Приложение Е

МИНОБРНАУКИ РОССИИ
ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО УЧРЕЖДЕНИЯ «ФЕДЕРАЛЬНЫЙ НАУЧНЫЙ ЦЕНТР НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ
ИНСТИТУТ СИСТЕМНЫХ ИССЛЕДОВАНИЙ РОССИЙСКОЙ АКАДЕМИИ НАУК»
Межведомственный суперкомпьютерный центр
Российской академии наук
(МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН)

УТВЕРЖДАЮ
Директор МСЦ РАН – руководитель ЦКП вычислительными ресурсами
МСЦ РАН – филиала ФГУ ФНЦ НИИСИ РАН

Б.М. Шабанов
«30» 09 2020 г.

Описание услуги

«Создание услуги по высокопроизводительным параллельным вычислениям с распараллеливанием по данным»

Описание услуги рассмотрено и одобрено на заседании Ученого совета МСЦ РАН – секции Ученого совета ФГУ ФНЦ НИИСИ РАН
«17» 09 2020 г., протокол № 4

Москва 2020

Общие положения

Среди различных прикладных задач, решаемых с помощью суперкомпьютеров, важный класс составляют задачи распараллеливания по данным, при котором одна и та же последовательность вычислений (прикладной алгоритм) выполняется над всеми элементами множества (пула) входных данных. При организации параллельных вычислений с распараллеливанием по данным существенным является то, что между процессами параллельной программы отсутствуют информационные обмены. При этом достаточно часто вычислительный алгоритм может быть реализован в виде одной последовательной программы (далее – ОПП), для которой порция входных данных определяется значениями одного или нескольких параметров. Из таких порций вычислительной работы складывается пул входных данных для решения прикладной задачи, который в итоге определяется множеством всех возможных значений параметров ОПП во всех их комбинациях.

Для организации параллельных вычислений с распараллеливанием по данным может быть применен программный комплекс (далее – ПК) «Пирамида», предназначенный для функционирования на вычислительной установке с иерархической структурой. В составе вычислительной установки выделяют центральный сервер, серверы управления кластером и вычислительные модули в составе кластеров. ПК «Пирамида» изначально создавался с целью освобождения прикладного пользователя от решения задачи организации параллельных вычислений. Прикладной вычислительный алгоритм реализуется в виде ОПП, а за запуск множества экземпляров ОПП на вычислительных модулях и распределение вычислительной работы целиком отвечает ПК «Пирамида». ПК «Пирамида» с задаваемым периодом автоматически сохраняет контрольные точки, что позволяет восстановить вычисления при перезапуске вычислительной задачи.

Для функционирования ПК «Пирамида» необходимо формализованное описание иерархической структуры вычислительной установки. В режиме коллективного пользования суперкомпьютером пользователь оформляет прикладные расчеты в виде вычислительного задания, включающего расчетную программу, требования к объему ресурсов и времени выполнения, а также исходные данные. Ресурсы для поступивших заданий выделяются динамически из числа свободных на момент запуска задания на выполнение. Для возможности применять ПК «Пирамида» в системе коллективного пользования суперкомпьютером необходима разработка программной подсистемы автоматического формирования иерархической структуры динамически выделяемых заданиям ресурсов и формализованного описания этой структуры для ПК «Пирамида». Указанная программная подсистема лежит в основе созданной услуги по высокопроизводительным параллельным вычислениям с распараллеливанием по данным.

Состав услуги

В состав услуги по высокопроизводительным параллельным вычислениям с распараллеливанием по данным включаются:

- Предоставление пользователям новой клиентской команды Системы управления прохождением параллельных заданий (далее – СУППЗ), указанная команда автоматически формирует вычислительное задание СУППЗ для запуска ОПП через ПК «Пирамида».
- Прохождение сформированного задания через очередь СУППЗ, запуск задания на динамически выделенных ресурсах суперкомпьютера с автоматическим формированием описания этих ресурсов для ПК «Пирамида».
- Автоматическое разворачивание ПК «Пирамида» на выделенных заданию ресурсах и осуществление высокопроизводительных вычислений с распараллеливанием по данным с помощью ПК «Пирамида».

Порядок оказания услуги

Услуга доступна любому пользователю, зарегистрированному в ЦКП вычислительными ресурсами МСЦ РАН – филиала ФГУ ФНЦ НИИСИ РАН. Для того чтобы воспользоваться услугой, пользователь должен:

- разработать ОПП в соответствии с требованиями пп. 2.1.5 – 2.1.11 Руководства программиста ПК «Пирамида» (приложение № 1);
- подготовить паспорт задания для ПК «Пирамида» в соответствии с п. 4.1.1 Руководства программиста ПК «Пирамида» (приложение № 1);
- запустить задание в соответствии с п. 3.3 Руководства программиста ПК «Пирамида» (приложение № 1).

Приложение № 1
к описанию услуги
«Создание услуги по
высокопроизводительным
параллельным зычислениям
с распараллеливанием по данным»

ПРОГРАММНЫЙ КОМПЛЕКС ОРГАНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ С РАСПАРАЛЛЕЛИВАНИЕМ ПО ДАННЫМ «ПИРАМИДА» (ПК «Пирамида»)

Руководство программиста

Листов 43

Аннотация

Документ содержит сведения, необходимые для программистов, использующих программный комплекс организации параллельных вычислений модернизированный (ПК «Пирамида»). ПК «Пирамида» предназначен для обеспечения распределения вычислительной работы при решении задач в иерархически организованных вычислительных системах (ИОВС).

В разделе 1 представлены сведения о назначении, составе и принципах функционирования ПК «Пирамида». Приведена информация, необходимая для понимания функций ПК «Пирамида» и условия его применения.

Раздел 2 содержит сведения о характеристиках программного изделия, используемой модели организации вычислительного процесса.

Раздел 3 содержит информацию о формате запуска ПК «Пирамида».

Раздел 4 содержит информацию о входных и выходных данных программного изделия, правилах передачи параметров в пользовательскую программу и возврата значений и результатов из нее.

Раздел 5 содержит описание формата сообщений ПК «Пирамида» программисту.

В приложении А приведены тестовые примеры для проверки работоспособности ПК «Пирамида».

Содержание

1 Назначение и условия применения ПК «Пирамида»

1.1 Назначение ПК «Пирамида»

1.1.1 Программный комплекс организации параллельных вычислений модернизированный (ПК «Пирамида») предназначен для функционирования на вычислительной установке, структура которой представлена на рисунке 1.

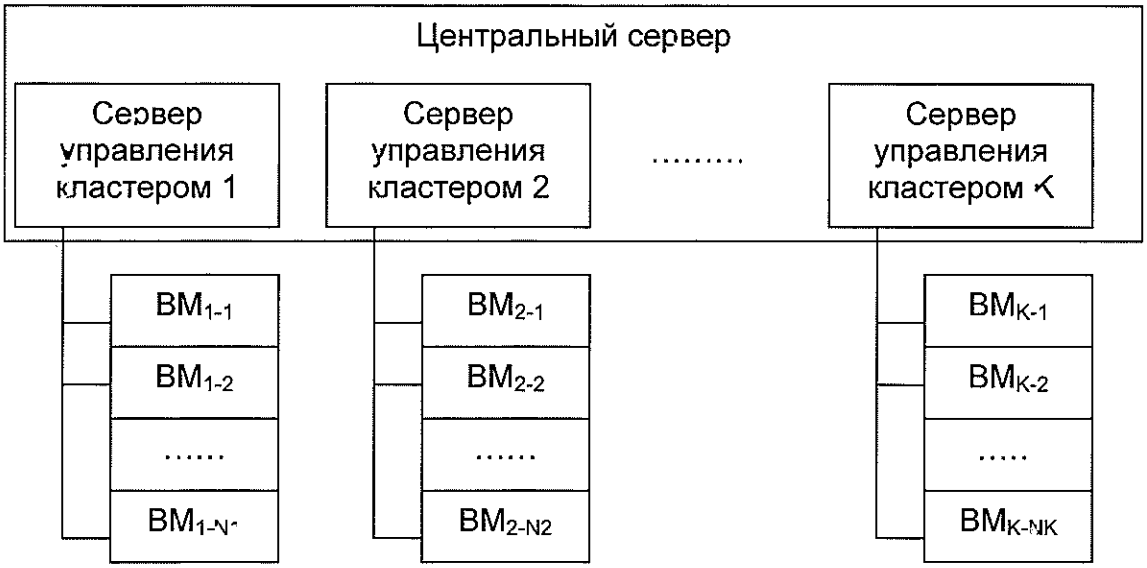

Рисунок 1 – Структура вычислительной установки

Вычислительная установка состоит из вычислительных модулей, управляемых центральным сервером управления. Логически все вычислительные модули могут быть поделены на кластеры, каждым кластером управляет виртуальный (программный) сервер управления кластером в составе центрального сервера управления, причем i-й кластер состоит из N_i вычислительных модулей ($ВМ_{i-1}$ – $ВМ_{i-Ni}$).

