применимость системы в условиях отсутствия общепринятых стандартов на средства реализации агентных систем.
2. Автономность: минимальная зависимость от других программных и аппаратных компонентов, позволяющая использовать систему в виде опционального модуля для существующих и разрабатываемых интеллектуальных агентов.
3. Быстродействие и ресурсоемкость: учитывая выделенные ранее особенности функционирования интеллектуальных агентов, такие как потенциально жесткие временные рамки и ограниченные вычислительные ресурсы, особое внимание должно быть уделено обеспечению адекватного времени реакции системы при условии минимизации потребления вычислительных ресурсов.
4. Масштабируемость: поддержка произвольного набора алгоритмов планирования в рамках единого критерия эффективности. Как было отмечено ранее, продолжающиеся исследования по тематике планирования действий обуславливают постоянное расширение спектра доступных алгоритмов.
5. Реализация различных режимов работы, определяемых характером результатов, предоставляемых системой:
   - Рекомендательный режим: результатом является алгоритм планирования, выбираемый системой с помощью решающих правил на основании прогнозируемых значений критерия эффективности.
   - Справочный режим: система возвращает прогнозируемые значения критерия для всех рассматриваемых алгоритмов. Выбор оптимального алгоритма построения плана осуществляется интеллектуальным агентом.

Основными интерфейсными требованиями, определяющими возможные способы взаимодействия с системой, являются:

1. Поддержка распределенных вычислений (сетевого доступа).
2. Поддержка различных режимов взаимодействия с системой, в частности:
   - интерактивный с возможностью доступа через графический интерфейс пользователя;
   - пакетный с доступом к системе из интерпретируемого командного файла (скрипта);
   - программируемый интерфейс для языков высокого уровня/языков программирования агентов.

2.3 Особенности практической реализации

Для удовлетворения перечисленным требованиям при построении прототипа системы, структурно-функциональная схема которого представлена на рис. 2.1, использовались следующие технические решения:

- объектно-ориентированная реализация на языке Java, обеспечивающая функционирование на всех вычислительных платформах, для которых существует среда выполнения JRE (Java Runtime Environment);
- аппарат искусственных нейронных сетей для повышения быстродействия системы и качества предоставляемых результатов, а также минимизации используемых вычислительных ресурсов;
- масштабирование системы посредством использования для каждого алгоритма планирования автономного вычислительного канала (персептрона) с настройкой при помощи независимого конфигурационного файла;
- интегрированный WWW-сервер, обеспечивающий универсальный и эффективный доступ к системе по протоколу HTTP 1.0 (стандарт RFC1945) для различных режимов взаимодействия;
- выбор режима работы посредством ключа, позволяющего специфицировать тип запроса в момент обращения к системе (рекомендованный алгоритм или вектор прогнозируемых значений критерия оптимальности для всех доступных алгоритмов).

Рис. 2.1. Структурно-функциональная схема экспертной системы выбора оптимального алгоритма планирования действий

По результатам выполнения процедуры обучения нейронных сетей для каждого из поддерживаемых алгоритмов планирования формируется конфигурационный файл, содержащий топологию искусственной нейронной сети и значения весовых коэффициентов синаптических связей. При старте экспертной системы посредством аргументов командной строки указываются номер коммуникационного порта и список алгоритмов планирования, на основании чего система инициирует персептроны, соответствующие каждому из рассматриваемых алгоритмов, и активизирует интегрированный WWW сервер. Для обращения к системе клиент, которым может являться броузер с интерактивной HTML формой, сценарий пакетной обработки или непосредственно интеллектуальный агент, посылает по протоколу HTTP запрос типа GET заданного формата, содержащий требуемую для работы экспертной системы информацию, включающую значения характеристик среды и задачи планирования, а также вид запроса. После предварительной проверки полученных данных происходит активация нейронных сетей, причем значения, получаемые на выходе каждой из сетей вместе с видом запроса передаются в подсистему принятия решений, которая формирует ответ, передаваемый клиенту через WWW сервер.

Реализованный прототип системы удовлетворяет всем функциональным и техническим требованиям и может быть использован как в независимом режиме, так и в качестве оптимизирующего компонента подсистемы планирования действий.

