ИПМ им. М.В. Келдыша РАН
Отдел ИВСиЛС, сектор эксплуатации МВС

ИДСТУ СО РАН

Отдел высокопроизводительных
и распределенных вычислительных систем

Руководство пользователя системы МВС-1000/16

2007

Содержание

1. Общие понятия

Построение системы МВС-1000/16 с точки зрения пользователя выглядит следующим образом. Система состоит из следующих важных компонент: многопроцессорного вычислителя, управляющей ЭВМ, сервера доступа и файлового сервера. Вычислитель состоит из двухпроцессорных модулей – узлов, каждый из которых уникально именован в системе. Для примеров далее будет считаться, что сервер доступа объединен с управляющей ЭВМ и имеет сетевое имя **`irc-1`**, имена узлов – от **`irc-2`** до **`irc-17`**.

Задача пользователя выполняется на одном или нескольких процессорах вычислителя. Задачи могут запускаться и завершаться независимо друг от друга. Вычислитель делится между задачами динамически с точностью до узла. В то же время, отдельный узел не делится между задачами: если одна задача получила некоторый узел, другая задача воспользоваться им не сможет до завершения первой. При запуске задачи на счет она получает набор узлов с произвольными именами из числа свободных. При этом другие узлы задаче не доступны.

На файловом сервере каждый пользователь имеет свой домашний каталог **`/home/имя_пользователя`**. Сервер служит для подготовки и хранения исходных текстов программ пользователей, данных для пользовательских задач, результатов расчетов, для компиляции и подготовки самих задач.

Управляющая ЭВМ служит для доступа пользователей, а также для запуска (завершения, управления) на многопроцессорном вычислителе пользовательских задач. На управляющей ЭВМ ведется *очередь* задач к вычислителю. Заметим, что

возможно объединение функций файлового сервера и управляющей ЭВМ на одной рабочей станции.

Старт (завершение, получение информации) задачи инициируется запуском на сервере доступа пользователем команд системы запуска (см. ниже).

2. Система запуска задач

Следует отметить, что “разъяснение” системе того, сколько процессоров требуется данной задаче, какие программы должны быть загружены на каждом, в каких файлах окажется стандартный ввод и стандартный вывод и т. п. само по себе является нетривиальной задачей. Кроме этого, необходимо ведение очередей пользовательских задач, контроль за их исполнением и освобождение после окончания счета задачи тех вычислительных ресурсов, которые она занимала. В МВС-1000/16 для решения этой проблемы служит специальная программная компонента - **СИСТЕМА ЗАПУСКА ЗАДАЧ**. Она прозрачна для программы, обслуживает исключительно взаимодействие системы с пользователем в процессе запуска программы на счет и последующего контроля.

Задачей называется параллельная прикладная программа, предназначенная для выполнения на нескольких процессорах системы МВС-1000/16.

Весь производимый задачей в процессе выполнения стандартный вывод (печать на экран), а также стандартный вывод сообщений об ошибках будут перенаправлены в специальные *выходные файлы*. По желанию пользователь может задать системе эти файла, равно как и файл *стандартный ввода*. Выходные файлы, файл стандартного ввода, а также некоторые другие служебные файлы создаются в *рабочем каталоге задачи на управляющей ЭВМ (далее - каталоге стандартного ввода/вывода)*. Кроме этого, возможна работа с *интерактивными* задачами (подробнее см. п. 15).

Задача пользователя выполняется на МВС-1000/16 определенное время (значение по умолчанию задается в пользовательском конфигурационном файле, подробнее см. п. 17). Пользователь имеет возможность самостоятельно специфицировать время выполнения. По истечении специфицированного времени задача будет снята со счета.

Каждая запущенная задача получает уникальное имя, состоящее из символьного имени задачи и ее номера, что позволяет запускать одну и ту же задачу одновременно в нескольких экземплярах. Для каждого экземпляра система создаст собственный каталог стандартного ввода/вывода.

Каждая запущенная задача контролируется специальным процессом управляющей ЭВМ – менеджером задачи. Стандартный вывод этого процесса также сохраняется в каталоге стандартного ввода/вывода.

Для автоматизации процесса запуска задачи и служит система запуска задач. Отметим, что система запуска имеет свой конфигурационный файл, структура которого описана в п. 17.

3. Основные принципы построения очередей

ВНИМАНИЕ! Данный раздел содержит информацию об организации системы очередей для МВС-1000/16. Просьба к пользователям внимательно ознакомиться с его содержанием и соблюдать правила постановки в очередь задач. Нарушение указанных правил может привести к тому, что Ваша задача будет вечно стоять в очереди и **НИКОГДА** не запустится.

Все задачи пользователей делятся на три категории – *отладочные, пакетные и фоновые*.

Отладочные задачи – это короткие по времени задачи, которые запускаются исключительно в целях отладки.

Пакетные задачи – это средние по времени задачи, которые производят реальные расчеты и выполняются, не прерываясь.

Фоновые задачи – задачи с большим временем счета, которые могут прерываться системой. Для фоновой задачи пользователь должен явно указать *квант* – минимальное время счета фоновой задачи, в течение которого задачу прерывать нельзя.

Планирование очередей в каждый момент времени производится в соответствии с параметрами текущего *режима планирования*. Режим планирования определяется следующими параметрами:

- дата и время включения режима;
- максимальное время, отведенное для отладочных задач;
- максимальное время, отведенное для пакетных задач;
- число процессоров, которые “резервируются” для отладочных задач;
- шкала приоритетов пользователей.

Рассмотрим каждый параметр подробнее.

Дата и время включения определяют время, начиная с которого параметры режима вступают в силу. Параметры режима действуют вплоть до включения следующего режима.

Максимальное время, отведенное для отладочных задач, определяет время счета отладочной задачи. Если пользователь хочет сообщить системе, что его задача отладочная, то он должен указать для задачи время счета, не превышающее значение максимального времени для отладочных задач. При превышении данного значения задача автоматически будет считаться пакетной.