1.1.2 Рассмотрим схему распараллеливания, на которую будет распространяться принятая в ПК «Пирамида» организация параллельных вычислений. В ее основе лежит схема «распараллеливания по данным». Пользователь может реализовать вычислительный алгоритм в виде одной последовательной программы (ОПП). ОПП принимает на вход значения параметров и обрабатывает соответствующую порцию данных. Предполагается, что вся необходимая для вычислений информация известна заранее, и межмодульное взаимодействие не требуется.

1.1.3 Исполняемый программный модуль ОПП представляется в виде однопотокового или многопотокового приложения (число потоков в ОПП может соответствовать числу процессорных ядер в вычислительном модуле).

1.1.4 На каждом вычислительном модуле выполняются один или несколько исполняемых модулей одной и той же 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.

Применяемый в ПК «Пирамида» подход позволяет существенно сократить время реализации алгоритмов и трудозатраты на их распараллеливание. ПК «Пирамида» предоставляет пользователю инструмент для описания законов формирования параметров, обеспечивает бесперебойное выполнение его программы на всех доступных вычислительных модулях системы и сбор результатов вычислений.

Конкретное значение числа запускаемых исполняемых модулей (экземпляров) ОПП на каждом ВМ определяет программист, использующий ПК «Пирамида».

1.1.5 ПК «Пирамида» задействует ресурсы вычислительного комплекса, распределяя вычислительную работу между множеством экземпляров ОПП, каждый из которых выполняет вычисления с определенным набором векторов параметров.

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

- разделение работы между экземплярами ОПП путем перебора всех возможных комбинаций параметров;
- раздача работы экземплярам ОПП путем передачи параметров и получение результатов расчетов от выполненных экземпляров ОПП;
- мониторинг исправности вычислительных ресурсов.

1.1.7 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. В случае обна-

ружения недоступности одного или нескольких ВМ ПК «Пирамида» автоматически начинает перераспределять вычислительную работу между исправными (доступными) ВМ.

После восстановления доступности одного или нескольких ВМ ПК «Пирамида» автоматически вновь включает их в состав решающего поля и начинает распределять на ставшие доступными ВМ вычислительную работу.

1.1.8 Распределение вычислительной работы между ВМ осуществляется порциями, размер которых задает программист, использующий ПК «Пирамида». В гетерогенной вычислительной системе ПК «Пирамида» осуществляет автоматическую балансировку вычислительной нагрузки между ВМ различной производительности. На более производительные ВМ будет распределяться большее число порций вычислительной работы, чем на менее производительные.

1.2 Состав ПК «Пирамида»

1.2.1 Программный комплекс организации параллельных вычислений (ПК «Пирамида») с распараллеливанием по данным представляет собой иерархическую систему менеджеров, показанную на рисунке 2. В состав системы входят:

- центральный менеджер (ЦМ), функционирующий на центральном сервере управления;
- менеджеры кластеров (МК), реализующие функции серверов управления кластерами;
- менеджеры вычислительных модулей (ММ), функционирующие на вычислительных модулях (ВМ).

1.2.2 Менеджер кластера на каждом ВМ запускает менеджеры ВМ (ММ).

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 экземпляра ОПП. Код завершения ОПП и результаты передаются в менеджер кластера. Если через

время таймаута испсляемый модуль ОПП не завершается, менеджер ВМ принудительно завершает его работу.

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

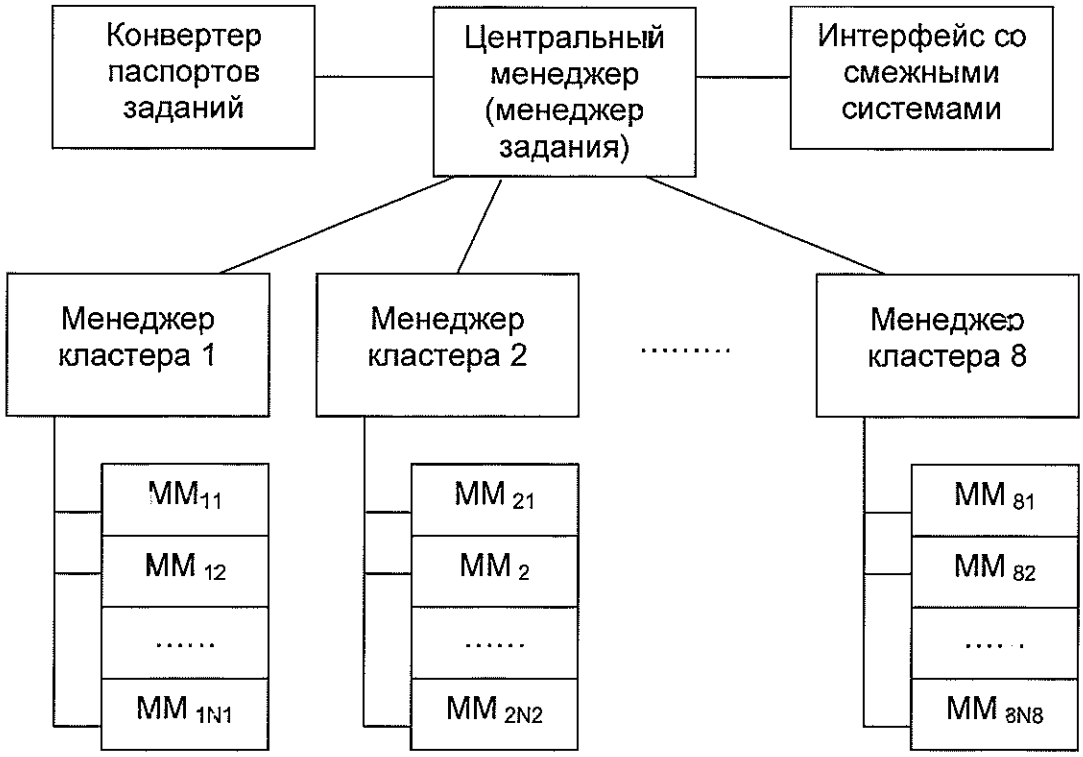

Рисунок 2 – Иерархическая структура ПК «Пирамида»

1.2.3 Функции центрального менеджера реализует программный модуль с именем `epk`; функции менеджера кластера – программный модуль с именем `cluster`; функции менеджера ВМ – программный модуль с именем `node`.

2 Характеристики ПК «Пирамида»

2.1 Организация вычислительного процесса в ПК «Пирамида»

2.1.1 Инициализация работы менеджеров ПК производится путем запуска менеджера задания на центральном сервере. На вход менеджера задания подается описание ресурсов ПК «Пирамида» и паспорт задания.

Менеджер задания порождает менеджеров кластеров, передавая им на вход описание ресурсов и паспорт задания. Менеджеры кластеров в свою очередь порождают менеджеров ВМ или менеджеров подчиненных кластеров.