2.4 Использование системы выбора алгоритма планирования для управления комплексной вычислительной средой

2.4.1 Задача управления комплексной вычислительной средой

Вычислительные среды, состоящие из совокупности аппаратных ресурсов, системного и прикладного программного обеспечения, являются одной из наиболее перспективных областей как для исследования, так и для применения интеллектуальных агентных систем ввиду уникального сочетания следующих характеристик:

- Реалистичность среды: в отличие от значительного числа идеализированных предметных областей, применяемых в исследованиях искусственного интеллекта, вычислительные среды принадлежат к реальному окружению.
- Относительно невысокие стоимостные требования к разработке и внедрению агентных систем. Поскольку в большинстве случаев интеллектуальные агенты представлены программными продуктами, они естественным образом вписываются в программную составляющую вычислительной среды.
- Отсутствие необходимости в решении ряда второстепенных проблем, например, учет искажений сигналов сенсорами агента.

В то же время современные вычислительные среды характеризуются многообразием взаимосвязанных неоднородных компонентов, определяющих необходимость в многошаговых процедурах управления.

Традиционные подходы к управлению компьютерными сетями связаны с использованием систем мониторинга, которые позволяют получить подробную информацию о состоянии компонентов среды, а также произвести базовый анализ значений показателей и идентификацию проблем по принципу проверки относительно регламентных диапазонов, после чего следует уведомление оператора о выявленных неполадках. Задачей нового уровня является обеспечение интеллектуального реагирования, состоящего в реализации автоматизированной целенаправленной деятельности с целью устранения выявленных неполадок, а в перспективе – выполнение превентивных действий в отношении потенциальных проблем. В простейшем случае подобная функциональность может быть достигнута в рамках реактивного планирования посредством выполнения предопределенных планов действий в зависимости от состояния среды по принципу ситуационного автомата. Более универсальный подход состоит в использовании аппарата планирования действий, при котором определяется обобщенная модель среды, а планы действий составляются динамически в зависимости от текущего состояния среды и поставленной цели.