Максимальное время, отведенное для пакетных задач, определяет время счета для пакетной задачи. **ВНИМАНИЕ!** Пакетные задачи, время которых превышает максимальное отведенное для пакетных задач, не будут включены в счет в текущем режиме. **ЕСЛИ ВЫ ХОТИТЕ СЧИТАТЬ ДОЛГО, ДЕЛАЙТЕ ЗАДАЧУ ФОНОВОЙ!**

Число процессоров, которые “резервируются” для отладочных задач. Все пакетные задачи в сумме не могут занимать процессоров больше, чем разность между общим числом процессоров и значением данного параметра. При этом процессоры числом, указанным в данном параметре, в текущем режиме будут использоваться только для счета отладочных задач (“резервируются”). Подчеркнем, что “резервируются” не конкретные процессоры или вычислительные модули. Система гарантирует, что определенное в данном параметре число процессоров не будет использовано для счета пакетных задач, а какие конкретно процессоры попадут в это число, зависит от текущей ситуации. Данный параметр введен для дневных режимов, чтобы обеспечить постоянное наличие свободных процессоров для отладочных задач.

Шкала приоритетов пользователей. Задачи планируются системой согласно приоритетам пользователей, т.е. задача пользователя с высоким приоритетом может посчитаться раньше, чем задача пользователя с низким приоритетом. Приоритет пользователя определяется по указанной шкале и напрямую зависит от суммарного времени счета пользователя за учетный период. Например, если шкала имеет следующий вид:

(120,300,600,1200,0)

то это означает, что наивысшим приоритетом (*очередь 0*) будут обладать задачи пользователей, которые за учетный период считали менее 120 минут, чуть меньшим (*очередь 1*) приоритетом – тех, кто считал менее 300 минут, еще меньшим (*очередь 2*) – тех, кто считал менее 600 минут и т.д. Низшим для данного режима (*очередь 4*) приоритетом будут обладать задачи пользователей, считавших более 1200 минут. При вычислении приоритета задачи учитывается заказываемое пользователем ее время счета. Последний ноль означает “конец списка”. **ВНИМАНИЕ!** При вычислении приоритета задачи учитывается заказываемое пользователем ее время счета. Размер учетного периода определяется администратором системы.

Планировщик пытается выделить процессоры из числа свободных сначала для счета задач из очереди 0, потом — из очереди 1 и т.д. Внутри одной очереди ресурсы выделяются в порядке (от меньшего к большему) номеров в *списке задач*, в котором находятся все считающиеся и ждущие задачи. Если свободных процессоров для текущей задачи нет, определяется момент, когда нужное число процессоров будет доступно, и устанавливается время старта данной задачи. Никакая менее приоритетная задача не может занять процессоры так, чтобы это отодвинуло старт более приоритетной. Однако следует понимать, что при отсутствии конфликта по ресурсам менее приоритетная задача может стартовать раньше более приоритетной. В случае конфликта по ресурсам опережающий старт менее приоритетной задачи возможен лишь, если ее время счета таково, что она освободит ресурсы раньше запланированного старта более приоритетной.

Планирование фоновых задач осуществляется следующим образом. Пользователь должен указать квант для фоновой задачи. Система гарантирует, что если фоновая задача была выбрана на счет, то ей будет дано для счета время, не меньшее указанного кванта. По истечении кванта задача может быть снята системой со счета и заново поставлена в очередь. **ВНИМАНИЕ!** Организацию контрольных точек и повторного старта должен обеспечить сам пользователь. Это означает, что при повторном старте фоновой задачи система не восстанавливает состояние вычислений данной задачи.

Если указанный пользователем квант не превышает максимального времени для отладочных задач, то его фоновая задача может быть включена в решение наряду с прочими отладочными задачами. Приоритет фоновой задачи зависит от типа расписания, сгенерированного администратором. В одном варианте приоритет фоновой задачи рассчитывается так же, как и обычной задачи, что может приводить к некоторым ошибкам в прогнозе запуска задач. В другом варианте дается более точная оценка времени запуска задач, но приоритет фоновой задачи при этом ниже обычной (очередь 6) и не зависит от приоритета владельца.

Если указанный пользователем квант не превышает максимального времени для пакетных задач, то фоновая задача планируется, как пакетная. Она уже не сможет занимать процессоры, “зарезервированные” для отладочных задач. При превышении квантом максимального времени для пакетных задач фоновая задача не сможет войти в счет в текущем режиме.

4. Задание дополнительного времени для завершения счета

При запуске задачи пользователь имеет возможность указать время предупреждения задачи о снятии со счета. Время это может быть любым, но не

больше установленного администратором системы предела. Далее будем называть это время *дополнительным временем*.

Задачи тех пользователей, которые не заказали дополнительное время, планируются системой очередей обычным образом. Если же дополнительное время заказано, то при планировании задачи оно будет прибавляться к основному времени (для отладочных и пакетных задач), либо к кванту (для фоновых задач).

При снятии со счета задачи с заказанным дополнительным временем система предпримет некоторые действия по предупреждению процессов задачи о предстоящем завершении. Для своевременной реакции на предупреждение системы разработчики настоятельно рекомендуют пользоваться специальной функцией **`cp_signal`** (**`cpf_signal`**) API для организации контрольных точек (см. документ “Библиотека интерфейсных вызовов (API) для организации контрольных точек (КТ)”).

Другим способом отследить время снятия задачи со счета является указание системе разослать заданный пользователям сигнал всем процессам задачи. **ВНИМАНИЕ!** Система рассылает сигнал о предстоящем снятии задачи со счета только в том случае, когда пользователь явно укажет при старте задачи дополнительное время и соответствующий сигнал. Сигнал задается пользователем либо по числовому номеру (от 1 до 20), либо по мнемонике, принятой в стандартной команде **`kill`**. Другими словами, формат задания сигнала должен соответствовать принятому в команде **`kill`**. Более подробную информацию о сигналах смотрите в описании команды **`kill`**.

Каким бы способом пользователь не воспользовался для извещения своей задачи о предстоящем снятии со счета, при получении данного извещения задача пользователя должна будет выполнить все необходимые действия для своего завершения. По истечении заданного дополнительного времени задача будет безусловно снята со счета.