2.1.2 Описание ресурсов вычислительной установки представляет собой конфигурационный файл в формате ini и содержит иерархически упорядоченный список вычислительных модулей с указанием их типа и характеристик. Подробно формат конфигурационного файла с описанием ресурсов вычислительной установки рассмотрен в 4.1.1.

2.1.3 Паспорт задания содержит описание одной последовательной программы (ОПП), ее параметров и условий выполнения. Паспорт задания представляет собой файл в формате ini.

2.1.4 Для каждого типа вычислительного ресурса паспорт задания может содержать т.н. профиль запуска, задаваемый в элементе `profiles` паспорта задания. Профилем запуска определяются особенности работы ПК «Пирамида» с типом вычислительного ресурса, соответствующим этому профилю.

2.1.5 Одна последовательная программа (ОПП), реализующая прикладную логику, разрабатывается пользователем-программистом для запуска через ПК «Пирамида». Исполняемые программные модули ОПП могут быть представлены в нескольких вариантах, причем для каждого варианта в паспорте задания должен быть предусмотрен отдельный профиль запуска. Имя профиля запуска всегда должно совпадать с соответствующим именем типа вычислительного ресурса в файле описания ресурсов ПК «Пирамида». Допустимые варианты исполняемых модулей ОПП и профилей их запуска следующие:

- в виде многопотокового консольного приложения для универсальной ЭВМ (профиль запуска с именем x86);

- в виде многопотокового консольного приложения, использующего графический ускоритель (профиль запуска с именем GPU);

- в ином определенном пользователем виде (профиль запуска с заданным пользователем именем).

Перечисленные варианты могут применяться одновременно, если вычислительная установка содержит гетерогенные вычислительные модули.

2.1.6 Текущим для каждого экземпляра ОПП при его запуске будет рабочий каталог задания, имя которого задается элементом `directory` паспорта задания. Все файлы, порожденные за время выполнения экземпляра ОПП в рабочем каталоге задания, считаются результатом работы и будут автоматически отправлены на центральный сервер.

2.1.7 Входные данные передаются ПК «Пирамида» в ОПП через параметры ОПП. Параметры могут быть числовыми или строковыми и характеризуются своими значениями. Для числового параметра значением является целое число, для строкового – строка. Для каждого из параметров ОПП задается определенный диапазон значений, т.е. множество всех возможных значений, которые может принимать параметр.

2.1.8 Для числового параметра диапазон задается в виде тройки чисел – начальное значение, конечное значение, шаг перебора. Например, тройка {12, 21, 3} определит множество значений, состоящее из чисел 12, 15, 18 и 21.

Три числа, определяющие диапазон значений числового параметра (начальное значение, конечное значение, шаг перебора), будут переданы в ОПП как один аргумент из трех полей, разделенных пробелами. При вызове экземпляра ОПП аргумент числового параметра будет заключен в кавычки. Например, диапазон {30, 50, 2} будет представлен аргументом "30 50 2".

Если числовой параметр примет одиночное значение (начальное значение будет совпадать с конечным при шаге перебора 1), то он будет передан в ОПП как одно число, без указания конечного значения и шага перебора.

2.1.9 Строковый параметр может быть задан двумя способами – в виде списка строк или в виде файла со списком строк.

Значение строкового параметра ОПП представляет собой список строк, разделяемых пробелами. Список передается в ОПП как один аргумент, для чего при вызове ОПП список заключается в двойные кавычки.

ВНИМАНИЕ! При формировании входного набора строк для ОПП пользователь должен исключить появление в строках символов «пробел» и «двойная кавычка». Рекомендуется заменить эти символы заранее определенной комбинацией других символов. Разбор заданной комбинации должен осуществляться экземпляром ОПП, т.е. должен быть предусмотрен пользователем-программистом.

Файл со списком строк должен располагаться в общей для всех ВМ и центрального сервера файловой системе.

2.1.10 Каждая команда запуска исполняемого модуля (экземпляра) ОПП формируется на основе шаблона команды запуска, заданного в элементе `commandTemplate` паспорта задания в соответствующем профиле запуска. Для команды запуска экземпляра ОПП перечисляется все множество наборов параметров, определяемое элементом `parameters` (множество наборов параметров представляет собой прямое произведение множеств, определяемых всеми элементами `parameter`).

При формировании команды запуска экземпляра ОПП производится замена макроподстановок из шаблона `commandTemplate` на значения из набора параметров в соответствии с именами макроподстановок.

Кроме макроподстановок параметров в шаблоне `commandTemplate` может быть использована макроподстановка `${THREADS_COUNT}`, обозначающая число потоков в ядре реконфигурируемого вычислителя (универсального процессора). При формировании команды запуска экземпляра ОПП на конкретном ВМ вместо `${THREADS_COUNT}` будет подставлено число потоков, указанное для данного ВМ в конфигурационном файле описания вычислительных ресурсов.

2.1.11 Каждый запуск исполняемого модуля (экземпляра) ОПП на ВМ завершается определенным результатом. Результатом запуска экземпляра ОПП являются:

- код возврата исполняемого модуля ОПП;

- порожденные стандартные потоки вывода и ошибок исполняемого модуля ОПП;

- созданные в рабочем каталоге за время выполнения экземпляра ОПП файлы.

Каждый результат анализируется компонентами ПК «Пирамида». Код возврата может быть:

- 0 – нормальное завершение ОПП; результат – в стандартном потоке вывода;

- результирующий – означает, что запуск экземпляра ОПП завершился практически значимым результатом;

- терминирующий;

- остальные значения – аварийное (ошибочное) завершение ОПП; диагностика – в стандартном потоке ошибок.

При обнаружении терминирующего кода возврата (означающего, например, успешное решение задачи ОПП) ПК «Пирамида» завершает выполнение.

2.1.12 Время выполнения ОПП контролируется менеджером ВМ и не может превышать значения, заданного атрибутом `maxTime` паспорта задания. Данный атрибут указывается для каждого профиля запуска. При превышении значения времени `maxTime` экземпляр ОПП принудительно завершается и формируется аварийный код возврата. Если код возврата аварийный (по причине истечения времени или по другой причине), то осуществляется попытка перезапуска экземпляра ОПП с теми же параметрами. Число попыток перезапуска задается в паспорте задания атрибутом `maxErrors` элемента соответствующего профиля запуска. При превышении указанного числа попыток порция данных помечается как «ошибочная» и исключается из дальнейшей вычислительной работы.

2.1.13 При любом значении кода возврата файлы, порожденные экземпляром ОПП в рабочем каталоге задания (включая стандартные потоки вывода и ошибок), пересылаются на центральный сервер управления. При этом для каждого результата отмечается, на каком наборе параметров ОПП он получен.

2.1.14 На центральном сервере управления ПК «Пирамида» осуществляет периодическое сохранение контрольной точки. Интервал создания контрольной точки задается в секундах в паспорте задания, элемент `checkpoint`. Контрольная точка

формируется в одном из двух подкаталогов рабочего каталога ПК «Пирамида» с именами `chp1` и `chp2`. На текущую контрольную точку указывает символическая ссылка с именем `work`, также расположенная в рабочем каталоге ПК «Пирамида».

Работа ПК «Пирамида» может быть прервана в любой момент времени и возобновлена с последней созданной контрольной точки. Будьте внимательны при запуске ПК «Пирамида» с новым паспортом задания в том же рабочем каталоге. Для нового запуска (или повторения предыдущего запуска с начала) необходимо удалить из рабочего каталога подкаталоги `chp1` и `chp2`, а также символическую ссылку `work`.

2.2 Распределение вычислительной работы в ПК «Пирамида»

2.2.1 Слайсом вычислительной работы будем называть совокупность диапазонов значений параметров ОПП. Слайс может быть задан различными способами, например, в виде текстового файла следующего формата. Каждая строка файла соответствует одному параметру (1-я строка – 1-му параметру, 2-я – 2-му, и т.д.). Для числового параметра строка состоит из трех чисел – начального значения, конечного значения и шага перебора. Для строкового параметра – список строк или имя файла со списком строк. Число возможных значений параметров, задаваемых слайсом, назовем размером слайса или размером вычислительной работы.