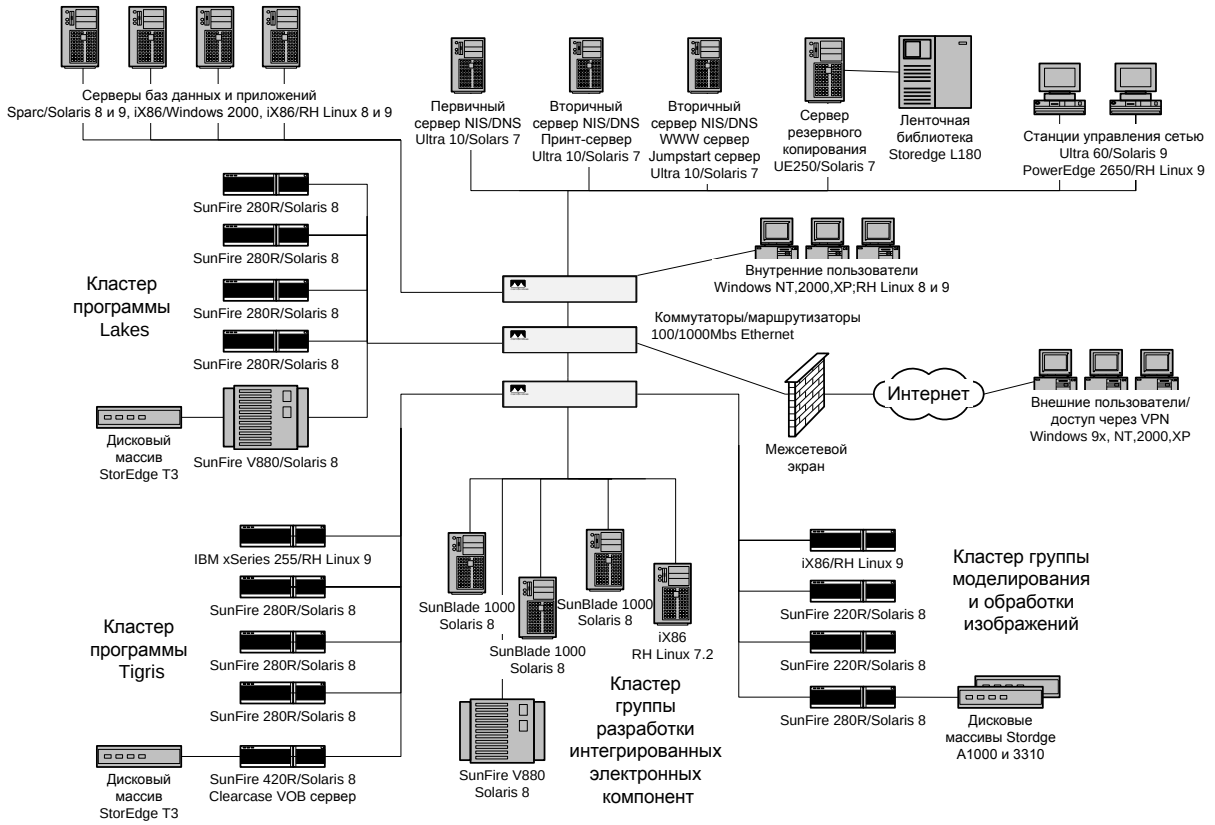

Рис. 2.2. Архитектура корпоративной сети

В качестве примера практического использования предлагаемого подхода к выбору оптимального алгоритма планирования использована его интеграция в подсистему планирования системы автоматизированного управления гетерогенными распределенными вычислительными ресурсами (рис. 2.2):

1. Аппаратные компоненты: процессоры различной архитектуры, оперативная память, дисковая и ленточная память, сетевые соединения.
2. Системное программное обеспечение (ПО): операционные системы (Sun Solaris версий 2.6, 7 ,8 и 9, Red Hat Linux версий 7.2, 8 и 9, а также Windows 9x/NT4.0/2000/XP), система резервного копирования (Legato Networker), менеджеры лицензий (Globetrotter FlexLM), система мониторинга (Big Brother), система распределения нагрузки (Sun Grid Engine), командные файлы (скрипты Unix shell, Perl).
3. Прикладное программное обеспечение, включающее в себя инженерные приложения (пакет математического ПО Matlab, система обработки изображений Sandpiper, средства моделирования микропроцессоров Synopsys и т. д.), средства разработки: система управления версиями Rational Clearcase, среда разработки (Sun Workshop, компиляторы GNU gcc/g++, средства отладки GNU gdb, ddd), базы данных (Oracle, MySQL), интерфейсное ПО (веб-сервер Apache, веб-браузеры Netscape и Internet Explorer).

Для функционирования управляющего программного обеспечения, включающего подсистемы мониторинга, управления моделью сети, диспетчеризации задач и планирования действий, используется группа рабочих станций управления сетью. Для сбора и предварительного анализа информации о состоянии управляемых объектов применен набор локально исполняемых программных агентов.

Основной задачей системы управления является поддержка функционирования вычислительного комплекса с заданными показателями надежности и доступности (системное администрирование), включающая в себя:

1. Мониторинг состояния вычислительного комплекса с уведомлением оператора и пользователей.
2. Устранение неполадок в проактивном (до наступления сбоя) и реактивном (после наступления сбоя) режимах, автономно или с вмешательством оператора.
3. Оптимизация выполнения ряда системных задач, например резервного копирования, налагающего ряд ограничений на доступность системы за счет значительного увеличения локального дискового ввода/вывода, высокой утилизации ЛВС для передачи данных на сервер резервного копирования и потенциальной блокировки локальных данных для обеспечения целостности резервной копии.
4. Оптимизация использования вычислительных ресурсов для выполнения прикладных задач, типичными примерами которых являются:
   - компиляция встроенного программного обеспечения;
   - компиляция программного интерфейса приложений (API);
   - программная симуляция работы микропроцессоров;
   - пакетная обработка изображений.
5. Сбор и обработка статистических данных и их наглядное представление (таблицы, графики, отчеты).

Необходимо отметить, что основные составляющие рассматриваемой задачи являются динамическими и в общем случае изменяются с течением времени по априори неизвестному закону. Например, совокупность аппаратных ресурсов меняется под влиянием следующих факторов:

1. Плановое обновление парка оборудования (по истечении сроков аренды).
2. Корректировка стратегии использования информационных технологий на уровне компании или подразделения за счет внешних и внутренних экономических факторов, в первую очередь финансового положения компании, а также появления на рынке новых технологических решений.

Наряду с изменениями во множестве аппаратных ресурсов, набор доступных действий постоянно меняется за счет:

1. Разработки новых компонентов в процессе решения текущих задач.
2. Обновления используемых продуктов с потенциальным расширением их функциональных свойств.
3. Интеграции в вычислительную среду новых программных продуктов.