Системой гарантируется, что отладочные и пакетные задачи при запуске сначала получат для счета свое заказанное время, по истечении которого задача будет извещена о снятии со счета и выделено дополнительное время для завершения. Для фоновых задач при запуске гарантируется получение, как минимум, одного указанного кванта. Если по истечении кванта система принимает решение о снятии задачи со счета, то сначала задача предупреждается о снятии со счета, а затем выделяется дополнительное время для завершения.

5. Ресурсы локальной дисковой памяти

5.1 Общие понятия

Пользователь имеет возможность использовать для своих задач локальную дисковую память вычислительных модулей (ЛДП). Система планирует ЛДП как отдельный вычислительный ресурс. Основными характеристиками ресурса ЛДП являются:

1. Уникальное имя ресурса ЛДП.
2. Пользователь - владелец ресурса ЛДП.
3. Количество вычислительных модулей в ресурсе ЛДП.
4. Размер ЛДП на одном вычислительном модуле.

Ресурс ЛДП может быть выделен по специальной команде администратора системы. При выделении ресурса ЛДП к его характеристикам добавляется список модулей, на которых он размещен. После выделения ресурса ЛДП его владелец

(пользователь) при запуске любой своей задачи имеет право в качестве параметра указать имя этого ресурса. Сервер очередей, планируя такую задачу, будет обеспечивать попадание задачи на модули, содержащие указанный ресурс. В настоящем варианте системы от пользователя требуется равенство числа модулей, заказываемых под задачу, числу модулей в ресурсе ЛДП.

На локальном диске вычислительного модуля ресурс ЛДП пользователя размещается в специально выделенной администратором файловой системе. Сам ресурс ЛДП представляет собой каталог с именем вида

```
/store/user
```

где **store** – имя файловой системы (задается администратором), **user** – имя пользователя ресурса ЛДП. Пользователь имеет право читать файлы из этого каталога, а также записывать в него в информацию в объеме, не превышающем заказанный размер ресурса ЛДП.

ВНИМАНИЕ! В данной версии системы запуска отсутствует понятие каталога монтирования, введенное в предыдущей версии. Каталог монтирования не нужно указывать при запуске задач, использующих ресурс ЛДП.

Выделенный ресурс ЛДП может быть освобожден по команде администратора. По этой команде (если она завершилась нормально) также удаляются все файлы, в т.ч. файлы контрольных точек, расположенные в этом ЛДП. Причинами невыполнения данной команды могут быть наличие в очереди задач с заказом ресурса ЛДП и использование ресурса ЛДП его владельцем.

Для одной конкретной задачи пользователь имеет возможность получить ресурс ЛДП без вмешательства администратора. Для этого при запуске задачи пользователь должен указать имя и основные характеристики требуемого ресурса ЛДП. Задача с заказом ЛДП будет поставлена в очередь. Сервер очередей при наличии свободной ЛДП выделит ресурс ЛДП для задачи, после чего будет планировать ее на модули, содержащие ресурс. Такой ресурс, выделенный для конкретной задачи, называется *разовым или временным*.

После завершения задачи (для фоновой задачи – завершения всех ее квантов) разовый ресурс ЛДП будет автоматически удален. Разовый ресурс ЛДП может быть использован только той задачей, для которой был заказан. Пользователь не имеет возможности поставить в очередь еще одну задачу с заказом разового ресурса ЛДП, выделенного для другой задачи.

Для продвижения в очереди задачи, требующей ресурса ЛДП, администратор может директивно выделить ресурс ЛДП, задав такое же имя, как и у ресурса стоящей в очереди задачи. В этом случае сервер очередей, обнаружив, что ресурс с таким именем выделен администратором, станет планировать соответствующую задачу. Другими словами, если какая-либо задача долго стоит в очереди по причине невыделения ресурса ЛДП (например, не хватает дисковой памяти), пользователь может обратиться к администратору с просьбой о специальном выделении ресурса ЛДП для конкретной задачи.

При неисправности одного или нескольких модулей ресурса ЛДП задачи, их требующие, блокируются в очереди. Заказ ЛДП задач, блокированных тем или иным образом, в планировании запуска других задач не учитывается. Факт блокировки отмечается в выдаче команды **mqinfo** (см. п. 12.3).

5.2 Команды работы с ресурсами ЛДП

Непосредственно выделить ресурс ЛДП пользователь не может. Разовый ресурс ЛДП можно получить, указав в команде **mpirun** ключи **ldmname**, **ldmnodes** и **ldmspace**, задающие соответственно имя ресурса ЛДП, количество модулей и размер ресурса. Подробнее см. п. 6.

Пользователь может просмотреть состояние ЛДП по команде

```
ldminfo [-a] [-c] [пользователь/имя_ресурса]
```

Без параметров данная команда построчно выводит информацию обо всех выделенных ресурсах ЛДП, на каждый ресурс ЛДП - одна строка выдачи, содержащая имя пользователя-владельца, имя ресурса ЛДП, количество модулей, размер ЛДП на модуле и признак того, является ли ресурс ЛДП временным.

Явное указание полного имени ресурса в виде **пользователь/имя_ресурса** приведет к выдаче информации только о данном ресурсе.

Указание ключа **-a** видоизменит выдачу следующим образом. Каждая строка выдачи будет содержать имя пользователя, имя ресурса, имя модуля, на котором размещен ресурс, и признак временности. На каждый ресурс ЛДП в выдаче приходится столько строк, сколько модулей занимает данный ресурс.

Ключ **-c** формализует выдачу: из нее убирается заголовок, поля в строках разделяет строго один пробел. Рекомендуется использовать данный ключ для разбора выдачи в командных файлах.

5.3 Доступ к ресурсам ЛДП

Доступ к своему ресурсу ЛДП возможен двумя способами – из процессов параллельной задачи, использующей ресурс ЛДП и с сервера доступа.

Для доступа к ресурсам ЛДП из процессов задачи необходимо обращаться к каталогу вида

```
/store/user
```

где **store** – имя файловой системы (задается администратором), **user** – имя пользователя ресурса ЛДП. Пользователь имеет право читать файлы из этого каталога, а также записывать в него в информацию в объеме, не превышающем заказанный размер ресурса ЛДП.

