**ОГЛАВЛЕНИЕ**

[**введение** 3](#_Toc357645861)

[**Глава 1. задача совмещения пк «Пирамида» с суппз** 5](#_Toc357645862)

[1.1. Состав и возможности ПК «Пирамида» 5](#_Toc357645863)

[1.2. Система управления прохождением параллельных задач 9](#_Toc357645864)

[1.3. Механизм сопряжения СУППЗ с прикладными программными системами 11](#_Toc357645865)

[Выводы по первой главе 13](#_Toc357645866)

[**Глава 2. Алгоритмы формирования конфигурационного файла пк «Пирамида» и паспорта задания суппз** 14](#_Toc357645867)

[2.1. Алгоритм формирования конфигурационного файла ПК «Пирамида» 15](#_Toc357645868)

[2.2. Алгоритм формирования паспорта задания СУППЗ 17](#_Toc357645869)

[2.3. Тестирование разработанных программных средств 20](#_Toc357645870)

[Выводы по второй главе 26](#_Toc357645871)

[**Глава 3. пк «Пирамида» в составе суппз** 27](#_Toc357645872)

[3.1. Опытная эксплуатация ПК «Пирамида» в составе СУППЗ 27](#_Toc357645873)

[Выводы по третьей главе 33](#_Toc357645874)

[**Заключение** 34](#_Toc357645875)

[**Литература** 35](#_Toc357645876)

[**приложение 1. Конфигурационный файл epkrun.conf** 36](#_Toc357645877)

введение

Программный комплекс «Пирамида» представляет собой систему, позволяющую организовать параллельное выполнение последовательной программы с распараллеливанием по данным, посредством запуска множества ее экземпляров. Достоинство указанного комплекса заключается в том, что он позволяет ускорить процесс разработки параллельной программы пользователем, не обладающим достаточными навыками в области программирования с организацией параллельных вычислений.

На настоящий момент, отсутствуют какие-либо средства позволяющие организовать коллективный доступ пользователей к вычислительной установке с возможностью использования на ней ПК «Пирамида» при организации вычислений. С задачей организации многопользовательского доступа справляется система управления прохождением параллельных задач (СУППЗ), которая обеспечивает выделение и освобождение ресурсов вычислительной установки, а также ведение очередей на запуск задач. Соответственно, актуальным представляется решение задачи сопряжения системы управления прохождением параллельных задач с программным комплексом «Пирамида».

Целью данной дипломной работы является разработка интерфейса и реализация сопряжения программного комплекса «Пирамида» и системы управления прохождение параллельных задач.

В рамках дипломной работы подлежат выполнению следующие задачи:

 изучение и практическое освоение ПК «Пирамида» и СУППЗ;

 изучение механизма сопряжения СУППЗ с прикладными программными системами;

 разработка алгоритмов и реализация программных модулей формирования конфигурационного файла и паспорта задания;

 проведение опытной эксплуатации ПК «Пирамида» в составе СУППЗ.

Практическим результатом дипломной работы является разработанный интерфейс, позволяющий объединять программный комплекс «Пирамида» с системой управления прохождением параллельных задач.

Настоящая дипломная работа изложена в пояснительной записке, состоящей из трех глав.

Первая глава пояснительной записки посвящена краткой характеристике ПК «Пирамида» и системы управления прохождением параллельных задач. Также рассмотрен механизм сопряжения СУППЗ с прикладными программными системами.

Вторая глава содержит в себе разработанную автором схему запуска ПК «Пирамида» через СУППЗ, описание алгоритма формирования конфигурационного файла на основе списков вычислительных модулей предоставленных СУППЗ, алгоритма формирования паспорта задания для СУППЗ, а также результаты тестирования разработанных программных средств.

В третьей главе пояснительной записки описан процесс проведения опытной эксплуатации ПК «Пирамида» в составе СУППЗ, рекомендации администратору ПК «Пирамида» по включению программного комплекса в состав СУППЗ.

Глава 1. задача совмещения пк «Пирамида» с суппз

1.1. Состав и возможности ПК «Пирамида»

Программный комплекс организации параллельных вычислений с распараллеливанием по данным (ПК «Пирамида») предназначен для функционирования на вычислительной установке, структура которой представлена на рис. 1.1.

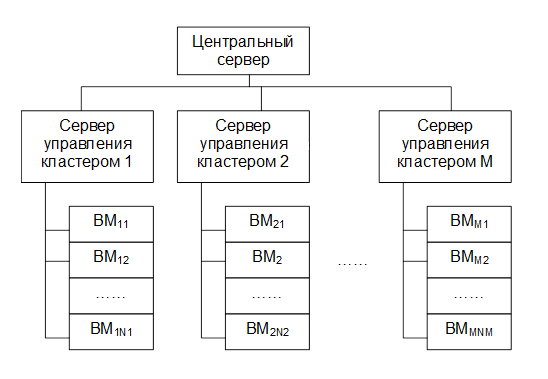


Рис. 1.1. Структура вычислительной установки

Вычислительная установка состоит из вычислительных кластеров, причем i-й кластер состоит из Ni вычислительных модулей (ВМi1 – ВМiNi). Каждый кластер управляется своим сервером управления, кроме этого в системе имеется центральный сервер управления.

Рассмотрим схему распараллеливания, на которую будет распространяться принятая в ПК «Пирамида» организация параллельных вычислений. В ее основе лежит схема «распараллеливания по данным». Пользователь реализует вычислительный алгоритм в виде обычного приложения (или приложения, использующего ПЛИС) так, чтобы одна последовательная программа (ОПП) принимала на вход значения параметров и обрабатывала соответствующую порцию данных. Предполагается, что вся необходимая для вычислений информация известна заранее, и межмодульное взаимодействие не требуется.

На каждом вычислительном модуле выполняется одна и та же копия программы с различными значениями входных параметров. От автора алгоритма не требуется знания основ параллельных вычислений, структуры и состава вычислительного комплекса. Он может использовать произвольные средства разработки приложений, придерживаясь лишь нескольких несложных рекомендаций по их созданию.

Ресурсы в ПК «Пирамида» должны иметь идентификаторы, соответствующие IP-адресам (сетевым именам) ВМ. Для каждого ВМ может быть задано число одновременно запускаемых на нем ОПП.

ПК «Пирамида» обеспечивает надежность вычислений. Выход из строя одного или нескольких вычислительных модулей, а также одного или нескольких кластеров, не останавливает расчеты, а только снижает их скорость.

Прикладная логика целиком реализуется в пользовательской ОПП, а компоненты ПК «Пирамида» решают задачи управления вычислительным процессом. К этим задачам следует отнести:

* разделение работы между ОПП путем перебора всех возможных комбинаций параметров;
* раздача работы ОПП путем передачи параметров и получение результатов расчетов от ОПП;
* мониторинг исправности вычислительных ресурсов. Доступность (исправность) ресурсов может быть задана ПК «Пирамида» перед запуском ОПП на основании предварительного тестирования или предыдущих запусков задачи.

ПК «Пирамида» представляет собой иерархическую систему менеджеров, показанную на рис. 1.2. В состав системы входят:

* центральный менеджер (ЦМ), функционирующий на центральном сервере управления;
* менеджеры кластеров (МК), функционирующие на серверах управления кластерами;
* менеджеры вычислительных модулей (ММ), функционирующие на вычислительных модулях (ВМ) кластеров.



Рис. 1.2. Иерархическая структура ПК «ПИРАМИДА»

На сервере управления кластером функционирует менеджер кластера (МК). Менеджер кластера на каждом ВМ запускает менеджеры ВМ (ММ). Если помимо ВМ в состав кластера входит другой кластер, то менеджер кластера верхнего уровня запускает на сервере управления подчиненного кластера соответствующий менеджер кластера.