Множество ограничений, накладываемых на процесс функционирования вычислительной системы, претерпевает качественные и количественные изменения в таких категориях как надежность, доступность, производительность и безопасность.

Принимая во внимание выделенные динамические особенности гетерогенной вычислительной среды, следует сделать вывод о невысокой эффективности применения жестко заданного алгоритма решения при проектировании и разработке системы управления в силу следующих факторов:

1. Низкая адаптивность к новым аппаратным и программным компонентам. Невозможность функционирования системы при качественных изменениях во множестве ресурсов.
2. Неэффективное использование ресурсов при изменении их функциональных характеристик (количественных изменениях во множестве ресурсов).

Наиболее перспективным подходом к построению системы управления является использование алгоритмов планирования для оперативной генерации плана действий для решения каждой подзадачи в заданный момент времени. Данный подход обладает следующими достоинствами:

1. Высокая функциональная гибкость, обеспечиваемая отсутствием жестко заданной последовательности действий.
2. Высокая эффективность за счет использования существующего программного кода.
3. Аналитические и диагностические возможности: при невозможности удовлетворения целевого состояния целиком в пределах текущего множества действий (программных компонентов) система способна выделить недостающую составляющую и предоставить рекомендации для разработки соответствующего программного фрагмента.

Необходимо отметить, что динамическое генерирование алгоритма решения (плана) обуславливает высокие вычислительные нагрузки на предварительном этапе (до непосредственного воздействия на состояние системы), выливающиеся, в частности, в потенциально длительные временные задержки, ограничивающие круг решаемых задач. Для минимизации негативного эффекта данного ограничения без потери функциональных свойств системы необходимо обеспечить использование наиболее эффективного алгоритма построения плана решения для данной прикладной задачи, что достигается за счет применения

разработанной экспертной системы априорного динамического выбора алгоритма планирования.

2.4.2 Моделирование предметной области

Одним из основных отличий реальной среды от идеализированной является отказ от предположения о закрытости предметной области, означающей, в частности, то, что агент может не располагать полной информацией о текущем состоянии среды. Поэтому для построения модели вычислительной среды, необходимой для применения алгоритмов планирования, использовано UWL-подобное представление, являющееся одним из наиболее известных расширений ADL для сред с неполной информацией.

Основными особенностями языка UWL, синтаксис которого в форме Бэкуса-Наура представлен в табл. 2.1, являются:

1. Трехзначная логика: истина (T), ложь (F) и неизвестность (U).
2. Три типа целей:
   - достижение: добиться выполнения заданного условия;
   - сопровождение: не допустить нарушения заданного условия в процессе выполнения плана;
   - сбор информации: дополнение модели, имеющейся в распоряжении агента, без изменения состояния среды.
3. Два типа переменных: стандартные переменные и переменные времени выполнения, значения которых являются неизвестными вплоть до исполнения заданных шагов плана.
4. Два типа постусловий: изменение среды и получение информации.
5. Два типа условных шагов: p-условие, определяющее выполняемую ветвь плана в зависимости от наблюдаемого значения истинности и v-условие, определяющее ветвь плана в зависимости от текущего значения переменной времени выполнения.

Ниже приведены примеры представления доступных действий системы управления с помощью языка UWL:

```
Имя:      (perform_backup ?server ?dir)
Предусловия:
     (найти ((isa server ?server ) . T))
     (найти ((isa directory ?server ?dir) . T))
     (достичь ((isa saveset ?server ?dir) . T))
     (найти ((backedup ?server ?dir) . F))
```

```
     (найти ((active ?server) . F))
     (достичь ((burhead.ready burhead) . T)
Постусловия:  (изменить ((backedup ?server ?dir) . T))

Имя:      (backup_status ?server ?dir)
Предусловия:  нет
Постусловия:
     (определить ((backedup ?server ?dir) .!boolean))
```

В первом случае определяется команда резервного копирования для пары <сервер; каталог>, которое выполняется в случае наличия емкости на сервере резервного копирования и невысокой активности локальной системы. Во втором случае описывается команда для сбора информации о статусе резервного копирования для заданной пары.

Необходимо также отметить, что для описания большинства действий может быть применена схема, независимая от конкретных используемых программных продуктов, при этом реализация конкретных действий кодируется на основе соответствующих прикладных или системных команд и/или командных файлов.