Необходимо помнить, что, несмотря на одинаковое название данного каталога на всех модулях, на каждом модуле он указывает на свою, непересекающуюся с другими, часть ресурса ЛДП.

Доступ к ресурсу ЛДП можно получить с сервера доступа, независимо от того, считается или нет задача, использующая данный ресурс. Для этого, прежде всего, необходимо смонтировать ресурс командой:

```
mountldm -rname <имя_ресурса_ЛДП> -node <имя_модуля>
```

имя_ресурса_ЛДП – имя ресурса ЛДП, доступ к которому необходимо получить;
имя_модуля – имя вычислительного модуля, входящего в состав ресурса ЛДП, доступ к которому необходимо получить;

Команда **mountldm** для доступа к части ресурса ЛДП **имя_ресурса_ЛДП**, размещенной на модуле **имя_модуля**, печатает на стандартный вывод (при

успешном своем выполнении) имя каталога монтирования. Для доступа к ресурсу ЛДП необходимо перейти в этот каталог.

Например, у пользователя **max** имеется ресурс с именем **my_ldm**, размещенный на модулях **irc-3** и **irc-8**. Пусть необходимо получить доступ к части этого ресурса, расположенной на модуле **irc-8**. Для этого можно выполнить следующую команду:

```
cd `mountldm –rname my_ldm –node irc-8`
```

При успешном выполнении данной команды пользователь окажется в каталоге монтирования **/ldm/max.my_ldm.irc-8** – именно к нему будет примонтирована требуемая часть ресурса ЛДП.

ВНИМАНИЕ! Доступ к ресурсам ЛДП с сервера доступа базируется на системе **automount** (автомонтирование). Пользователь должен помнить о следующих особенностях работы с данной системой:

1. После того, как пользователь покинет каталог монтирования (например, закончит работу с примонтированным ресурсом ЛДП), ресурс ЛДП, по истечении заданного администратором таймаута, автоматически отмонтируется от сервера доступа и перестает быть доступен. Для повторного доступа пользователь должен заново выполнить команду **mountldm**.

2. Если пользователь допускает ошибку в команде **mountldm** (например, неверно указывает имя ресурса или вычислительного модуля), то выдаваемое диагностическое сообщение будет всегда одно и тоже:

```
No such file or directory
```

3. Во время работы пользователя с ресурсом ЛДП, данный ресурс может быть удален (например, завершилась задача, использующая разовый ресурс). При попытке обратиться к смонтированному ресурсу пользователь получит диагностическое сообщение:

```
Stale NFS file handle
```

В этом случае пользователь должен покинуть каталог монтирования, чтобы система смогла размонтировать удаленный ресурс ЛДП.

5.4 Получение информации о занятости ЛДП на вычислительных модулях

Выполняется по команде

```
ldmspace
```

Каждая строка выдачи данной команды содержит имя модуля, общий объем ЛДП, который можно использовать под ресурсы, размер занятой под ресурсы ЛДП и размер свободной ЛДП.

6. Запуск программ, использующих MPI

Запуск на исполнение MPI-программы производится с помощью команды:

```
mpirun [параметры_mpirun...] <имя_программы> [параметры_программы...]
       [-host <host>]
```

Параметры команды **mpirun** следующие:

-h - интерактивная подсказка по параметрам команды **mpirun**.

-maxtime <максимальное_время> - максимальное время счета.
От этого времени зависит положение задачи в очереди. После истечения этого времени задача принудительно заканчивается.

-np <число_процессоров> - число процессоров, требуемое программе.

-quantum <значение_кванта_времени> - параметр указывает, что задача является фоновой, и задает размер кванта для фоновой задачи.

-restart - указание этого ключа приведет к тому, что после своего завершения задача будет вновь поставлена в очередь.
Для удаления из очереди такой задачи пользуйтесь стандартной командой **mqdel,** а для ее завершения – командами **mkill** или **mterm**.

-wait - указание этого ключа приведет к тому, что в случае успешного старта задачи (либо постановки в очередь) команда **mpirun** “подвиснет” и будет ждать завершения задачи.
Для обычной задачи под завершением понимается обычное завершение или снятие со счета. Для фоновой задачи – завершение последнего кванта либо полное снятие со счета по команде **mterm**. Кроме этого, дождаться завершения задачи можно, используя команду **mwait** (см. п. 14).

-stdiodir <имя_директории> - параметр задает имя каталога стандартного ввода/вывода, в который будут записываться протокол запуска задачи, файл стандартного вывода и имена модулей, на которых запускалась задача.

-stdin <имя_файла> - параметр задает имя файла, на который будет перенаправлен стандартный ввод задачи.

-stdout <имя_файла> - параметр задает имя файла, на который будет перенаправлен стандартный вывод задачи.

-stderr <имя_файла> - параметр задает имя файла, на который будет перенаправлен стандартный вывод сообщений об ошибках задачи.

-interactive - задание этого ключа делает задачу интерактивной (см. п. 15), а также отменяет действия ключей **stdin**, **stdout**, **stderr**.

-ldmname <имя_ресурса_ЛДП> - параметр указывает на то, что задача будет использовать ресурс ЛДП с указанным именем.
Если на момент запуска задачи ресурс ЛДП с указанным именем не существует, он будет создан временным, о чем пользователь извещается специальным сообщением, например:

Warning! LDM resource ldm does not exists, it will be created temporary

Для временного ресурса пользователь обязан задать размер требуемой дисковой памяти на каждом модуле (параметр **ldmspace**). Если в очереди или в счете у пользователя уже есть задача, использующая (или требующая) временный ресурс с таким же именем, то система выдаст сообщение об ошибке, и данная задача не будет поставлена в очередь. Подробнее о ресурсах ЛДП см. п. 5.

-ldmspace <размер_ресурса_ЛДП_на_одном_модуле> - параметр имеет действие только, если указано имя ресурса ЛДП (параметр **ldmname**) и ресурс ЛДП с заданным именем не существует (т.е. для задачи требуется разовый или временный ресурс ЛДП).
Параметр **ldmspace** задает размер локальной дисковой памяти на одном модуле, требуемой под временный ресурс ЛДП. Локальная дисковая память заданного размера будет выделена на каждом модуле. Размер ресурса ЛДП задается в килобайтах. Подробнее о ресурсах ЛДП см. п. 5.