Менеджер ВМ запускает один или несколько экземпляров ОПП и контролирует их выполнение. Менеджер ВМ вызывает ОПП как исполняемый модуль, задавая значения параметров, с которыми будет исполняться ОПП. После этого, этот менеджер ВМ ждёт (определённое время – время таймаута) завершения ОПП. Код завершения ОПП и результаты передаются в менеджер кластера. Если через время таймаута ОПП не завершается, менеджер ВМ завершает работу с возвращением аварийного результата МК.

Всем целым управляет центральный менеджер задания (МЗ), запущенный на центральном сервере.

ПК «Пирамида» имеет следующие характеристики:

* число поддерживаемых кластеров центральным менеджером – до 8 кластеров;
* число поддерживаемых ВМ менеджером кластера – до 128 ВМ;
* число одновременно запускаемых менеджеров ВМ экземпляров ОПП – до 128 экземпляров;
* максимальное число параметров ОПП – 10 параметров.

Обращение к ПК «Пирамида» осуществляется посредством запуска центрального менеджера задания – программного модуля с именем epk**.** Центральный менеджер задания должен запускаться на центральном сервере управления. Формат запуска следующий:

epk -f <конфигурационный\_файл> -p <паспорт\_задания>

Параметры:

конфигурационный\_файл – файл с описанием вычислительных ресурсов и параметров запуска ПК «ПИРАМИДА».

паспорт\_задания – файл в формате XML с описанием формата вызова и параметров ОПП.

Более подробно формат запуска ПК «Пирамида» и входные параметры описаны в документе «Программный комплекс «Пирамида». Руководство пользователя»[1].

Менеджеры кластеров и ВМ запускаются центральным менеджером автоматически в соответствии с информацией конфигурационного файла ПК «Пирамида».

1.2. Система управления прохождением параллельных задач

Система управления прохождением параллельных задач (СУППЗ) обеспечивает коллективный доступ пользователей к многопроцессорной вычислительной установке, с целью предоставления пользователям дополнительных вычислительных ресурсов. В процессе работы, СУППЗ выполняет следующие функции:

* прием запросов пользователя;
* планирование запуска пользовательских программ;
* запуск и останов программ;
* контроль выполнения программ;
* выделение и освобождение вычислительных ресурсов.

СУППЗ представляет собой совокупность взаимодействующих процессов управляющей ЭВМ (рис. 1.3):

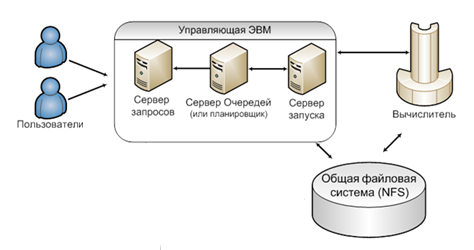


Рис 1.3. Схема взаимодействия процессов СУППЗ

* процесс-сервер запросов;
* процесс-сервер очередей;
* процесс-сервер запуска пользовательских программ.

В контексте СУППЗ, пользовательская программа, предназначенная для выполнения, называется заданием.

Основной задачей управляющей ЭВМ является ведение очереди заданий. Для решения указанной задачи на управляющей ЭВМ запущены сервер запросов, сервер очередей и сервер запуска.

Сервер запросов выполняет обработку пользовательских команд. В качестве команды может выступать запрос на выполнение задания, запрос на остановку выполнения задания, запрос на удаление задания из очереди задач, запрос состояния задания, запрос состояния очереди задач и другие.

Сервер очередей осуществляет планирование заданий и обеспечивает постановку их на выполнение, для этого сервер очередей вызывает сервер запуска.

СУППЗ оперирует с задачами пользователей. В контексте СУППЗ, задачей или заданием называется прикладная программа, предназначенная для выполнения.

Для описания задачи используется специальный файл – паспорт задания. Паспорт задания – текстовый файл с описание параметров задачи и скриптов для подготовки к запуску. Через паспорт задачи реализуется еще одна необходимая функция – сопряжение СУППЗ с прикладными программными системами.

Ключевыми параметрами задания являются:

* имя задачи;
* время выполнения задачи;
* вычислительные ресурсы.

К вычислительным ресурсам относятся:

* процессоры;
* оперативная память;
* дисковая память;
* дополнительные вычислительные устройства.

Готовую для запуска задачу пользователь отправляет в очередь СУППЗ. Далее компоненты СУППЗ, отвечающие за обработку задач, проверяют корректность паспорта задания, и при обнаружении ошибки возвращают пользователю диагностическое сообщение. Корректно оформленная задача обрабатывается планировщиком СУППЗ и ей назначаются ресурсы и время запуска (резервирование). При наступлении времени запуска задача ставится на счет, а по окончании счета принудительно снимается со счета.

При неудачном запуске задача блокируется и через определенный администратором интервал времени будет повторно поставлена в очередь. Кроме этого, администратор самостоятельно может блокировать всю очередь или конкретную задачу.

## 1.3. Механизм сопряжения СУППЗ с прикладными программными системами

В системе управления прохождением параллельных задач имеется механизм сопряжения с прикладными программными системами. Более подробно указанный механизм будет рассмотрен на примере программной системы MPI.

Для того чтобы СУППЗ корректно запустила программу и следила за ее выполнением, пользователь должен создать для своей программы паспорт задания. Паспорт задания представляет программу в очереди и оформляется в виде отдельного файла. Формат файла паспорта задания СУППЗ представлен в руководстве администратора СУПЗ[3].

Для программ, использующих MPI, паспорт задания создается и ставится в очередь автоматически, посредством применения команды mpirun.

В паспорте задания в обязательном порядке присутствует секция Batch, в которой размещается исполняемый код запуска задания на счет. Из секции Batch генерируется специальный файл runmvs.bat, который, после прохождения очереди, запускает задачу на счет. При запуске задачи, файлу runmvs.bat в качестве параметра передается файл, в котором содержится список вычислительных модулей, предназначенных для счета задачи. Схема запуска MPI-программы представлена на рис. 1.4.

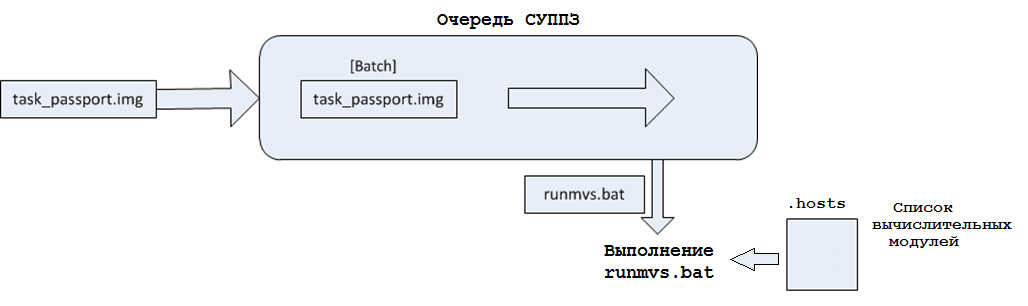


Рис. 1.4. Схема запуска MPI-программы через интерфейс сопряжения СУППЗ

СУППЗ выделяет вычислительные ресурсы для счета задач по мере их освобождения, при этом заранее не известно, какие именно модули будут выделены.