2.2.2 Рассмотрим пример файла слайса для трех параметров:

```
str1 str2 str3
18 23 2
1 3 1
```

Размер слайса из приведенного примера равен 54 (3 значения по первому параметру, 6 значений по второму и 3 значения по третьему).

2.2.3 Слайс, в совокупности с паспортом задания, полностью определяет порцию вычислительной работы. Слайс с единичным числом значений по каждому параметру называется элементарной работой. Другими словами, элементарная работа – это слайс размером 1.

2.2.4 Нетрудно видеть, что вычислительная работа произвольного размера не может быть представлена единичным слайсом. Допустим, слайс из примера 2.2.2 необходимо разделить на два множества значений параметров – размером 10 и 44 соответственно. Для описания как первого, так и второго множеств придется использовать более одного слайса. Приведем пример:

```
Представление работы размером 10:
Слайс 1 (размером 9):
Str1 str2 str3
18 22 2
1 1 1
Слайс 2 (размером 1):
Str1
24 24 2
1 1 1
Представление оставшейся работы размером 44:
Слайс 1 (размером 2):
str2 str3
24 24 2
1 1 1
Слайс 2 (размером 6):
Str1 str2 str3
26 28 2
1 1 1
Слайс 3 (размером 36):
Str1 str2 str3
18 28 2
2 3 1
```

2.2.5 Вычислительной работой будем называть совокупность слайсов, определяющую множество значений параметров произвольного размера. Нетрудно видеть, что максимальное число слайсов в одной вычислительной работе не может быть более числа параметров.

2.2.6 Заметим, что ОПП может быть запущена либо с элементарной вычислительной работой, либо с вычислительной работой, представленной единственным

слайсом (когда каждому параметру соответствует только один диапазон). Поэтому по отношению к ОПП употребляется термин «максимальный размер вычислительной работы» для ОПП, определяющий максимальнс возможный размер слайса для однократного запуска ОПП. Например, если максимальный размер вычислительной работы для ОПП задан 44, то оставшаяся работа из примера п. 2.2.4 мсжет быть выполнена посредством трех запусков ОПП с каждым из трех слайсов размерами 2, 6 и 36 соответственно.

2.2.7 ПК «Пирамида» содержит функцию деления произвольной вычислительной работы на вычислительные работы произвольного размера. В частности, работа из примера 2.2.2 с помощью указанной функции может быть поделена на две работы из примера 2.2.4. Функция деления работы является основой логики функционирования ПК «Пирамида», определяемой тремя параметрами-размерами вычислительной работы, задаваемым в элементе `portions` паспорта задания:

- размер вычислительной работы для кластера (атрибут `cluster`). Вычислительная работа размером `cluster` выделяется центральным менеджером каждому менеджеру кластера;

- размер вычислительной работы для ВМ (атрибут `node`). Вычислительная работа размером `node` выделяется менеджером кластера каждому менеджеру ВМ;

- размер вычислительной работы для ОПП (атрибут `program`). Вычислительная работа размером `program` выделяется менеджером ВМ каждой ОПП. Обращаем внимание, что `program` является максимально возможным размером вычислительной работы для ОПП. Реальный размер вычислительной работы, передаваемой в ОПП, зависит от размеров составляющих работу слайсов и может быть меньше `program`.

2.2.8 Общая логика функционирования ПК «Пирамида» определяется следующим образом. Центральный менеджер после инициализации формирует общую вычислительную работу, содержащую все множество значений параметров, которые необходимо перебрать.

С помощью функции деления центральный менеджер отделяет от общей работы порции вычислительной работы размером, равные размеру работы для кластера (`cluster`). Отделенные порции передаются менеджерам кластеров.

Менеджеры кластеров производят дальнейшее деление полученной работы на порции размером, равным размеру работы для ВМ (`node`) и передают отделенные порции своим менеджерам ВМ. Аналогичным образом менеджеры ВМ производят деление работы на порции размером, равным максимальному размеру работы для ОПП (`program`). Далее каждый менеджер ВМ производит запуск экземпляров ОПП со слайсами из выделенных порций вычислительной работы.

2.2.9 ПК «Пирамида» способен работать в гетерогенной и ненадежной вычислительной среде. Алгоритмы работы менеджеров всех уровней рассчитаны на то, что подчиненные менеджеры могут выполнять выделенную вычислительную работу с разной скоростью или выйти из строя в любой момент времени. Для обеспечения надежности вычислений используется следующий подход.

Получив вычислительную работу, менеджер начинает ее делить на порции, соответствующие размеру своего уровня (`cluster`, `node`, `program`). Выделенные порции передаются подчиненным менеджерам. От каждого подчиненного менеджера при этом ожидается результат выполнения работы. При получении результата подчиненному менеджеру выделяется и передается новая порция вычислительной работы.

В некоторый момент возникает ситуация, когда каждый подчиненный менеджер находится в состоянии выполнения своей части работы, но при этом исходная вычислительная работа исчерпана (подчиненные менеджеры досчитывают последние порции работы). В этом случае при завершении работы какого-либо подчиненного менеджера происходит перераспределение оставшихся порций вычислительной работы.

Завершившемуся ранее других подчиненному менеджеру передается порция вычислительной работы одного из не закончивших выполнение подчиненных менеджеров. Таким образом обеспечивается надежность вычислений – если какой-либо подчиненный менеджер вышел из строя, его порция вычислительной работы

будет гарантированно выполнена другим менеджером. При этом исключается дублирование выполняемой вычислительной работы.

2.2.10 При такой организации надежность и масштабируемость достигаются автоматически. Если какой-либо вычислительный модуль выходит из строя, его менеджер перестает передавать ему вычислительную работу. В этом случае вычисления продолжаются другими ВМ ПК «Пирамида» с некоторым снижением общей производительности.

2.3 Характеристики ПК «Пирамида»

2.3.1 Число поддерживаемых кластеров центральным менеджером – до 20 кластеров.

2.3.2 Число поддерживаемых ВМ менеджером кластера – до 256 ВМ.

2.3.3 Число одновременно запускаемых менеджеров ВМ экземпляров ОПП – до 32 экземпляров.

2.3.4 Максимальный размер вычислительной работы для всего ПК «Пирамида» – 2^{63} вариантов.

2.3.5 Максимальное число параметров ОПП – 10 параметров.

3 Обращение к ПК «Пирамида»

3.1 Обращение к ПК «Пирамида» может быть осуществлено в одном из двух режимов: режиме монопольного доступа к вычислительной системе или режиме коллективного пользования через Систему управления прохождением параллельных заданий (СУППЗ).

3.2 В монопольном режиме 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» в соответствии с 4.1.2.

`паспорт_задания` – файл в формате ini с описанием формата вызова и параметров ОПП в соответствии с 4.1.1.

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

3.3 В режиме коллективного пользования обращение к ПК «Пирамида» осуществляется посредством запуска команды СУППЗ `prun`. Формат запуска следующий:

```
prun [опции] -tsk <имя_задания>
```

где `имя_задания` – имя задания СУППЗ;

`опции` – параметры команды `prun`.

Команда `prun` автоматически формирует задание СУППЗ и направляет его в очередь. Завершение команды `prun` не означает завершение или запуск задания, а лишь его постановку в очередь СУППЗ. После прохождения очереди СУППЗ автоматически будут сформирован конфигурационный файл ПУ «Пирамида» и вызван центральный менеджер ПК «Пирамида».

Все параметры команды prun являются ключевыми, т.е. предваряются ключевым словом со знаком -.

3.3.1 Параметр

-maxtime <максимальное_время>

задает максимальное время выполнения задания в минутах. После истечения этого времени задание будет принудительно завершено.