-termtime <дополнительное_время> - параметр задает дополнительное время для завершения задачи. Время задается в минутах. Подробнее о дополнительном времени см. п. 4.

-termsignal <сигнал_для_завершения> - параметр имеет действие только, если задано дополнительное время для завершения задачи (параметр **termtime**). Параметр **termsignal** задает сигнал, который будет разослан всем процессам задачи в качестве предупреждения о предстоящем завершении. Формат задания сигнала должен соответствовать команде **kill**. Пользователь должен самостоятельно определить обработчик сигнала в своей программе. Подробнее о дополнительном времени см. п. 4. Подробнее о сигналах см. описание системных вызовов **kill()** и **signal()**.

-transform <имя_командного_файла>
При запуске задачи происходит преобразование списка выделенных задаче вычислительных модулей в формат среды программирования MPICH. Параметр **имя_командного_файла** задает командный файл, который выполнит указанное преобразование**.** Необходимо учесть, что при вызове данный командный файл получит два параметра: файл со списком узлов выделенных задаче и полное имя файла запускаемой программы. Подробнее форматы этих файлов и описание ключа **transform** можно найти в разделе “Интерфейс между средой для разработки параллельных программ и системой управления прохождением задач” Руководства администратора.

Ключ **-host** описан в п. 17.

Удачно запущенная задача получает определенный номер, который добавляется к имени задачи. Это позволяет пользователю запускать одновременно несколько экземпляров задачи с одним и тем же именем – система присвоит каждому экземпляру задачи уникальный номер. Для каждого экземпляра будет создан отдельный каталог стандартного ввода/вывода. По завершении задачи ее номер “освобождается” и будет использован повторно.

При удачном старте система выдаст пользователю следующую информацию:

- имена свободных узлов в системе на момент запуска задачи;
- имена выделенных под задачу узлов;

- сведения о принятых системой установках по умолчанию.

Завершает выдачу сообщение об удачном старте задачи, причем в сообщении указывается присвоенный задаче номер, например:

```
Task “test.1” started successfully
```

Может случиться так, что задача не будет запущена сразу, а поставлена в очередь. В этом случае реакция системы будет следующей:

```
Task “test.1” was queued
```

В процессе работы команды **mpirun** образуется файл паспорта задачи **<имя_программы>.img,** формат которого описан ниже. Данный файл может быть использован в команде **mrunf.**

7. Формат файла-паспорта задачи и запуск задачи через него

Пользователь может оформить паспорт задачи в виде отдельного файла и потом запустить задачу с этим паспортом специальной командой. Формат файла-паспорта задачи следующий:

```
# Это комментарий
# Следующая строка - название секции
[General]
# Далее идет содержимое секции
# Следующая срока определяет имя задачи
task_name = testmod
# Следующая срока определяет имя каталога
# стандартного ввода/вывода
host_directory = /usr/people/lacis/testmod/work
# Следующая срока определяет необходимое число
# процессоров для выполнения задачи
# Если указывается значение any, то число процессоров
# должно быть определено при старте задачи
cpu_count = any
# Следующая строка - название секции
# Секция не обязательная
[TimeRequest]
# Время (по максимуму), необходимое для выполнения задачи,
# указывается в минутах
limit = 240
# квант для фоновых задач
quant = 20
# дополнительное время для завершения задачи
# указывается в минутах
term_time = 20
# сигнал, который необходимо послать всем процессам задачи,
# как предупреждение о снятии со счета.
# Формат задания сигнала должен соответствовать
# формату задания сигнала в команде kill
term_signal = USR1
#
# Следующая строка - название секции
[Redirections]
# Имя файла стандартного вывода задачи
stdout = /home/user/test.out
# Имя файла стандартного ввода задачи
stdin = /home/user/test.in
# Имя файла стандартного вывода ошибок задачи
stderr = /home/user/test.err
# Параметр может принимать значения yes и no.
```

```
# Задание yes означает, что задача интерактивная (см. п. 15)
# Задание yes отменяет значения остальных параметров секции
interactive = no
#
# Следующая строка - название секции
# Секция описывает ресурс ЛДП, требуемый для задачи
# Подробнее о ресурсах ЛДП см. п. 5
[LocalDisks]
# Имя ресурса ЛДП, требуемого для задачи
ldm_name = test.ldm
# Количество модулей под ресурс ЛДП (для разового ресурса)
ldm_nodes = 8
# Размер разового ресурса ЛДП на одном модуле (в килобайтах)
ldm_space = 4000000
#
# Следующая строка - название секции
# Секция обязательная
[Batch]
# Далее идет содержимое секции
# Здесь помещается текст командного файла,
# который будет выполнен на нулевом по счету узле,
# среди выделенных для задачи
```

Сформировав, таким образом, паспорт задачи и сохранив его в файле, Вы можете запустить задачу с указанным в паспорте именем с помощью команды:

mrunf [-fo] <имя_файла-паспорта>

При запуске система проверяет корректность данных паспорта. При обнаружении ошибок выдается сообщение пользователю, а запуск останавливается.

Если пользователь задал в паспорте значение **any** для числа процессоров или номера консольного вывода, то система попросит пользователя явно указать значения этих параметров перед запуском задачи.

Задание ключа **-fo** (formal output) позволяет получить “формальную” выдачу. При успешном запуске (постановке в очередь) задачи, если задан ключ **-fo**, на стандартный вывод печатается полное имя задачи, включая полученный номер и имя логической системы.

8. Получение информации о запущенных задачах

Информация о запущенных пользователем задачах можно получить с помощью команды:

mps [имя_задачи.номер_задачи]

При отсутствии параметра на экран будет выдан список всех запущенных пользователем и работающих или находящихся в очереди на момент выдачи команды задач. Если задача находится в очереди, это будет отмечено словом **queued** рядом с именем задачи.