Для корректной работы ПК «Пирамида» необходимо, чтобы выделенные СУППЗ вычислительные модули взаимодействовали в виде иерархического дерева. Иными словами, один из выделенных СУППЗ вычислительных модулей должен выполнять функции центрального менеджера, а остальные вычислительные модули должны взять на себя функции менеджеров кластеров и вычислительных модулей.

Таким образом, при совмещении СУППЗ и ПК «Пирамида» выявлены следующие препятствия:

 неизвестно, какие именно модули будут выделены СУППЗ;

 необходимость назначения ролей центрального менеджера, менеджеров кластеров и менеджеров вычислительных модулей выделенным ресурсам.

Выводы по первой главе

Автором были выполнены следующие задачи:

* проведено изучение и практическое освоение программного комплекса «Пирамида» и системы управления прохождением параллельных задач;
* изучен механизм сопряжения СУППЗ с прикладными программными системами, в частности с интерфейсом передачи сообщений MPI.

Проделанная работа позволяет сделать вывод, что механизм сопряжения СУППЗ с прикладными программными системами позволяет организовать взаимодействие с программным комплексом «Пирамида». Подобное взаимодействие возможно при условии решения задачи назначения ролей центрального менеджера, менеджеров кластеров и менеджеров вычислительных модулей выделенным СУППЗ вычислительным ресурсам.

Глава 2. Алгоритмы формирования конфигурационного файла пк «Пирамида» и паспорта задания суппз

В настоящей дипломной работе, рассмотренный ранее механизм сопряжения СУППЗ с прикладными программными системами используется следующим образом:

* пользователь запускает командный файл epkrun, на вход которому подает паспорт задания для ПК «Пирамида». Команда epkrun анализирует свои параметры запуска, составляет паспорт задания для СУППЗ и выполняет постановку паспорта в очередь;
* в процессе выполнения файла runmvs.bat при постановке задачи на счет, на основе выделенных вычислительных модулей генерируется конфигурационный файл ПК «Пирамида», в котором выделенные модули будут структурированы в виде иерархической древовидной структуры;
* после генерации конфигурационного файла, на выделенных модулях происходит последовательный запуск исполняемых файлов ПК «Пирамида» в соответствие с составленной в конфигурационном файле иерархической структурой.

Полная схема представлена на рис. 2.1.

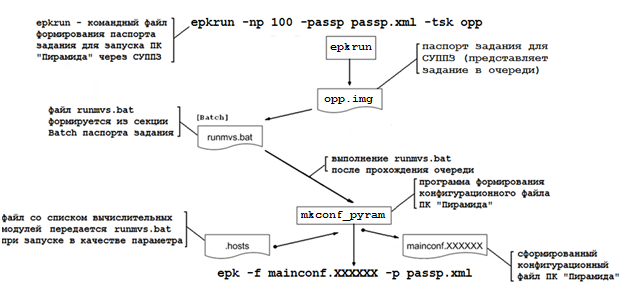


Рис. 2.1. Схема сопряжения ПК «Пирамида» и СУППЗ

При вызове командного файла epkrun происходит формирование паспорта задачи и постановка его в очередь. Из секции Batch паспорта генерируется файл runmvs.bat, который на момент запуска получает файл со списком выделенных вычислительных модулей. В процессе запуска задачи файл runmvs.bat, на основе полученного списка формирует конфигурационный файл запуска ПК «Пирамида» и производит запуск командного файла центрального менеджера epk.

В общем случае, для каждой конкретной задачи, на вычислителе разворачивается собственный экземпляр ПК «Пирамида», на котором производится вся вычислительная работа.

В ходе выполнения поставленных в настоящей дипломной работе задач автором были разработаны командные файлы формирования паспорта задания СУППЗ (epkrun, runmvs.bat) и программа формирования конфигурационного файла ПК «Пирамида» (mkconf\_pyram).

2.1. Алгоритм формирования конфигурационного файла ПК «Пирамида»

Для того, чтобы ПК «Пирамида» корректно работал на выделенных вычислительных модулях, необходимо чтобы выделенные вычислительные модули были способны выполнять функции центрального сервера, серверов управления кластерами и вычислительных модулей для ПК «Пирамида».

Следовательно, каждому выделенному СУППЗ вычислительному модулю необходимо задать место в иерархическом дереве ПК «Пирамида». Выделенные вычислительные модули отображаются в специальном файле в виде списка. Каждый модуль отображается записью в отдельной строке. Формат строки:

Имя\_модуля:количество\_процессоров\_в\_модуле

Алгоритм формирования конфигурационного файла реализован в виде программы mkconf\_pyram и представлен на рис. 2.2.



Рис. 2.2. Алгоритм формирования конфигурационного файла ПК «Пирамида»

Работа алгоритма подразумевает наличие двух ограничений – ограничения на количество кластеров и ограничения на количество вычислительных модулей в кластере.

Необходимость задания подобных ограничений обусловлена тем, что у администратора ПК «Пирамида» в составе СУППЗ должна быть возможность строить иерархическое дерево ПК «Пирамида» оптимальной конфигурации, с целью равномерно загружать работой выделяемые пользователям вычислительные модули.

Если пользователь в ходе вызова задал число процессоров большее, чем необходимо при заданных ограничениях, алгоритм завершится, сообщив об этом в уведомлении.

В результате работы алгоритма выделенные СУППЗ вычислительные модули перестраиваются в иерархическую структуру и выводятся в стандартный поток вывода в формате конфигурационного файла ПК «Пирамида».

2.2. Алгоритм формирования паспорта задания СУППЗ

Алгоритм формирования паспорта задания СУППЗ реализован на основе рассмотренного ранее механизма сопряжения СУППЗ с интерфейсом MPI. Реализация алгоритма представляет собой командный файл epkrun, при запуске которого выполняются следующие действия:

1. Анализ задаваемых пользователем входных параметров и считывание требуемых пользователем ограничений, накладываемых на конфигурацию ПК «Пирамида». Если какой-либо параметр задан неправильно, отсутствует обязательный параметр или в конфигурационном файле отсутствуют необходимые значения для задания ограничений, алгоритм завершает свою работу, выводя соответствующее диагностическое сообщение.

2. Подсчет количество заказываемых пользователем вычислительных модулей. Заказанные у СУППЗ вычислительные модули будут использованы, как вычислительные модули ПК «Пирамида». После подсчета вычислительных модулей заказа, алгоритм выполняет подсчет количества кластеров, которые будут развернуты, учетом заданных пользователем ограничений. Выполнив подсчет количества кластеров, алгоритм добавляет к заказу дополнительное количество вычислительных модулей, необходимых для выполнения функций серверов кластеров и центрального сервера.

3. Создание файла паспорта задания с учетом того, чтобы генерируемый при запуске задачи файл runmvs.bat проделывал следующее действия:

* сортировка выделенных вычислительных модулей по количеству процессоров в вычислительном модуле;
* создание конфигурационного файла запуска ПК «Пирамида»;
* развертывание ПК «Пирамида» на выделенных вычислительных модулях.

4. Постановка сформированного паспорта в очередь.

Сортировка вычислительных модулей выполняется с той целью, чтобы функции центрального менеджера и менеджеров кластеров выполняли вычислительные модули с меньшим количеством процессоров, по сравнению с остальными. Подобная реализация аргументируется тем, что центральный менеджер и менеджеры кластеров не участвую в вычислениях, т.е. вся нагрузка в процессе работы ПК «Пирамида» приходится на вычислительные модули.

Алгоритм формирования паспорта задания СУППЗ представлен на рисунке 2.3.