3.3.2 Параметр

-np <число_ядер>

задает необходимое для выполнения задания число процессорных ядер.

3.3.3 Параметр

-passp <паспорт>

файл в формате ini с описанием формата вызова и параметров ОПП в соответствии с 4.1.1.

3.3.4 Параметр

-quantum <квант>

задает время (в минутах) кванта для фонового задания СУППЗ.

3.3.5 Параметр

-ppn <число_ОПП_на_ВМ>

задает число процессов ОПП ПК «Пирамида» на одном вычислительном модуле.

3.3.6 Параметр

-thread <число_потсков_на_процесс>

задает число потоков ОПП ПК «Пирамида» в соответствии с 4.1.1.2 и А.1.2.

3.3.7 Параметр

-s <имя_логической_системы>

задает имя логической системы СУППЗ в соответствии с 2.1.7 Руководства программиста СУППЗ. Если параметр не задан, будет выбрана логическая система по умолчанию.

4 Входные и выходные данные

4.1 Входные данные

4.1.1 Паспорт задания

4.1.1.1 Паспорт задания в формате ini-файла является основным входным документом ПК «Пирамида», определяющим формат запуска ОПП и множество ее параметров и их значений.

4.1.1.2 Примерная структура паспорта задания приведена ниже.

```
# Это комментарий
# Название секции заключается в квадратные скобки и должно
# начинаться с первой позиции новой строки. Порядок следования
# секций в файле не имеет значения, за исключением, когда в
# файле присутствуют секции в одинаковыми именами. В этом
# случае к исполнению будет принята секция, указанная в файле
# первой
# Следующая строка – название секции, содержащей общие
# сведения о сценарии (секция обязательная)
[General]
# Следующая строка определяет идентификатор ОПП (элемент
# обязательный)
name = my_script
# Следующая строка определяет, разрешено ли ведение истории сбо
# всех выполненных запусках сценария (принимает значения yes
# или no; элемент необязательный; если не задан, история не
# ведётся)
historyenabled= yes
# Следующая строка определяет описание, характерное для
# данного сценария (элемент необязательный)
description = my_script
# Следующая строка определяет значение кода возврата ОПП, при
# получении которого должна быть завершена работа ПК «Пирамида»
# (элемент необязательный)
stopcode = 1
# Следующая строка определяет значение кода возврата ОПП,
```

```
# получение которого свидетельствует о том, что запуск экземпляра ОПП
#  завершился получением практически важного результата
# (элемент необязательный)
resultcode = 2
# Следующая строка определяет путь к рабочему каталогу задания
# (элемент обязательный)
dir = /home/decart
# Следующая строка определяет, расположен ли рабочий каталог
# задания в общей файловой системе (принимает значения yes
# или no; обязателен)
nfs = no
# Следующая строка определяет периодичность сохранения
# контрольных точек в секундах (элемент необязательный; если
# не задан, контрольные точки не сохраняются)
checkpoint = 10

# Следующая строка – название секции, содержащей сведения о
# параметрах ведения журналов ПК «Пирамида». Секция необязательная,
# если не задана, компоненты ПК «Пирамида» не будут вести журналы
[Log]
# Следующая строка определяет путь к каталогу для файлов-
# журналов (элемент обязательный)
path = /home/decart/log
# Следующая строка определяет, расположен ли каталог для
# файлов-журналов в общей файловой системе (принимает значения
#  yes  или no, параметр обязателен)
nfs = no
# Следующая строка определяет, будет ли вести журнал
# центральный сервер управления (принимает значения
#  yes  или no, параметр обязателен)
epk = yes
# Следующая строка определяет, будут ли вести журналы
# серверы кластеров (принимает значения
#  yes  или no, параметр обязателен)
```

```
cluster = yes
# Следующая строка определяет, будут ли вести журналы
# вычислительные модули (принимает значения
# yes или no, параметр обязателен)
node = no

# Следующая строка – название секции, содержащей сведения о
# размерах порций вычислительной работы (секция обязательная)
[Portions]
# Следующая строка определяет размеры вычислительной работы,
# выделяемой центральным сервером кластеру (элемент
# обязательный)
cluster = 16
# Следующая строка определяет размеры вычислительной работы,
# выделяемой кластером вычислительному модулю (элемент
# обязательный)
node = 4
# Следующая строка определяет размеры вычислительной работы,
# выделяемой вычислительным модулем одной последовательной
# программе (элемент обязательный)
program = 2

# Секция содержит список всех параметров для данного сценария
# (секция обязательная)
[Parameters]
# Каждая строка содержит имя параметра и определяет его тип
# одним из возможных значений: from_to, list, file
# (параметр задаётся диапазоном значений, списком значений или
# файлом со значениями соответственно. Формат:
# имя_параметра=тип_параметра
# Обязательно наличие как минимум одного параметра). Имя
# параметра уникально в сценарии и используется как символ
# подстановки в шаблонах команд запуска
param1 = from_to
```

```
param2 = list
param3 = file

# Для каждого параметра, указанного в секции Parameters,
# паспорт задания должен содержать описательную секцию. Имя
# секции задается идентификатором parameter и именем
# параметра через двоеточие

# Далее приводятся варианты формата секции parameter для
# параметров всех типов
# Секция содержит границы диапазона для параметра, задаваемого
# диапазоном значений
[parameter:param1]
# Следующая строка определяет систему счисления для данного
# параметра (принимает значения 2, 8, 10 и 16; элемент
# необязательный; если не задан, по умолчанию устанавливается 10)
base = 10
# Следующая строка определяет начальное значение для данного
# параметра (элемент обязательный)
from = 1
# Следующая строка определяет конечное значение для данного
# параметра (элемент обязательный)
to = 10
# Следующая строка определяет шаг значения для данного
# параметра (элемент обязательный)
step = 2

# Секция содержит список значений для параметра, задаваемого
# списком значений
[parameter:param2]
# Следующая строка определяет количество элементов в списке
items.num="3"
# Следующие строки определяют список
# значений для данного параметра
aaa
```

```
aab
aac

# Секция содержит имя файла со списком значений для параметра,
# задаваемого файлом
[parameter:param3]
# Следующая строка определяет имя файла со списком значений
# для данного параметра (элемент обязательный)
params_list.txt

# Секция содержит список всех профилей для данного сценария
# (секция обязательная)
[Profiles]
# Каждая строка содержит имя профиля и определяет его тип.
# «1» – универсальный процессор, «2» – графический ускоритель
# Формат:
# имя_профиля=тип_профиля
# (обязательно наличие как минимум одного профиля)
x86=1
GPU=3
# Для каждого профиля, указанного в секции Profiles,
# паспорт задания должен содержать описательную секцию. Имя
# секции задается идентификатором profile и именем профиля
# через двоеточие

# Секция содержит правила и ограничения запусков для данного
# профиля
[profile:GPU]
# Следующая строка определяет максимальное
# время выполнения ОПП для данного профиля в секундах.
maxtime = 60
# Следующая строка определяет максимальное число допустимых
# ошибочных запусков экземпляра ОПП с фиксированным набором
# параметров (элемент необязательный)
maxerrors = 1
```

```
# Следующая строка определяет шаблон команды запуска для
# данного профиля. Должен включать строку команды запуска с
# макроподстановками вида ${param1}, где param1 – имя
# одного из заданных в секции Parameters элементов. Строка
# команды запуска ОПГ будет содержать макроподстановки с
# именами всех заданных в секции Parameters элементов
# Строка команды запуска может содержать макроподстановку
# ${THREADS_COUNT}, обозначающую число потоков в ядре
# процессора (реконфигурируемого ускорителя)
# (элемент обязательный)
commandtemplate = a.out input.dat ${param1}
${THREADS_COUNT}
# Следующая строка определяет список имен файлов, необходимых
# для работы заданий данного профиля. Имена файлов разделены
# двоеточием и могут содержать макроподстановки (элемент
# необязательный)
files="a.out;input.dat"
```

4.1.2 Конфигурационный файл ПК «Пирамида»

4.1.2.1 Конфигурационный файл ПК «Пирамида» определяет вычислительные ресурсы и параметры для исполнения компонент ПК «Пирамида». Конфигурационный файл представляет собой набор секций, каждая из которых состоит из параметров и их значений. Рассмотрим формат конфигурационного файла на нижеприведенном примере.