Параметром для команды служит имя задачи и – через точку – ее номер. При задании параметра система выдаст информацию о задаче – время старта (и завершения, если задача уже завершилась), имена узлов, на которых выполняется задача, номер процесса-менеджера задачи.

Для получения полного (включающего имя логической системы и номер) имени последней поставленной в очередь (запущенной) задачи можно воспользоваться командой:

```
mlt
```

При успешном выполнении команда напечатает на стандартный вывод полное имя последней поставленной в очередь (запущенной) задачи. Команда **mlt** выдаст имя задачи, даже если задача на момент выполнения команды уже завершилась или была удалена из очереди.

9. Завершение запущенной задачи

Завершить запущенную задачу можно командой:

```
mkill [имя_задачи.номер_задачи]
```

Параметром для команды служит имя задачи и – через точку – ее номер. Данная команда допускает задание в качестве параметра маски Unix-формата с использованием символов-джокеров. По этой команде будут завершены все задачи, имена которых удовлетворят заданной маске. Завершить все свои задачи можно командой:

```
mkill '*'
```

При отсутствии параметра пользователю будет выдан список всех запущенных задач и предложено ввести номер (по списку) той задачи, которую нужно завершить. Перед завершением задачи в этом случае будет задан вопрос о полном завершении задачи. Полное завершение фоновой задачи означает ее завершение без возможности повторов. В противном случае завершится только текущая итерация, и, если заказанное время не исчерпалось, задача будет запущена вновь. Это же замечание относится к команде **mkill** с заданным параметром – команда производит завершение лишь текущей итерации.

Безусловное полное завершение задачи происходит по команде

```
mterm [имя_задачи.номер_задачи]
```

Значение параметра данной команды аналогично **mkill**, только **mterm** вызывает полное завершение всех задач, имена которых удовлетворят заданной маске. Вызов **mterm** без параметров аналогичен вызову **mkill** без параметров.

Следует отметить, что команды **mkill** и **mterm** лишь сообщают системе запуска о необходимости завершить задачу, но реально не дожидаются ее завершения. После выполнения данных команд задача может находиться в системе еще довольно продолжительное время, которое нужно системе для корректного освобождения вычислительных ресурсов. Для ожидания завершения задачи нужно использовать команду

```
mcancel <имя_задачи.номер_задачи>
```

Заметим, что данная команда безусловно удаляет задачу из системы запуска, вне зависимости от того, находится она на выполнении или в очереди. Команда **mcancel** не закончит своего выполнения до тех пор, пока задача не покинет систему.

10. Просмотр стандартного вывода задачи

Стандартный вывод (печать на экран) задачи можно просмотреть с помощью команды (исключение составляют задачи, запущенные под управлением отладчика TotalView, подробнее см. п. 18):

```
mout [имя_задачи.номер_задачи [что_выдавать]]
```

При отсутствии параметров будет предложен выбор из списка запущенных задач. При отсутствии параметра **что_выдавать** будет предложен вопрос. В качестве ответа можно задать либо **out**, что означает выдачу стандартного вывода, либо **err**, что означает просмотр стандартного вывода сообщений об ошибках, либо **log**. В случае задания **log** на экран будет выведен стандартный вывод специального системного процесса-менеджера задачи.

Если Вы задали параметры команде **mout**, то она выдаст вам запрошенную информацию, даже если задача уже завершилась на момент вызова команды.

Прервать выдачу команды **mout** можно нажатием клавиш **Ctrl-C**. Снятие вывода **НЕ** означает снятие задачи – задача продолжит счет.

При запуске задачи система создаст в указанном в паспорте каталоге стандартного ввода/вывода подкаталог, имя которого будет совпадать с именем запущенной задачи (включая номер через точку). В этом подкаталоге и будут непосредственно размещены файлы стандартного ввода/вывода задачи. Файл стандартного вывода имеет имя **output**, стандартного вывода сообщений об ошибках – **errors**, стандартного ввода – **input**. Файл стандартного вывода процесса-менеджера имеет имя **manager.log**. Система сама не уничтожает эти файлы и подкаталог, зато перезаписывает их, если задача будет запущена повторно с этим же номером. После завершения задачи данные файлы полностью доступны пользователю.

11. Получение информации о свободных процессорах

Сколько на данный момент времени в системе имеется свободных процессоров можно узнать с помощью команды:

```
mfree
```

12. Команды работы с очередями

12.1 Постановка в очередь

Постановка задачи в очередь осуществляется командами запуска задачи **mrunf** или **mpirun.** По команде **mps** можно посмотреть все запущенные задачи, при этом стоящие в очереди задачи будут помечены атрибутом **queued**.

12.2 Удаление задачи из очереди

По команде **mqdel** задача удаляется из очереди:

```
mqdel <имя_задачи.номер_задачи>
```

Кроме этого для удаления задачи из очереди можно использовать команду

```
mcancel <имя_задачи.номер_задачи>
```

Заметим, что данная команда безусловно удаляет задачу из системы запуска, вне зависимости от того, находится она на выполнении или в очереди. Команда **mcancel** не закончит своего выполнения до тех пор, пока задача не покинет систему.

12.3 Просмотр очереди

Просмотр очереди осуществляется по команде

mqinfo

Данная команда не имеет параметров. Информация, выдаваемая командой, выглядит примерно следующим образом:

```
Current time:  Mon Mar 22 13:10:07 2003
Queue state at Mon Mar 22 13:06:09 2003
Current schedule: 2
Accumulation period: 168 hours

--- 22.03.00 07:30: R=1 total – 32 debug – 16/15, packet – 300, priority scale - (120 300)
      img1.3 : guest     6.3  0*  15(tsk1)  150/15+1      16      >:03.22.02 18:03
-- queue --
      img1.4 : guest     6.5 0*  15(tsk1)   150/15+1      b~16    <:03.22.02 14:33
      tnet.7 : ant          1*   10  15/15      ~10   <:Mon Mar 22 13:31:09 2000
--- 22.03.00 19:00: R=5 total – 32 debug - 0/15, packet - 600, priority scale - (10000)
      tnet.3 : ant          1*   24  500/500   ~312   <:Mon Mar 22 20:31:09 2000
--- 23.03.00 07:30: R=1 total – 32 debug - 16/15, packet - 300, priority scale - (120 300)
      tnet.1 : ant          1*   1   15/15     A~~~   <:Mon Mar 20 20:31:09 2000

      Free: 6 proc   Available: 17  Locked: 9

   ant: Inf:3 R=1 run=0 wait=0/5 all=0/7   *1.00 Sr/Sf=10/26
   gns: Inf:3 R=1 run=1 wait=0/5 all=1/7   *1.00 Sr/Sf=300/0
```

В заголовке выдачи сообщается, когда было в последний раз перезаписано состояние очереди, а также (**Current schedule**) – номер текущего расписания. Если администратор отключил часть процессоров от системы управления очередью, то в той же строке будет показано их количество. Кроме того, выдается учетный период (**Accumulation period**), за который производится суммирование времени счета завершившихся задач одного пользователя.