Таблица 2.1

BNF представление языка UWL

Задача планирования	::=	**Цель:** цель* | **Начало:** литерал* | **Действия:** оператор*
план	::=	*(p-условие | v-условие | шаг)** | ∅
значение истинности	::=	**T** | **U** | **F**
предикат	::=	постоянный символ, обозначающий имя предиката
константа	::=	постоянный символ
переменная	::=	символ в форме *?c*, обозначающий переменную
r-переменная	::=	символ в форме *!c*, обозначающий переменную времени исполнения
пк	::=	*переменная* | *константа*
rпк	::=	*r-переменная* | *переменная* | *константа*
пзи	::=	*переменная* | *значение истинности*
rпзи	::=	*r-переменная* | *переменная* | *значение истинности*
содержание	::=	(*предикат пк**)
r-содержание	::=	(*предикат rпк**)
литерал	::=	(*содержание . пзи*)
r-литерал	::=	(r- *содержание . пзи*)
цель	::=	((**достичь** *литерал*) | (**сохранить** *литерал*) | (**найти** *литерал*))*
постусловия	::=	((**изменить** *литерал*) | (**определить** r-*литерал*))*
оператор	::=	**Имя:** *имя* **Предусловия:** *цель** **Постусловия:** *постусловие**
шаг	::=	экземпляр оператора с определенными переменными
p-условие	::=	(**p-условие** *содержание* (*план*) (*план*))
v-условие	::=	(**v-условие** (*r-переменная = константа*))(*план*) (*план*))

2.4.3 Архитектура системы управления вычислительной средой

Структурно-функциональная схема системы управления представлена на рис. 2.3. Центральным компонентом системы является подсистема управления моделью вычислительной среды, содержащая формализованные знания в виде совокупности переменных состояния среды, их текущих значений и набора допустимых действий. Множества переменных и действий определяются системным инженером и могут корректироваться по мере изменения аппаратных, программных и функциональных свойств вычислительной среды. Поскольку в общем случае, в отличие от концепции классического планирования, состояние вычислительной среды изменяется под влиянием не только деятельности агента, но и набора внешних и внутренних факторов, значения переменных состояния определяются и корректируются на основании информации о текущем состоянии среды, предоставляемой штатной подсистемой мониторинга.

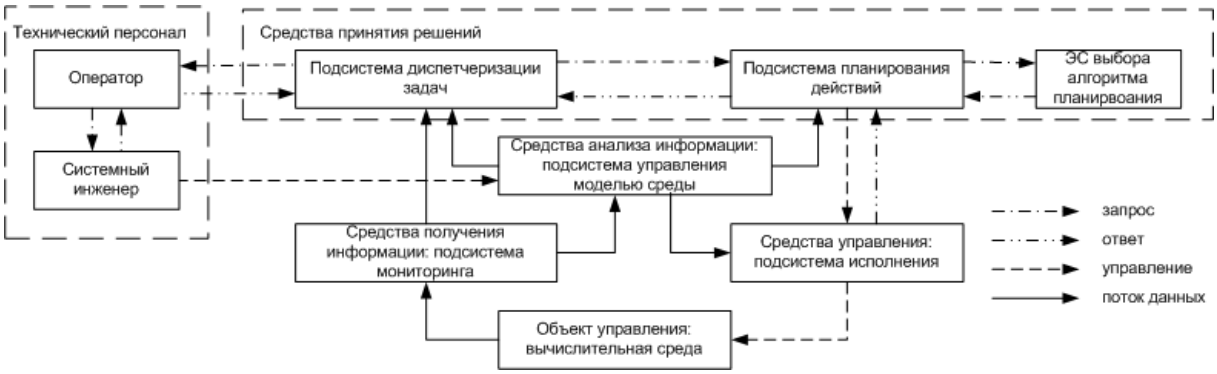

Рис. 2.3. Структурно-функциональная схема системы автоматизированного управления вычислительной средой

Подсистема диспетчеризации задач управляет текущим набором задач, стоящих перед системой, и выполняет две основные функции:

1. Формирование набора задач с использованием двух источников: проблемы, выявленные подсистемой мониторинга в автоматическом режиме (нехватка дискового пространства, неработающий сервис, аппаратный отказ), и задачи, задаваемые оператором в ручном режиме, что позволяет расширить функциональный потенциал системы.
2. Выбор задач на основании приоритетов и общий контроль за процессом решения.