Рис. 2.3. Алгоритм формирования паспорта задания СУППЗ

Важной особенностью алгоритма формирования паспорта задания для СУППЗ является то, что пользователь при запуске указывает количество процессоров, которые будут задействованы в ПК «Пирамида» как вычислительные модули. После заказа пользователем определенного количества процессоров, алгоритм подсчитывает количество кластеров, которые возможно построить, с учетом ограничений, указанных в файле epkrun.conf. После подсчета количества кластеров, алгоритм добавляет к заказу пользователя некоторое количество процессоров под менеджеров кластеров и центрального менеджера ПК «Пирамида».

После прохождения очереди задача ставится на счет посредством выполнения файла runmvs.bat. Алгоритм работы формируемого исполняемого файла runmvs.bat показан на рисунке 2.4.



Рис. 2.4. Алгоритм работы runmvs.bat

2.3. Тестирование разработанных программных средств

Все разработанные программные средства реализованы в виде командного файла epkrun и исполняемого файла mkconf\_pyram, вызываемого файлом epkrun.

Формат вызова командного файла epkrun:

epkrun [обязательные\_параметры] [необязательные\_параметры]

Секция обязательные\_параметры включает в себя такие параметры, как:

* np – количество процессоров для счета задачи;
* passp – имя паспорта задачи;
* tsk – имя запускаемой задачи.

Необязательные параметры содержат следующие параметры:

* maxtime – максимальное время счета задачи (в минутах);
* quant – интервал времени в течение которого задача не может быть прервана (в минутах, не должен превышать максимальное время счета).

Порядок следования обязательных и необязательных параметров не имеет значения, однако если один из обязательных параметров не указан, запущенная задача завершится с соответствующим сообщением в стандартном потоке ошибок.

Особое внимание следует уделить параметру np.

Главной особенностью программы epkrun является то, что задаче, при постановке в очередь, выделяется большее количество процессоров по сравнения с тем, что указанно при вызове. Это обусловлено тем, что заказанные пользователем процессоры будут использованы, как вычислительные модули ПК «Пирамида». В этой связи в паспорт задачи помещаются дополнительные процессоры, для обеспечения работы менеджеров кластеров и центрального менеджера.

Тестирование командного файла epkrun, объединяющего все разработанные средства проходят с использованием стратегии «белого ящика». В ходе тестирования на вход командному файлу epkrun подавались как правильные, так и неправильные наборы входных данных, с целью проверить корректность работы по всем веткам.

Тестирование на корректных наборах данных проверяет возможность произвольного следования входных параметров.

Тестирование на ошибочных входных данных предполагает вызов всех обработчиков ошибок, и в настоящей дипломной работе включает в себя:

1. Отсутствие основного файла с конфигурацией.

2. Неизвестный входной параметр .

3. Неверный формат основного конфигурационного файла:

 отсутствие секции;

 отсутствие параметра в секции.

4. Неверное количество процессоров:

 малое количество для создания минимальной конфигурации ПК «Пирамида»;

 большое количество процессоров, не соответствующих ограничениям на количество кластеров и вычислительных модулей.

Весь комплекс тестов проводился на СУППЗ, вычислительной установки rsc4.kiam.ru расположенной в институте прикладной математики им. Келдыша. Пользователь в указанной системе имеет логин trase.

Результаты тестирования на правильных наборах входных данных представлены на рисунках 2.5-2.7.

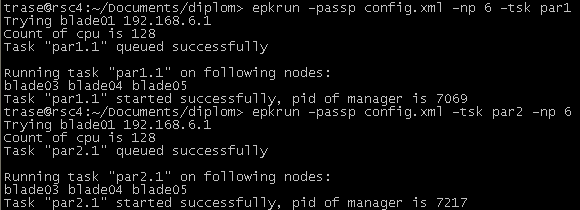


Рис. 2.5. Постановка задач в очередь задач с разным порядком следования входных параметров

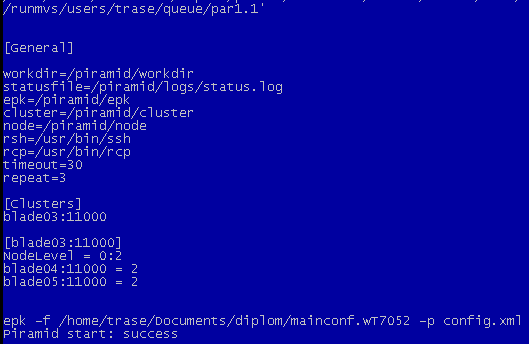


Рис. 2.6. Сформированный конфигурационный файл запуска ПК «Пирамида» задачи par1.1

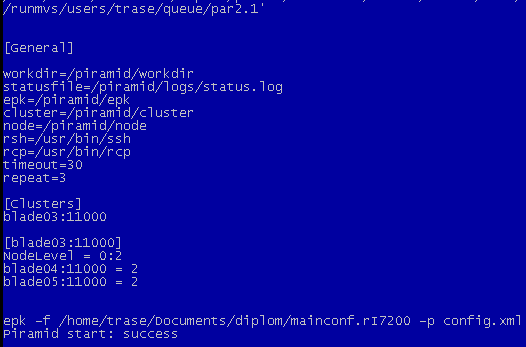


Рис. 2.7. Сформированный конфигурационный файл запуска ПК «Пирамида» задачи par2.1

Следует отметить, что в ходе тестирования запуск ПК «Пирамида» не производился. Задачей тестов было показать возможность прохождения сформированных паспортов через очередь и корректное формирование файлов запуска ПК «Пирамида».

Реакция командного файла epkrun на неверные входные параметры представлена на рисунках 2.8-2.12.

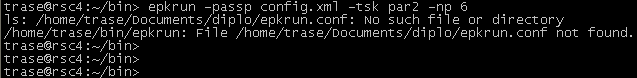


Рис. 2.8. Отсутствие основного конфигурационного файла epkrun.conf



Рис. 2.9. Неизвестный входной параметр

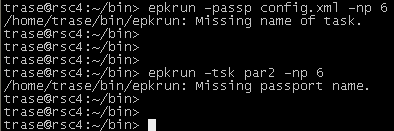


Рис. 2.10. Отсутствие обязательного параметра вызова

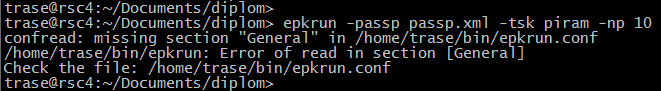


Рис. 2.11. Отсутствие секции в конфигурационном файле epkrun.conf

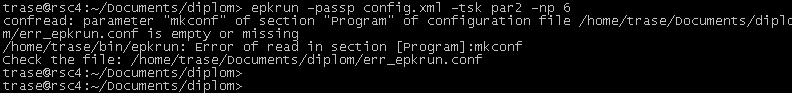


Рис. 2.12. Отсутствие параметра в конфигурационном файле epkrun.conf

При задании количества процессоров, не достаточного для построения минимального дерева ПК «Пирамида» задача будет успешно поставлена в очередь (см. рис. 2.13). Однако при подсчете выделенных вычислительных модулей, пользователю будет выдана ошибка (см. рис. 2.14).

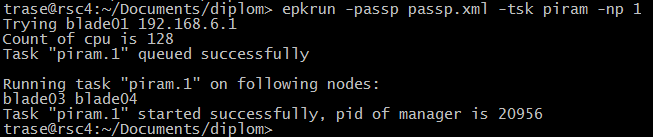


Рис. 2.13. Постановка задачи в очередь



Рис. 2.14. Выдача стандартного потока ошибок

В случае если пользователь в ходе вызова заказывает большое количество процессоров, при малых значениях ограничений на количество кластеров и вычислительных модулей в кластере, система ставит задачу на счет, однако экземпляр Пирамиды не разворачивается. Сделано это для того, чтобы модули, выделенные СУППЗ для счета, не простаивали. Подобная ситуация рассмотрена на рисунках 2.15-2.16.