Если очередь блокирована (заморожена), то это отображается словом **Hold** при выдаче:

```
Queue state at Wed Mar 22 12:40:07 2000
Current schedule: 2      Hold
Accumulation period: 168 hours

--- 22.03.00 07:30: R=3 total – 32 debug – 16/15, packet – 300, priority scale - (120 300)
--- 22.03.00 19:00: R=2 total – 32 debug - 0/15, packet - 600, priority scale - (10000)
--- 23.03.00 07:30: R=2 total – 32 debug - 16/15, packet - 300, priority scale - (120 300)
          tnet.7 : ant          1*   10  15/15      H~~~   <:Mon Mar 22 13:31:09 2000
          tnet.3 : ant          1*   24  500/500    H~~~   <:Mon Mar 22 20:31:09 2000
          tnet.1 : ant          1*   1   15/15      H~~~   <:Mon Mar 20 20:31:09 2000
      ..............................
```

Если блокирована система управления ресурсами ЛДП, то появляется индикатор **NoLDM** в той же строке, что и **Hold**. При этом старт задач, использующих ресурсы ЛДП, откладывается до принятия соответствующих мер администратором системы.

Информация о режимах начинается с ‘**---**‘ и содержит:

- дата и время включения режима;

- после **R=** - шкалу классов режимов, которым принадлежит данный ("**1**" означает нулевой класс, "**fc**" — классы со 2-го по 7-й включительно);
- после **total** – общее число планируемых процессоров;
- после **debug –** количество процессоров, отведенное под отладочные задачи/максимальное время (в минутах) для отладочных задач;
- после **packet –** максимальное время (в минутах) для пакетных задач;
- после **priority scale –** шкала приоритетов.

Следующий пример демонстрирует режим, который включается 22 марта 2000 г. в 7 часов 30 минут и имеет следующие параметры: всего планируется 32 процессора, максимальное время для отладочных задач – 15 минут, для пакетных – 300 минут, при этом под отладочные задачи отведено 16 процессоров, эти 16 процессоров не могут занимать пакетные задачи, шкала приоритетов определяет два приоритета – высокий для пользователей, считавших менее 180 минут, и низкий - для всех остальных. Данный режим относится лишь к нулевому классу (**R=1**), т.е. только те пользователи, которым разрешено считать в режимах нулевого класса, могут запускать здесь свои задачи. У администратора есть возможность создать режим, в котором всем пользователям режимов других классов дано разрешение запускать отладочные задачи. В настоящее время этот флаг при выдаче информации о режимах не отражается.

```
---22.03.00 07:30: R=1 total–32 debug-16/15, packet–300, priority scale-(180)
```

Информация о считающихся задачах содержит:

- имя задачи;
- имя пользователя;
- составной индикатор порядка выделения ресурсов: номер (0 - 6) очереди при выделении ресурсов и (через '.') номер в списке задач — чем меньше получившее десятичное число, тем раньше планировщик пытался выделить ресурс данной задаче;
- количество оставшихся повторов счета;
- количество процессоров, занимаемых задачей;
- заключенное в скобки имя ресурса ЛДП, если он используется задачей;
- остаток времени счета в минутах;
- квант счета в минутах;
- число минут, зарезервированных для завершения, если есть;
- количество минут до предполагаемого завершения задачи;
- предполагаемое время завершения (после **>:**).

Следующий пример демонстрирует считающуюся задачу **img1.3** пользователя **guest**. Задача занимает 15 процессоров, используя ресурс локального дискового пространства с именем **tsk1**, считается последний раз (более повторов не будет), оставшееся время счета – 150 минут, задача фоновая с квантом 15 минут, будет перепланирована системой через 15 минут 2 августа 2003 года в 18 часов 3 минуты. Задаче будет предоставлена 1 минута для завершения очередного кванта, если следующий квант может быть выдан задаче лишь после некоторого перерыва.

```
img1.3 : guest     6.3  0*  15(tsk1)  150/15+1      16    >:03.08.02 18:03
```

Информация о стоящих в очереди задачах отличается от информации о считающихся задачах тем, что вместо количества минут до завершения выдается

код (причины) ожидания и количество минут до предполагаемого старта (разделенные символом **~**), а вместо времени завершения — время постановки в очередь (после **<:**). Например:

```
img1.4 : guest      6.5 0*  15(tsk1)  150/15+1      b~16    <:03.08.02 14:33
```

Следующая выдача говорит о том, что задача находится в процессе запуска:

```
img1.3 : guest      6.3  0*  15(tsk1)  150/15+1      starting
```

Следующая выдача говорит о том, что задача находится в процессе завершения:

```
img1.3 : guest      6.3  0*  15(tsk1)  150/15+1      finishing
```

Задача может быть заморожена оператором или системой. В этом случае вместо времени, оставшегося до старта будет проставлено **~~~**, например:

```
img1.2 : ant        6.7  1*  16 300/300    A~~~   <:03.08.02 14:03
```

Как видим, индикатор блокированной задачи (**~~~**) предваряется буквенным кодом, отображающим причину блокировки или ожидания задачи. Ниже приводится расшифровка буквенных кодов.