Подсистема планирования составляет план действий, который передается в подсистему исполнения. Для синтеза плана действий используется алгоритм планирования, определяемый при помощи разработанной экспертной системы. В случае если в процессе выполнения плана происходит изменение состояния среды

под влиянием факторов, отличных от подсистемы исполнения, выполнение приостанавливается и происходит регенерация плана с учетом текущего состояния среды. Наличие законченного плана обеспечивает локальную состоятельность действий по отношению к поставленной цели в момент выполнения текущего шага и гарантирует достижение цели при условии единственности источника изменений.

Если план действий не может быть построен или задача остается нерешенной по достижении некоторого порогового значения, определяемого длиной плана или временем выполнения, задача исключается из рассмотрения и происходит уведомление оператора системы о необходимости внешнего вмешательства.

Состав программных средств, использованных при реализации системы, определен набором решаемых функциональных задач и присутствующих аппаратных и программных архитектур. Для реализации подсистемы управления моделью сети и средств принятия решений применен язык Common LISP (реализация GNU Common LISP) как наиболее адекватно отвечающий используемому UWL представлению. При построении средств мониторинга и управления задействованы командные файлы и утилиты, написанные на языках Bourne shell (для платформ Solaris и Linux), Perl (для платформ Solaris, Linux и Windows) и Visual Basic (для платформы Windows). Для обеспечения коммуникаций использован стек протоколов TCP/IP, непосредственная передача данных и команд управления осуществлена с помощью сетевых средств системы мониторинга (протокол BB) и Secure Shell v2. Как было отмечено ранее, экспертная система выбора алгоритма планирования реализована на языке Java с доступом по протоколу HTTP.

2.4.4 Оценка эффективности

Оценка эффективности предлагаемого решения автоматизированного управления вычислительными ресурсами производилась на трех основных уровнях, определяющих организационную структуру информационной модели предприятия:

1. Корпоративный уровень: эффективность использования средств, направленных на поддержание и развитие информационной инфраструктуры (общая стоимость владения ресурсами и эффективность их использования) в масштабе предприятия.
2. Пользовательский уровень: эффективность функционирования индивидуумов, непосредственно использующих вычислительную среду в качестве инструментария для выполнения своих служебных обязанностей.

3. Уровень поставщика информационных услуг, определяемый уровнем затрат на обслуживание и поддержку информационной инфраструктуры, удовлетворяющей заданному уровню качества предоставляемых услуг.

Основными факторами, оказывающими влияние на прямой и косвенный эффекты внедрения системы являются:

- На уровне предприятия: консолидация вычислительных ресурсов, сокращение парка дорогостоящей вычислительной техники без потери производительности со снижением как прямых (покупка/аренда), так и косвенных (транспортировка, энергоснабжение, инвентаризация) расходов.
- На уровне пользователя: гарантированный доступ к наиболее мощным из доступных ресурсов, обеспечивающий минимизацию времени решения задачи, автоматизированный мониторинг процесса исполнения, уменьшение потерь рабочего времени за счет использования пула вычислительных ресурсов.
- На уровне поставщика вычислительных услуг: повышение показателей уровня услуг (доступность, надежность), уменьшение затрат на поддержание (уменьшение временных затрат квалифицированных специалистов, увеличение соотношения поддерживаемых систем на единицу персонала).

В табл. 2.2 приведен сравнительный анализ прямого экономического эффекта от внедрения системы для всех трех уровней, рассчитанный в условных стоимостных единицах (УСЕ) для годового проекта, выполняемого группой, состоящей из 50 разработчиков программного обеспечения. Обобщенное изменение значения относительного показателя прямой эффективности (ОППЭ) составило 14.7% (или 59350 УСЕ), причем максимальный эффект был достигнут за счет консолидации дорогостоящих аппаратных компонентов и использования интеллектуального динамического распределения вычислительный ресурсов.

Таблица 2.2

Оценка экономической эффективности системы управления вычислительной средой

Уровень	Затраты		ОППЭ
	До внедрения	После внедрения	
Предприятия Стоимость аренды оборудования (рабочая станция Unix с интерфейсной картой PCi: 5000 УСЕ., сервер: 20000 УСЕ, рабочая станция Windows: 1500 УСЕ; 3 года аренды), тыс. УСЕ	83.3	58.3	31%
Пользователя Эффективность использования ресурсов (типовая задача: полная компиляция подсистемы, выполняемая один раз в день каждым разработчиком): Среднее время выполнения одной задачи, ч Затрачиваемое время за год 1 разр, ч Затрачиваемое время 50 разр, ч Стоимость ресурса, УСЕ./ч Общая стоимость, тыс. УСЕ	 0.48 120 6000 25 150	 0.42 105 5250 25 131.25	 12.5%
Поставщика ВУ Стоимость технической поддержки (одна единица персонала на 50 рабочих станций и 10 серверов), тыс. УСЕ	171.6	156	8.9%
Суммарное значение, тыс. УСЕ	404.9	345.55	14.7%