Рис. 2.15. Постановка задачи в очередь



Рис. 2.16. Сообщение об ошибке

Выводы по второй главе

Автором настоящей дипломной работы были выполнены следующие задачи:

* разработан алгоритм автоматического формирования конфигурационного файла ПК «Пирамида» на основе списка вычислительных, выделенных системой управления прохождением параллельных задач;
* разработаны командные файлы epkrun и runmvs.bat автоматически формирующие паспорт задания СУППЗ, использующего ПК «Пирамида».

Разработанный алгоритм реализован в виде программного средства формирования конфигурационного файла ПК «Пирамида» mkconf\_pyram. Тестирование разработанных программных средств с помощью стратегии «белого ящика» показало корректность работы разработанных программных средств в совокупности.

Глава 3. пк «Пирамида» в составе суппз

3.1. Опытная эксплуатация ПК «Пирамида» в составе СУППЗ

Проведение опытной эксплуатации ПК «Пирамида» в составе СУППЗ, происходило на вычислительном кластере кафедры №732 ИКСИ. Кластер состоит из девяти виртуальных машин – одна управляющая ЭВМ (supz) и восемь вычислительных модулей (vm1-vm8). Каждый вычислительный модуль кластера содержит в себе 8 процессоров, следовательно, в распоряжении пользователей кластера находится 64 процессора. Схема кластера представлена на рис. 3.1.

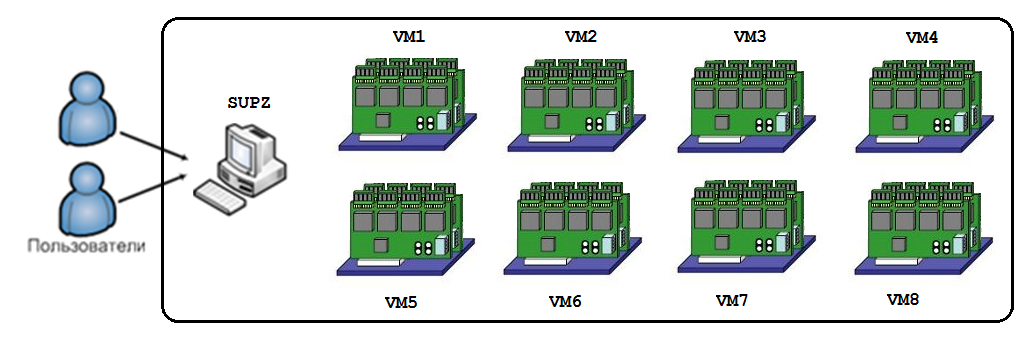


Рис. 3.1. Схема кластера

Все машины находились внутри подсети 192.168.32.Х, причем у каждой виртуальной машины – вычислительного модуля есть возможность взаимодействовать с управляющей машиной макета по протоколу rsh.

Доступ к вычислительному кластеру осуществляется по протоколу ssh, путем использования программного средства SecureCRT.

Запуск задачи на счет производится пользователем с логином user1 из каталога Programs, домашней директории, в которой находится исполняемый файл последовательной программы opp и предназначенный для нее паспорт задания config.xml ПК «Пирамида». Пользователь при этом находится на управляющей машине supz с адресом 192.168.32.160.

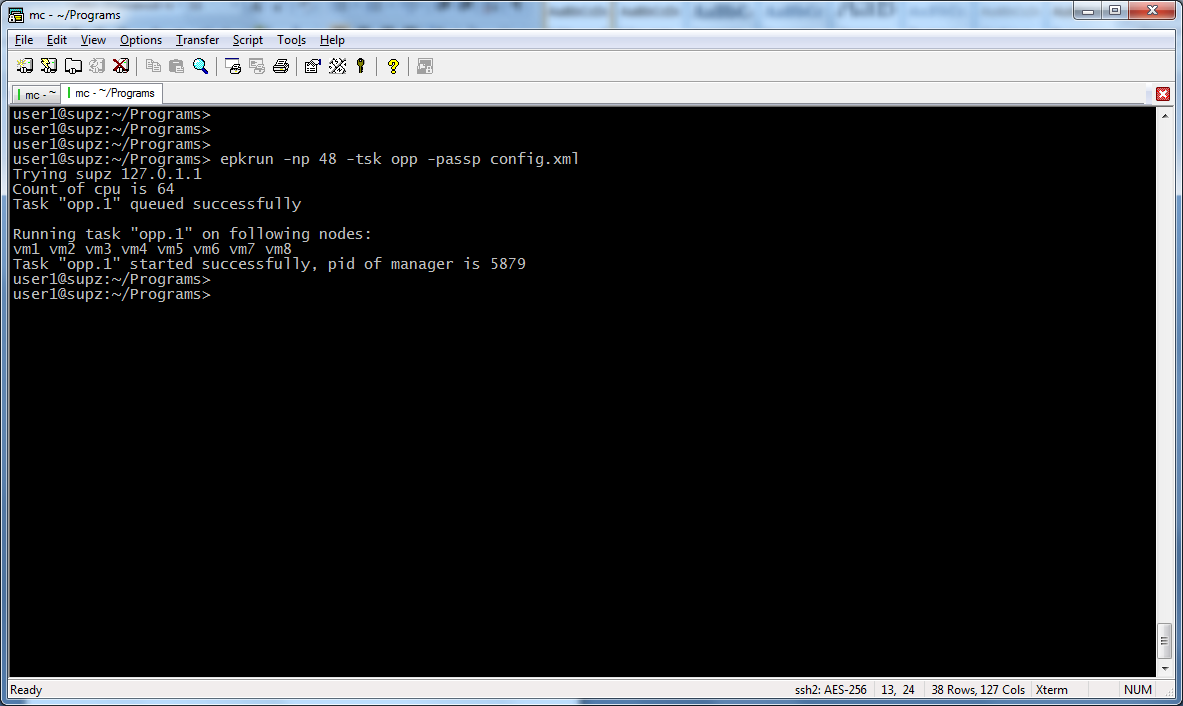


Рис. 3.2. Запуск ПК «Пирамида» через СУППЗ

Состояния очереди до и после запуска задачи представлены на рис 3.3‑3.4. Проверка состояния очереди осуществляется путем введения команды mqinfo.