4.1.2.2 Формат конфигурационного файла ПК «Пирамида»

```
# Это комментарий
# Далее – название секции
# Название секции заключается в квадратные
# скобки и должно начинаться с первой позиции
# новой строки.
# Порядок следования секций в файле не имеет
# значения, за исключением, когда в файле
# присутствуют секции с одинаковыми именами
```

```
# В этом случае к исполнению будет принята
# секция, указанная в файле первой.
[General]
# Обязательная секция с именем General
# Параметр workdir  определяет рабочий
# каталог ПК «Пирамида»,  который используется для
# создания и хранения различных временных файлов
workdir = /tmp/epk
# Параметр statusfile  определяет
# имя файла статуса вычислительных модулей,
# в котором содержится информация
# о текущем статусе вычислительных модулей
statusfile = /common/epk/nodesstatus.log
# Путь к исполняемому файлу центрального менеджера ПК «Пирамида»:
epk = /usr/local/bin/epk
# Путь к исполняемому файлу менеджера кластера ПК «Пирамида»:
cluster = /common/epk/cluster/cluster
# Путь к исполняемому файлу менеджера ВМ ПК «Пирамида»:
node = /common/epk/node/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 время,
```

```
# определяемое системным таймаутом. По истечении числа
# попыток кластер или ВМ, обслуживаемый не отвечающим менеджером
# нижнего уровня, помечается как неисправный
repeat = 3
# Обязательная секция с именем Clusters
# Содержит список серверов кластеров,
# логически организованных на
# центральном сервере управления
[Clusters]
# Каждый параметр секции определяет один кластер
# Формат параметра:
# имя_центрального_сервера:порт
# Менеджер кластера ПК «Пирамида» № 1
# будет ожидать соединения по порту 22001
server:22001
# Менеджер кластера ПК «Пирамида» № 2
# будет ожидать соединения по порту 22800
server:22800
# Для каждого кластера конфигурационный файл
# должен содержать описательную секцию
# Имя секции задается именем центрального сервера
# и номером порта через двоеточие
# Следующая секция содержит описание
# кластера с номером порта 22001
[server:22001]
# Менеджер ВМ ПК «Пирамида» с именем node11 будет ожидать соединения по
# порту 23001. Данный менеджер одновременно может одновременно
# ставить ОПП на счёт на 4-х ядрах процессора профиля
# x86 (на одно ядро один экземпляр ОПП), каждое ядро допускает
# одновременное выполнение 2-х потоков (передается ОПП через
# макроподстановку ${THREADS_COUNT}
node11:23001 = x86:4x2
```

```
# Менеджер ВМ ПК «Пирамида» с именем node12 будет ожидать соединения по
# порту 22800. Данный менеджер одновременно может одновременно
# ставить ОПП на счёт на 4-х ядрах процессора профиля
# x86 (на одно ядро один экземпляр ОПП), каждое ядро допускает
# одновременное выполнение 2-х потоков (передается ОПП через
# макроподстановку ${THREADS_COUNT}
node11:23001 = x86:4x2
# Следующая секция содержит описание для
# кластера с номером порта 22800
[server:22800]
# Описание производится аналогично кластеру 22001. Кластер
# имеет в подчинении ВМ с именами node21,
# node22, node23, которые будут ожидать соединения по порту 22800.
# Менеджер ВМ node21 может одновременно ставить ОПП на счёт на
# 2-х ядрах графического ускорителя (16 потоков в каждом ядре),
# остальные могут одновременно ставить ОПП
# на счёт на 1-м ядре графического ускорителя (8 потоков)
node21:22800=GPU:2x16
node22:22800=GPU:1x8
node23:22800=GPU:1x8
```

4.2 Выходные данные

4.2.1 Результаты работы каждого экземпляра ОПП принимаются из стандартного потока вывода ОПП. Вне зависимости от кода возврата ОПП результаты перенаправляются на центральный сервер управления и сохраняются в файлах `stdout.rez` (стандартные потоки вывода выполненных экземпляров ОПП) и `stderr.rez` (стандартные потоки ошибок выполненных экземпляров ОПП), расположенных в рабочем каталоге задания. Пользователю-программисту ОПП следует помнить, что в указанные файлы будут

помещены результаты всех без исключения состоявшихся запусков ОПП, произведших вывод информации в стандартные потоки вывода или ошибок.

4.2.2 В итоговых файлах результатов `stdout.rez` и `stdout.rez` каждый вывод ОПП предваряется информацией о команде запуска и параметрах ОПП, например:

```
*** node21:28001 *** RESULT *** /home/bin/opp 34 15 str2
34 - ok, 15 - ok, str2 - ok
```

В приведенном примере строка «`*** node21:28001 *** RESULT *** /home/bin/opp 34 15 str2`» записана ПК «Пирамида» и содержит информацию:

- об имени (node21) и порте соединения (28001) ВМ, на котором был выполнен экземпляр ОПП;
- о статусе обработанной порции данных (RESULT) согласно 4.2.4;
- о команде и параметрах запуска экземпляра ОПП.

Следующая строка «`34 - ok, 15 - ok, str2 - ok`» является результатом выполнения экземпляра ОПП, запущенного указанной командой.

4.2.3 В паспорте задания может быть определен элемент `log`, задающий имя каталога для файлов журнала работы ПК «Пирамида». Если элемент `log` задан, то в указанном каталоге будет вестись журнал ПК «Пирамида», содержащий протокол обработки порций вычислительной работы. Для каждой порции вычислительной работы в журнале фиксируется время начала (завершения) обработки, первый и последний элемент диапазона для каждого параметра. В записи о начале обработки порции фиксируется оставшаяся (невыполненная) работа. Если элемент `log` не задан, компоненты ПК «Пирамида» не будут вести журналы.

Атрибут `NFS` элемента `log` может принимать значения `yes` или `no` в зависимости от того, расположен каталог для файлов-журналов в общей файловой системе или нет.

Журнал ведется на каждом ВМ индивидуально в отдельном файле. Имя файла-журнала центрального сервера управления – `epk.log`. Имена файлов-журналов на серверах кластеров и остальных ВМ имеют следующий формат:

`host:port.log`

Здесь: `host` – сетевое имя ВМ или сервера кластера; `port` – номер порта соединения.

4.2.4 Для каждой порции, выделяемой менеджером, в журнале фиксируется время начала (завершения) обработки, первый и последний элемент диапазона для каждого параметра. Кроме этого, в записи о начале обработки порции фиксируется оставшаяся (невыполненная) работа (remained work). В записи об окончании работы фиксируется статус обработанной порции:

- NORMAL – «успешно»;
- FAIL – «аварийно»;
- ERROR – «ошибочно»;
- RESULT – «получен результат».

При этом каждый слайс вычислительной работы представляется одной строкой. Значения строковых параметров заключаются в двойные кавычки.

Статус RESULT («получен результат») присваивается обработанной порции при получении кода возврата, соответствующему параметру `result_code` паспорта задания.