Для оценки непосредственного эффекта применения экспертной системы выбора оптимального алгоритма планирования использованы следующие показатели:

$T_{reaction}$ – среднее время начальной реакции системы, связанное с идентификацией задачи и генерацией исходного плана действий, измеряемое в условных временных единицах (УВЕ);

$T_{planning}$ – среднее суммарное время, затраченное на планирование действий в процессе решения заданной задачи, измеряемое в УВЕ;

R_{failed} – отношение числа задач, нерешенных системой управления по причине неспособности построить план действий при налагаемых временных и емкостных ограничениях, к общему числу рассмотренных задач.

Таблица 2.3

Оценка эффективности системы выбора оптимального алгоритма планирования действий

Показатель эффективности	Значение до внедрения ЭС	Значение после внедрения ЭС	Прирост эффективности
$T_{reaction}$	127.6	98.3	23%
$T_{planning}$	471.2	334.6	29%
R_{failed}	0.06	0.0225	62%

Табл. 2.3 содержит информацию об усредненных значениях введенных показателей до и после оптимизации этапа планирования действий с помощью разработанной экспертной системы. Как видно из представленных результатов, время начальной реакции системы уменьшено на 23%, общие временные затраты на планирование действий – на 29%, а уровень отказов, связанных с подсистемой планирования действий, снижен почти в три раза.

2.5Выводы

Раскрыто решение задач, определяющих практическую применимость разработанного подхода к выбору оптимального алгоритма планирования:

1. Сформулированы необходимые условия применимости предлагаемого подхода к выбору оптимального алгоритма планирования и выделены классы задач, являющихся наиболее перспективными для его использования.
2. Сформирован развернутый набор требований к программной реализации экспертной системы выбора оптимального алгоритма.
3. Описан прототип экспертной системы, удовлетворяющий заданным функциональным и технологическим требованиям.
4. Представлена успешная апробация прототипа ЭС в качестве расширения подсистемы планирования системы управления гетерогенными вычислительными ресурсами корпоративной компьютерной сети.

Полученные результаты позволяют сделать вывод о перспективности применения предлагаемого подхода при решении практических задач планирования. Использованные при реализации прототипа технические решения обеспечивают высокую производительность и адаптивность системы, а также позволяют использовать ее как в независимом режиме, так и в качестве локального или удаленного компонента интеллектуального агента.

Хотя рассмотренное в настоящем проекте исследование ограничено жестко выбранным набором наиболее известных нелинейных алгоритмов планирования и их производных, допустимо использование предлагаемого подхода без существенных изменений как для качественно новых алгоритмов того же класса, так и для модификаций существующих алгоритмов. Единственным ограничением, причем носящим чисто технический характер, является необходимость получения достаточного объема обучающих данных, что может быть произведено в соответствии с изложенной выше методикой, при этом процесс получения и использования обучающих данных может быть автоматизирован.

В заключение необходимо отметить, что принципы, лежащие в основе предлагаемого подхода, являются достаточно общими и могут быть применены к алгоритмам планирования, находящимся за пределами класса нелинейных алгоритмов с поиском решения в пространстве частичных планов, при условии адекватной адаптации (касающейся в первую очередь характеристических параметров среды и задачи планирования). Это обстоятельство является особенно важным ввиду того, что в последнее время появились новые перспективные разработки в области систем планирования, в первую очередь такие высокопроизводительные планировщики как Graphplan и Blackbox , использующих альтернативные подходы к построению планов действий и превосходящие рассматриваемые нелинейные алгоритмы по результатам отдельных тестов. Вместе с тем, хотя появление новых систем и алгоритмов планирования безусловно оказывает большое влияние на реализацию подсистемы планирования интеллектуального агента, главный стимулирующий фактор данного подхода – отсутствие глобально оптимального алгоритма планирования – остается без изменений, сохраняя тем самым актуальность задачи априорного динамического выбора оптимального алгоритма планирования.