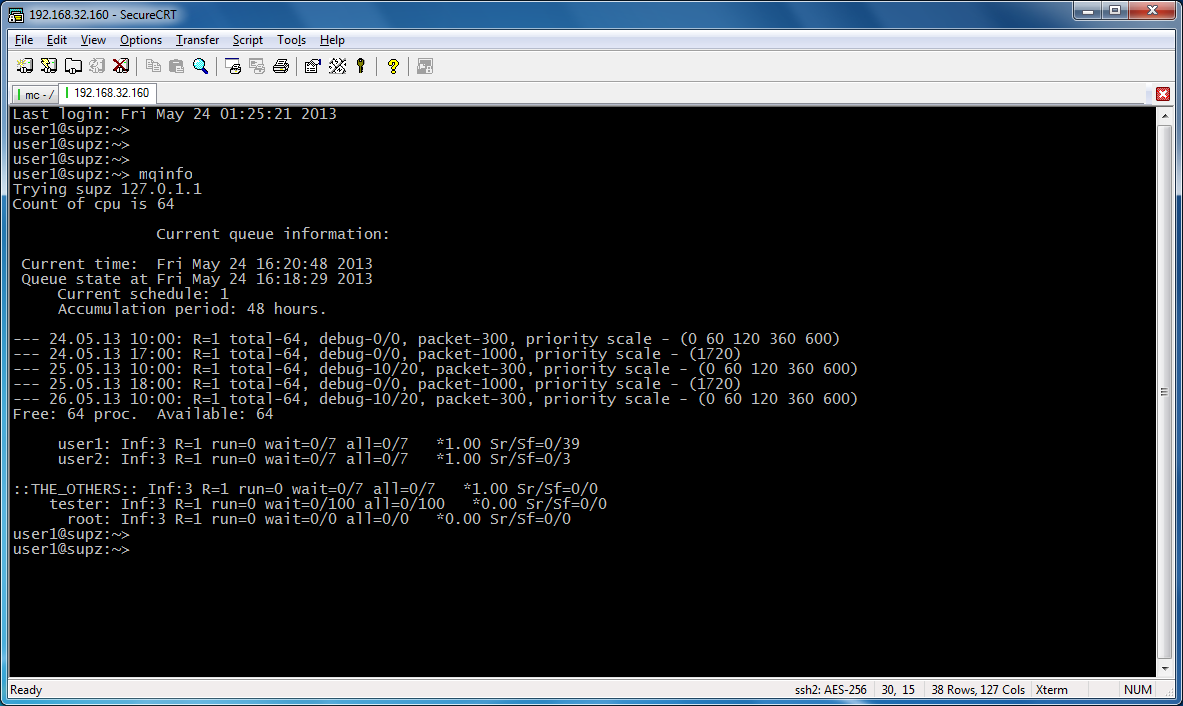


Рис. 3.3. Состояние очереди до запуска задачи

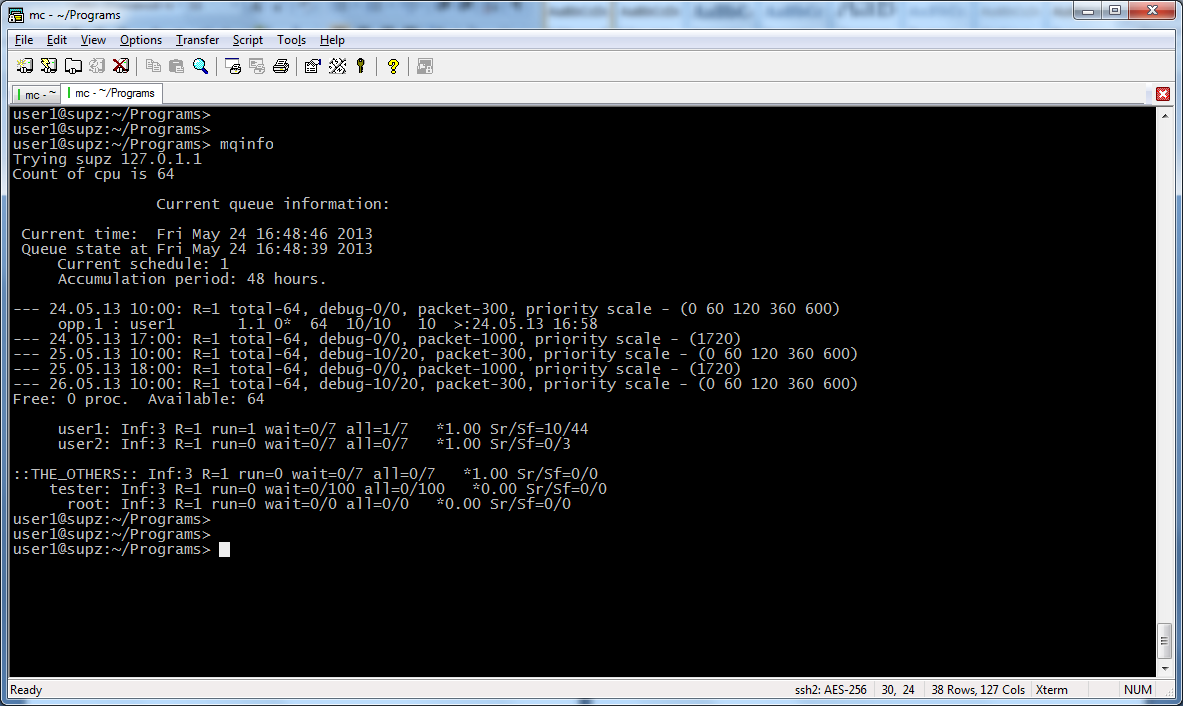


Рис. 3.4. Состояние очереди после запуска задачи

На рис. 3.2 параметру –np, обозначающему количество заказанных процессоров, при запуске задачи, было присвоено значение 48. Однако на рис. 3.4, показано, что у СУППЗ после постановки задачи на счет не остается свободных процессоров. Указанная ситуация объясняется тем что пользователь в процессе запуска указал количество требуемых вычислительных модулей, а при формировании паспорта задания СУППЗ к заказанному вычислительному модулю добавились дополнительные вычислительные ресурсы, для обеспечения функционирования менеджера кластера и центрального менеджера.

К моменту запуска ПК «Пирамида» был сформирован конфигурационный файл запуска с именем mainconf.LPGjFl (рис. 3.5).

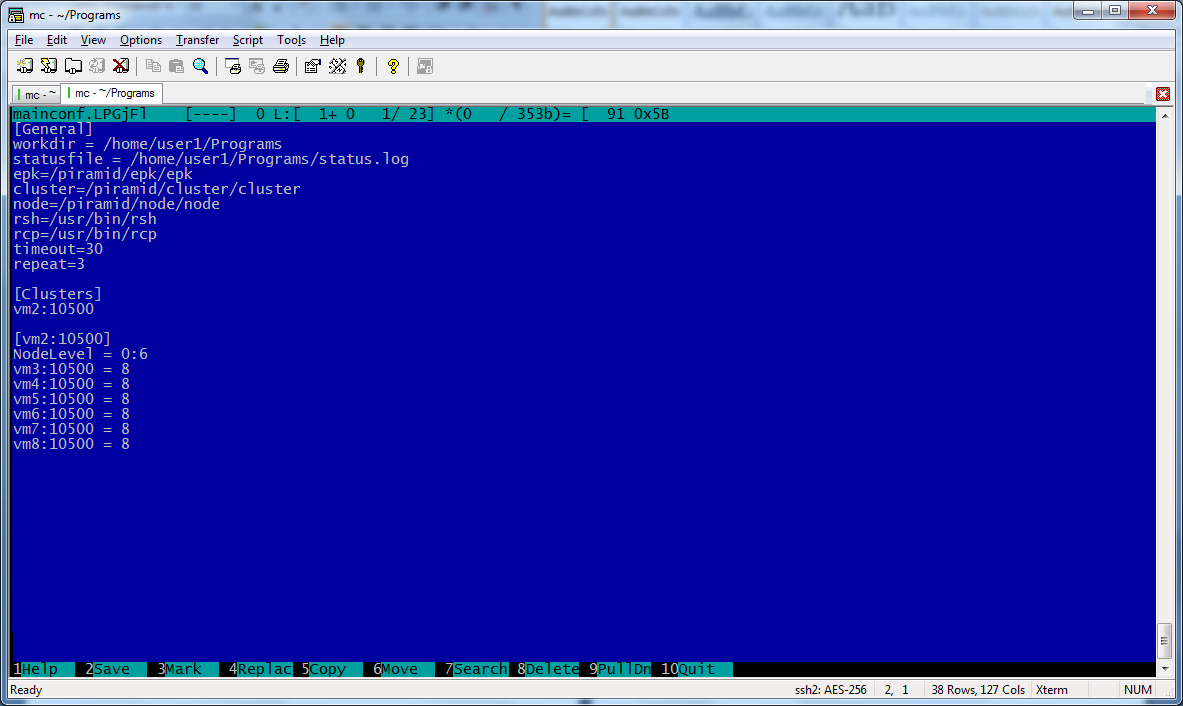


Рис. 3.5. Конфигурационный файл запуска ПК «Пирамида»

В секции Clusters показано, что Пирамида в процессе счета будет иметь один кластер, с менеджером vm2, и шесть вычислительных модулей в кластере – машины vm3-vm8. Рассмотренный конфигурационный файл существует до тех пор пока не завершит работу пока не завершит работу менеджер центрального сервера epk. После того как epk завершает работу, файл удаляется.