Пример записи о начале обработки порции из примера 2.2.2:

```
*** 15.08.2008 15:30:41 ***
Part in work:
"Str1"-"str3" 18-22 1-1
"Str1"-"Str1" 24-24 1-1
The remained work:
"str2"-"str3" 24-24 1-1
"Str1"-"str3" 26-28 1-1
"Str1"-"str3" 18-28 2-3
```

Пример записи об успешном завершении обработки порции из 2.2.2:

```
*** 15.08.2008 15:30:43 ***
Part complete (NORMAL):
"Str1"-"str3" 18-22 1-1
"Str1"-"Str1" 24-24 1-1
```

Пример записи об аварийном завершении обрабстки порции из 2.2.2:

```
*** 15.08.2008 15:30:43 ***
Part complete (FAIL):
"Str1"-"str3" 18-22 1-1
"Str1"-"Str1" 24-24 1-1
```

Пример записи об ошибочном завершении обработки порции из 2.2.2:

```
*** 15.08.2008 15:30:43 ***
Part complete (ERROR):
"Str1"-"str3" 18-22 1-1
"Str1"-"Str1" 24-24 1-1
```

4.2.5 Кроме файлов `stdout.rez` (стандартные потоки вывода выполненных экземпляров ОПП) и `stderr.rez` (стандартные потоки ошибок выполненных экземпляров ОПП) результатом работы ОПП считаются все файлы, созданные в рабочем каталоге задания во время выполнения экземпляра ОПП. Если рабочий каталог задания размещается в локальных файловых системах кластеров и ВМ, ПК «Пирамида» будет осуществлять сбор созданных ОПП файлов из локальных файловых систем ВМ в рабочий каталог задания на центральном сервере управления.

4.2.6 Центральный менеджер ПК «Пирамида» в рабочем каталоге задания ведет файл `work.status`, отражающий текущее состояние выполнения вычислительной работы. В файле `work.status` отмечается три порции вычислительной работы: выполненная, выполняющаяся и невыполненная. Пример файла `work.status`:

```
*** 15.08.2008 15:30:41 ***
Completed work:
"Str1"-"str3" 18-22 1-1
Part in work:
"Str1"-"Str1" 24-24 1-1
Remained work:
"str2"-"str3" 24-24 1-1
"Str1"-"str3" 26-28 1-1
"Str1"-"str3" 18-28 2-3
```

4.3 Контрольные точки

4.3.1 Контрольные точки формируются только на центральном сервере управления через равные промежутки времени. Интервал создания контрольной точки задается в секундах в паспорте задания (элемент `checkpoint`). Контрольная точка формируется в одном из двух подкаталогов рабочего каталога задания с именами `chp1` и `chp2`. На текущую контрольную точку указывает символическая ссылка с именем `work`, также расположенная в рабочем каталоге задания.

4.3.2 Работа ПК «Пирамида» может быть прервана в любой момент времени и возобновлена с последней созданной контрольной точки. Будьте внимательны при запуске ПК «Пирамида» с новым паспортом задания в том же рабочем каталоге. Для нового запуска (или повторения предыдущего запуска с начала) необходимо удалить из рабочего каталога подкаталоги `chp1` и `chp2`, а также символическую ссылку `work`.

5 Сообщения программисту

7.1 В нормальном режиме функционирования менеджеры ПК «Пирамида» сообщают о своем старте и завершении. При возникновении ошибки или аварийной ситуации выдается соответствующее диагностическое сообщение.

7.2 Центральный менеджер выводит диагностические сообщения в стандартный поток ошибок и в файл журнала. Менеджеры кластеров и ВМ выводят сообщения в файлы журнала и, при возможности и необходимости, – в выходной файл результатов работы.

7.3 Рассмотрим формат диагностических сообщений.

```
дата время сервер менеджер[pid]: имя:порт: имя функции:
диагностика функции: системная диагностика
```

Здесь:

дата – дата поступления сообщения (как правило, в формате "Mmm dd");

время – время поступления сообщения (как правило, в формате "hh:mm:ss");

сервер – сетевое имя центрального сервера, сервера управления кластером или ВМ;

менеджер – обозначение менеджера:

epk – центральный менеджер;

cluster manager – менеджер кластера;

node manager – менеджер ВМ;

pid – системный идентификатор процесса-менеджера;

имя:порт – имя и порт менеджера, заданные как в конфигурационном файле ПК «Пирамида», так в файле регистрации суперсервера xinetd;

Имя функции – имя функции, обнаружившей аварийную ситуацию. Ниже приводятся имена основных функций ПК «Пирамида»:

thread <имя> – функция потока управления кластером (ВМ или ОПП) с заданным именем;

threads или result – функция основного потока управления;

check_point_thread – функция потока создания контрольных точек;

initialize_work – функция инициализации работы менеджера ПК «Пирамида»;

read_configuration – функция чтения конфигурационного файла ПК «Пирамида».

диагностика функции представляет собой сообщение на английском языке, определяющее аварийное место в программе (какое действие вызвало аварийную ситуацию).

системная диагностика представляет собой сообщение на английском языке, полученное ПК «Пирамида» от операционной системы.

7.4 Все сообщения можно разделить на две группы в зависимости от того, вызвано сообщение ошибкой пользователя или системной ошибкой. В сообщениях второй группы отсутствует диагностика функции, само сообщение содержит лишь системную диагностику. Действия программиста в этом случае полностью определяются содержанием системной диагностики в соответствии с руководством пользователя ОС Linux (man pages).

7.5 Таблица 1 содержит список сообщений (поле «диагностика функции»), вызванных ошибками пользователя, и действий программиста (пользователя) по устранению аварийной ситуации.

Таблица 1

Диагностика функции	Описание ситуации	Действия программиста
invalid parameters	Неправильные параметры вызова исполняемого модуля	Проверить правильность установки ПК «Пирамида»

Продолжение таблицы 1

Диагностика функции	Описание ситуации	Действия программиста
directory for log file not found, working without log	Каталог для хранения файла журнала не найден. Причина – не задано (или неправильно задано) значение параметра <path> в элементе <log> паспорта задания.	При необходимости – правильно определить элемент <log> в паспорте задания.
Work directory not found	Рабочий каталог ПК «Пирамида» не найден. Причина – не задан (или неправильно задан) параметр workdir в секции [General] конфигурационного файла ПК «Пирамида».	Необходимо задать рабочий каталог ПК «Пирамида» в конфигурационном файле ПК «Пирамида», либо проверить наличие указанного в конфигурационном файле каталога.
directory for work results not found	Каталог для хранения результатов работы ПК «Пирамида» не найден. Причина – не задано (или неправильно задано) значение параметра <path> в элементе <directory> паспорта задания.	Необходимо задать каталог для хранения результатов работы ПК «Пирамида».
wrong configuration file format	Содержимое конфигурационного файла ПК «Пирамида» не соответствует заданному формату.	Необходимо проверить правильность формирования конфигурационного файла ПК «Пирамида».
wrong passport format	Содержимое паспорта задания не соответствует заданному формату.	Необходимо проверить правильность формирования паспорта задания в соответствии с 4.1.1.
failed to start manager (*имя_узла:порт*)	Не удалось запустить менеджер (*имя_узла:порт*).	Действия определяются содержанием системной диагностики.
manager (*имя_узла:порт*) crashed	Аварийное завершение работы менеджера (*имя_узла:порт*).	
unknown error with manager (*имя_узла:порт*)	Ошибка в работе менеджера (*имя_узла:порт*) по неустановленной причине.	

Окончание таблицы 1

Диагностика функции	Описание ситуации	Действия программиста
network error	Ошибка, вызванная отказом сетевого оборудования.	Действия определяются содержанием системной диагностики.
child manager on (*имя_узла:порт*) closed connection	Разрыв соединения с менеджером (*имя_узла:порт*).	
opp: the process failed to start	Ошибка запуска ОПП.	
opp: the process crashed some time after starting successfully	Аварийное завершение работы ОПП.	
opp: the last waitFor...() function timed out	ОПП не отвечает на команды.	
opp: an unknown error occurred	Ошибка в работе ОПП по неустановленной причине.	
host (*имя_узла:порт*) not found	Узел (*имя_узла:порт*) не найден.	Проверить правильность конфигурационного файла ПК «Пирамида» в соответствии с 4.1.2.
the bound address is already in use	Указанный адрес уже используется.	
error on read section [General]	Ошибка в чтении секции [General] конфигурационного файла ПК «Пирамида».	Исправить секцию [General] конфигурационного файла ПК «Пирамида» в соответствии с 4.1.2.
invalid count of nodes	Ошибка в конфигурационном файле ПК «Пирамида». Возможная причина - отсутствие секции описания кластера, хотя имя и порт менеджера кластера присутствуют в секции [Clusters].	
epk: error while starting! usage: epk -f configuration_filename -p xml_pasport	Ошибка при запуске ПК «Пирамида». Указаны не все входные параметры.	Указать все необходимые параметры при запуске ПК «Пирамида».

Приложение А

(обязательное)

Тестовые примеры

А.1 Тестовый пример № 1. Программа моделирования работы ОПП с тремя параметрами

А.1.1 Тестовый пример № 1 представлен программным модулем с именем opp. Программа принимает от ПК «Пирамида» три параметра – два числовых и один строковый. Параметры задаются в соответствии с требованиями ПК «Пирамида» к ОПП (см. 4.2).

А.1.2 Формат вызова тестового примера № 1:

```
opp <num1> <num2> <list> <timeout> <threads> [alerts_file]
```

num1 – первый числовой параметр;

num2 – второй числовой параметр;

list – строковый параметр;

timeout – максимальная задержка выполнения;

threads – количество потоков в ядре;

alerts_file – имя файла с комбинациями параметров, моделирующими аварийную ситуацию (необязательный параметр).

А.1.3 Программа opp перебирает и выводит в стандартный вывод все возможные комбинации трех параметров ПК «Пирамида» и осуществляет случайную задержку в интервале от 0 до timeout секунд. Если задан параметр alerts_file, то из файла будут прочитаны и приняты к исполнению комбинации параметров, моделирующие аварийную ситуацию.

А.1.4 Пример вызова программы opp без моделирования аварийной ситуации:

```
opp "1 5 1" "146 150 2" "str1 str2" 4 2
```

Стандартный вывод программы opp в этом случае будет представлять перебор всех комбинаций значений от 1 до 5 (с шагом 1), от 146 до 150 (с шагом 2) и строк str1 и str2. В первой строке стандартного вывода будет выведена информация о сетевом имени вычислительного модуля, на котором выполнялся тестовый пример, и о числе потоков на ядре.

```
*** host: cln1  threads: 2 ***
------------
2 146 str1
------------
3 146 str1
------------
4 146 str1
------------
5 146 str1
------------
1 148 str1
------------
2 148 str1
------------
3 148 str1
------------
4 148 str1
------------
5 148 str1
------------
1 150 str1
------------
2 150 str1
------------
3 150 str1
------------
4 150 str1
------------
5 150 str1
------------
1 146 str2
------------
2 146 str2
------------
3 146 str2
------------
4 146 str2
------------
5 146 str2
------------
1 148 str2
------------
2 148 str2
------------
```

```
3 148 str2
------------
4 148 str2
------------
5 148 str2
------------
1 150 str2
------------
2 150 str2
------------
3 150 str2
------------
4 150 str2
------------
5 150 str2
```

А.1.5 В программе opp возможно смоделировать аварийную ситуацию, задав параметр alerts_file и подготовив текстовый файл с соответствующим именем. Формат файла следующий:

```
комбинация_параметров_1 таймаут_1 код_возврата_1
комбинация_параметров_2 таймаут_2 код_возврата_2
.....
комбинация_параметров_20 таймаут_20 код_возврата_20
```

В файле можно задать до 20 комбинаций (наборов) параметров. Для каждой комбинации задается таймаут в секундах и код возврата (целое число от 0 до 255). Если в процессе перебора значений параметров программа opp обнаружит совпадение с хотя бы одной комбинацией, заданной в файле alerts_file, будут произведены следующие действия:

- программа opp будут ожидать истечения таймаута, соответствующего текущей комбинацией значений параметров;

- по истечении таймаута программа opp завершится с кодом возврата, соответствующим текущей комбинацией значений параметров.

А.1.6 Рассмотрим пример моделирования аварийных ситуаций. Создадим файл с именем alerts и следующим содержимым:

```
3 150 str1 30 3
5 148 str2 300 0
```

Указанный файл описывает две комбинации значений параметров: 3 150 str1 и 5 148 str2. Если программа opp в процессе перебора обнаружит комбинацию

3 150 str1, то она будет ожидать истечения таймаута в 30 секунд, после чего завершится с кодом возврата 3. Если программа opp в процессе перебора обнаружит комбинацию 5 148 str2, то она будет ожидать истечения таймаута в 300 секунд, после чего завершится с кодом возврата 0.

А.1.7 Таким образом возможно моделирование на определенных комбинациях значений параметров как аварийного завершения ОПП с ненулевым кодом возврата, так и истечения таймаута ОПП, заданного в XML-паспорте задания (п. 4.1.1.14).

А.1.8 Пример вызова программы opp с моделированием аварийной ситуации на комбинации значений параметров 3 150 str1:

```
opp "1 5 1" "146 150 2" "str1 str2" 4 8 alerts
```

Стандартный вывод программы opp в этом случае будет представлять перебор всех комбинаций значений от 1 до 5 (с шагом 1), от 146 до 150 (с шагом 2) и строк str1 и str2, однако при обнаружении комбинации 1 150 str1 программа выдаст сообщение «!!!Alert combination!!!», будет ожидать окончания указанного в файле alerts для комбинации 3 150 str1 таймаута (30 секунд) и завершится с кодом возврата 3. Стандартный вывод будет иметь следующий вид:

```
*** host: c1n1 threads: 8 ***
------------
2 146 str1
------------
3 146 str1
------------
4 146 str1
------------
5 146 str1
------------
1 148 str1
------------
2 148 str1
------------
3 148 str1
------------
4 148 str1
------------
5 148 str1
------------
1 150 str1
```

```
------------
2 150 str1
------------
3 150 str1
!!!Alert combination!!!
```

А.1.9 Оценка работоспособности ПК «Пирамида» производится путем анализа файла результата, который должен содержать все возможные комбинации значений параметров, заданные в паспорте задания.

А.1.10 Примерное содержание паспорта задания, оформленного в соответствии с 4.1.1, следующее.

```
[General]
historyEnabled="yes"
name="cpp"
description="Cpp x86"
stepcode="10"
resultcode="1"
checkpoint="1"
nfs="yes"
dir="/home1/kiselev/bin"

[Lcg]
path="/home1/kiselev/bin"
nfs="yes"
cluster="yes"
node="no"
epk="yes"

[Portions]
cluster="10000"
node="1000"
program="100"

[Parameters]
Param="FromTo"
Param1="FromTo"
Param2="list"

[parameter:Param]
base="10"
from="2"
to="10000"
step="2"

[parameter:Param1]
base="10"
from="1"
to="100"
```

```
step="1"

[parameter:Param2]
items.num="3"
hai
hoi
hui

[profiles]
x86

[profiles]
x86

[profile:x86]
maxErrors="5"
maxTime="5"
name="x86"
commandTemplate="opp ${Param} ${Param1} ${Param2} 20
${THREADS_COUNT} alerts_file"
files="opp;params.list"
```

А.2 Тестовый пример № 2. Программа моделирования работы ОПП с одним параметром

А.2.1 Тестовый пример № 2 представлен программным модулем с именем testprog. Программа принимает от ПК «Пирамида» один числовой параметр, содержащий только одно значение. Значение задается в соответствии с требованиями ПК «Пирамида» к ОПП согласно 4.2.

А.2.2 Формат вызова тестового примера № 2:

```
testprog <num> <N>
```

num – числовой параметр;

N – размер примерных вычислений.

А.2.3 Программа `testprog` производит N делений по модулю, создает файл с именем `num.dat` и записывает в него значение параметра `num`.

А.2.4 Пример вызова программы `testprog`:

```
testprog 5 10000
```

В результате будет создан файл `5.dat`, в котэрый будет помещена строка `"5"`. Програ€мма осуществит 10000 делений по мсдулю.