В результате работы ПК «Пирамида», в каталог из которого происходил запуск, записываются файлы-журналы работы программного комплекса, файлы с результатами работы, а также отчеты о выявленных ошибках.

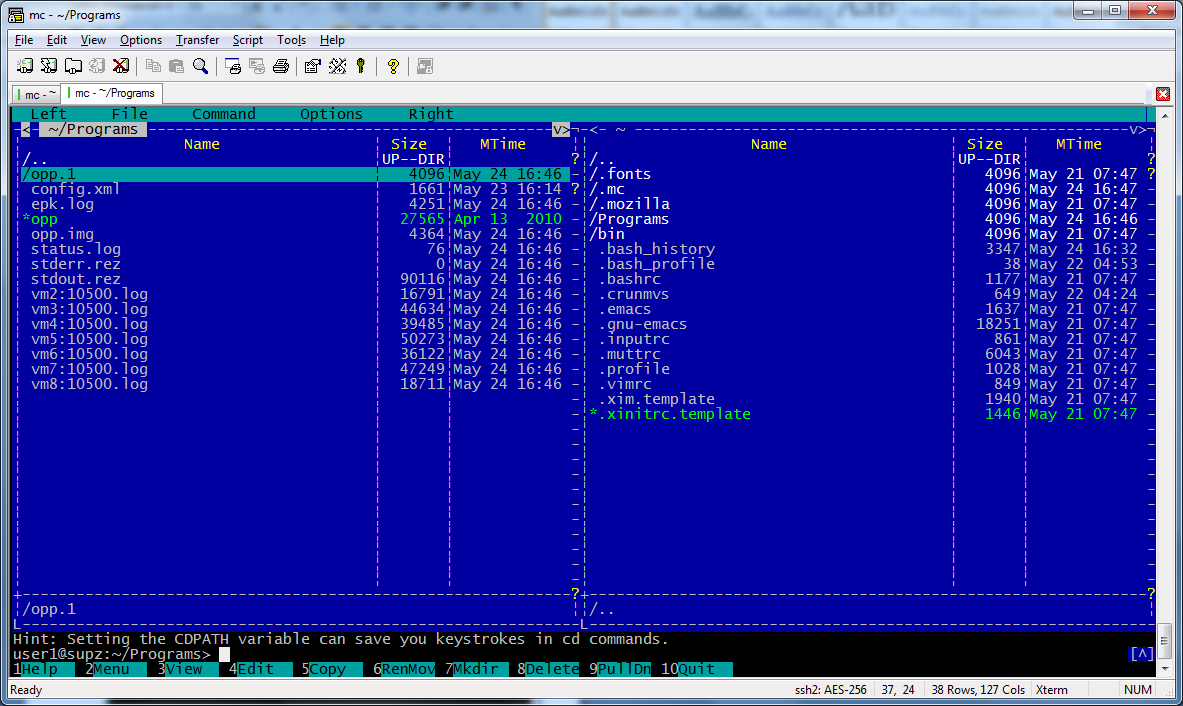


Рис. 3.6. Структура каталога после завершения работы ПК «Пирамида»

В составе СУППЗ, ПК «Пирамида» отображает ход своей работы файл errors, который находится в каталоге имя\_задачи.номер\_задачи, в данном случае в каталоге opp.1.

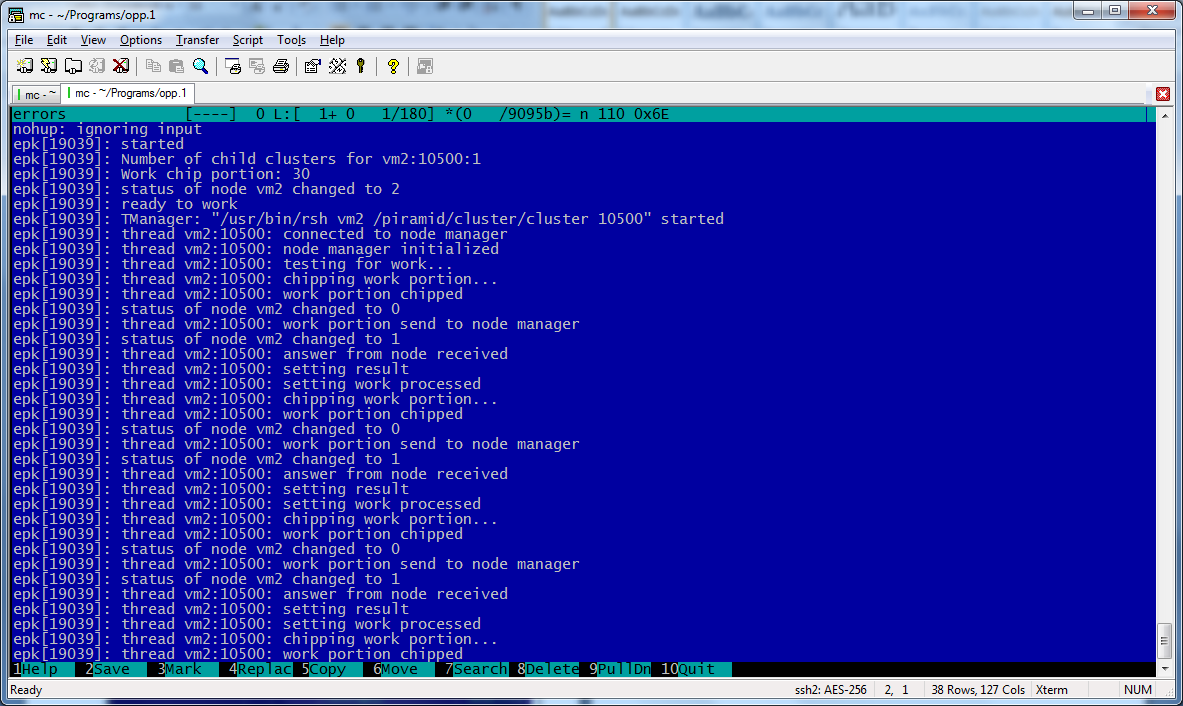


Рис. 3.7. Ход работы ПК «Пирамида»

Результаты счета записываются в итоговом файле результатоы stdout.rez, при этом отображается каждый вызов одной последовательной программы. Более подробно формат отображения результатов рассмотрен в документе «Программный комплекс «Пирамида». Руководство пользователя»[1].

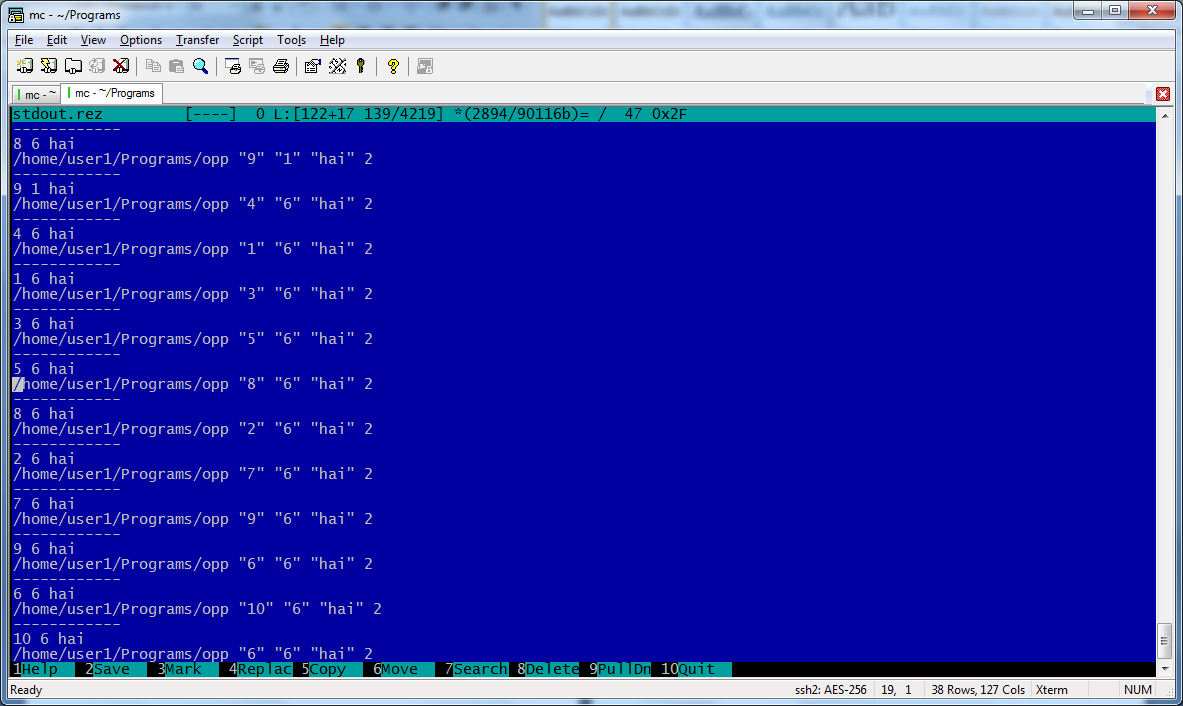


Рис. 3.8. Результаты работы ПК «Пирамида» в составе СУППЗ

При нахождении в системе нескольких пользователей и запуске одним из них задачи в отсутствие вычислительных ресурсов, система поместит запущенную задачу в очередь. После того как ресурсы освободятся, СУППЗ автоматически поставит задачу на счет, развернув для пользователя собственный экземпляр ПК «Пирамида».

Для корректной работы ПК «Пирамида» в составе СУППЗ необходимо чтобы на сервере доступа в каталоге /common была создана директория /pyramid с вложенной директорией /config.

В директории /common/pyramid содержатся файлы epkrun, epk, cluster, node, а также поддиректория /config, в которой находится основной файл с настройками для запуска Пирамиды epkrun.conf (см. приложение 1). Разрешение вносить изменения в файл epkrun.conf должен иметь только администратор ПК «Пирамида».

Помимо этого администратору ПК «Пирамида» необходимо изменить значение переменной CPU\_MULT в файле epkrun. Указанная переменная содержит значение количества процессоров в одном вычислительном модуле и используется при расчете количества кластеров на основе заказанных пользователем вычислительных модулей.

Выводы по третьей главе

Автором настоящей дипломной работы была выполнена задача по проведению опытной эксплуатации разработанных программных средств, в совокупности являющихся интерфейсом взаимодействия программного комплекса «Пирамида» и системы управления прохождением параллельных задач. Исходя из результатов эксплуатации, можно сделать вывод о корректности функционирования разработанного программного интерфейса.

На основе результатов опытной эксплуатации разработанного интерфейса были составлены практические рекомендации администратору ПК «Пирамида» о включении программного комплекса в состав СУППЗ.

Заключение

В настоящей дипломной работе был спроектирован и разработан механизм сопряжения СУППЗ с программным комплексом «Пирамида». Объем выполненной работы включил в себя установку и настройку ПК «Пирамида» и СУППЗ, разработку командных файлов формирования паспорта задания, а также программу формирования конфигурационный файл запуска ПК «Пирамида».

По окончании разработки упомянутых программных средств было произведено их тестирование на вычислительной установке rsc4.kiam.ru с использование стратегии «белого ящика».

Реализованные программные средства позволяют пользователю, не обладающему навыками параллельного программирования, ускорять время разработки параллельных программ и организовывать вычисления с распараллеливанием по данным.

В рамках настоящей дипломной работы автором была проведения опытная эксплуатация разработанного программного интерфейса с использованием вычислительных ресурсов кластера кафедры №732 ИКСИ.

Дальнейшие перспективы работы подразумевают создание программных средств автоматического формирования xml-паспорта задания для ПК «Пирамида», что позволит оставить за пользователем только разработку программы для проведения вычислений.

Литература

1. Программный комплекс «Пирамида». Руководство пользователя. 2010. 35 с.
2. Руководство пользователя системы МВС-100К. 26 с.
3. Руководство системного программиста (администратора) системы управления прохождением задач МВС-5000. М:. ИПМ им. Келдыша РАН. 2002.
4. Даниэл Дж. Баррет. Linux. Основные команды. Карманный справочник. М:. Изд-во КУДИЦ-ОБРАЗ. Москва, 2007.

приложение 1. Конфигурационный файл epkrun.conf

[General]

# Обязательная секция с именем General

# Параметр statusfile определяет

# имя файла статуса выичслительных модулей,

# в котором содержится информация

# о текущем статусе вычислительных модулей

statusfile = /common/epk/nodesstatus.log

# Путь к исполняемому файлу центрального менеджера (epk):

epk = /piramid/epk

# Путь к исполняемому файлу менеджера кластера (cluster):

cluster = /piramid/cluster

# Путь к исполняемому файлу менеджера модуля (node):

node = /piramid/node

# Параметр rsh определяет

# путь к исполняемому файлу подсистемы

# удаленного выполнения команд

rsh = /usr/bin/ssh

# Параметр timeout определяет время системного

# таймаута (в секундах). Системный таймаут используется при

# соединении и взаимодействии менеджеров ПК «ПИРАМИДА»

#соседних уровней иерархии. Определяет время ожидания менеджера

# верхнего уровня ответа (например, менеджера кластера)

# от менеджера нижнего уровня (например, менеджера ВМ)

timeout = 300

# Параметр repeat определяет число повторных

# попыток восстановления связи между менеджерами

# ПК«ПИРАМИДА» соседних уровней иерархии.

# Каждая повторная попытка

# восстановления связи предпринимается через вермя,

# определяемое системным таймаутом. По истечении числа

# попыток кластер или ВМ, обслуживаемый не отвечающим

# менеджером нижнего уровня, помечается как неисправный

[Program]

# Обязательная секция с именем Program

# содержит в себе имя программы формирующей иерархическую

# структуру ПК «Пирамида» на основе списка выделенных СУППЗ

# вычислительных

mkconf = /pyramid/mkconf\_pyram

[Piramid]

# Обязательная секция с именем Piramid

# содержит в себе имя численные значения, задающие ограничения

# на количество кластеров и вычислительных модулей в кластере

Clusters\_limit = 8

Nodes\_limit = 8