Коды причин ожидания вмешательства со стороны администратора или пользователя:

A — задача заблокирована администратором;
D — задача отложена до завершения режима отладки, инициированного каким-либо пользователем (не поддерживается системой запуска);
H — очередь в целом заблокирована администратором;
L — задача с заказом ресурса ЛДП ждет разрешения работы с (данным ресурсом) ЛДП;
O — рассматриваемых планировщиком режимов не хватает, чтобы сделать вывод о возможности или невозможности запуска задачи в одном из них;
P — в текущем расписании для данного пользователя запрещено запускать задачи с таким числом процессоров;
R — резервная задача не может быть запущена ни в одном из рассматриваемых планировщиком режимов;
T — квант задачи превышает время пакетных задач всех рассматриваемых планировщиком режимов;
U — задача заблокирована пользователем.

Коды причин ожидания изменения ситуации (очередного цикла планирования):

e — старт отложен из-за несущественной ошибки при запуске;
l — старт отложен из-за ошибки при запросе ресурса ЛДП.

Коды причин ожидания, приводящих к откладыванию старта на вполне определенный срок:

b — занятость процессоров считающимися задачами;
d — при старте задача не оставила бы в одном из режимов нужного для отладочных задач количества процессоров на слишком длительное время;

m — в одном из режимов пользователю запрещено запускать задачи;
p — при запуске задача отодвинула бы старт более приоритетной задачи;
t — квант задачи превышает время пакетных задач одного из режимов;
w — ждет наступления определенного времени (пока не реализовано).

Имеется информация о свободных процессорах в виде:

```
Free: 6 proc   Available: 17  Locked: 9
```

В этой же строчке выдается информация о количестве доступных для счета (**Available**) и заблокированных (**Locked**) процессоров.

Затем выдается суммарная информация по пользователю:

- login-имя;
- информация о запрете (**Block:**) счета (**Q**) или запуска (**R**) задач (если он установлен администратором);
- степень подробности выдачи информации (**Inf:**);
- шкала классов режимов (**R=**), в которых разрешено запускать задачи, в виде 16-го числа;
- после **run** - количество считающихся задач;
- после **wait** - количество ждущих задач/ограничение на количество ждущих задач;
- после **all** - суммарное количество задач в очереди/ограничение на суммарное количество задач в очереди;
- коэффициент умножения при вычислении цены задачи;
- после **Sr/Sf** - суммарная стоимость считающихся задач/суммарная стоимость закончившихся задач;

Например,

```
guest: Inf:2 run=1 wait=0/1 all=1/1   *1.00 Sr/Sf=15/14
u1125: Inf:2 R=1 run=0 wait=0/5 all=0/7   *1.00 Sr/Sf=0/365
u1310: Block:RQ Inf:2 R=1 run=0 wait=1/5 all=1/7   *1.00 Sr/Sf=0/0
u0319: Inf:2 R=3 run=0 wait=0/5 all=0/7   *1.00 Sr/Sf=0/88
u1305: Block:R Inf:2 R=1 run=1 wait=2/5 all=3/7   *1.00 Sr/Sf=0/0
```

Здесь показано, что пользователю **u1310** запрещено ставить задачи в очередь, задачи его и пользователя **u1305**, стоящие в очереди, заморожены; всем пользователям информация выдается со степенью подробности **2** (т.е. вся за исключением имен задач других пользователей и информации о пользователях, которые в учетный период не считали и не имеют задач в очереди); всем пользователям разрешено запускать задачи в режимах класса **0**, а пользователю **u1319** еще и в режимах класса **1**.

Специальное имя пользователя “**::THE_OTHERS::**” зарезервировано для выдачи информации о настройках обычных пользователей. Вообще говоря, настройки выдаются вместе с текущей информацией, но для тех пользователей, которые еще не считали, текущая информация не выдается.

После информации о настройках “**::THE_OTHERS::**” будут выданы настройки для специально выделенных пользователей, но только для тех, кто не имеет на текущий момент ни задач в очереди, ни накопленных сумм счета. Информация о настройках специальных пользователей имеющих задачи в очереди или

накопленные суммы счета выдается (как и для обычных пользователей) вместе с информацией о текущих значениях до строки с “**::THE_OTHERS::**”.

13. Определение статуса задачи

Для определения статуса задачи (“в очереди”, “запущена”, “завершилась”) можно использовать команду

mqtest <имя_задачи>

Данная команда определяет статус задачи и печатает соответствующее сообщение пользователю. Кроме этого, при задании ключа **–rcode** возможно получить статус задачи через код возврата (подробнее см. п. 20).

14. Ожидание завершения задачи

mwait <имя_задачи>

Команда позволяет дождаться завершения запущенной задачи. Данная команда “подвисает” до тех пор, пока задача полностью не завершится и не покинет систему. Для обычной задачи этот момент наступает во время завершения. Для фоновой задачи – во время завершения последнего кванта либо полного снятия со счета по команде **mterm**. Кроме этого, дождаться завершения задачи можно, задав ключ **–wait** в команде **mpirun** (см. п. 6).

15. Работа с интерактивными задачами

При запуске задачи в команде **mpirun** может быть указан ключ **interactive** (см. п. 6), а в секции **[Redirections]** файла-паспорта задачи - задан соответствующий параметр (см. п. 7). Запущенная таким образом задача получает статус *интерактивной*, и для работы с ней необходимо соблюдать некоторые правила.

После прохождения через очередь и запуска интерактивная задача при попытке чтения со стандартного ввода приостанавливается до тех пор, пока пользователь не выполнит команду

mattach [имя_задачи]

Если имя задачи не задано, система предложит выбрать его из списка запущенных. Команда **mattach** связывает свои стандартные потоки ввода, вывода и ошибок со стандартными потоками интерактивной задачи. После установления связи задача продолжит выполнение, при этом пользователь получает возможность в диалоге вводить требуемые задаче данные и просматривать ее вывод.

ВНИМАНИЕ. Для интерактивной задачи система игнорирует заданные перенаправления стандартных потоков (ввода, вывода и ошибок). Для перенаправления стандартных потоков своей интерактивной задачи пользователь должен перенаправить стандартные потоки команды **mattach**. Также следует

обратить внимание на то, что для интерактивных задач команда **mout** позволит просмотреть лишь стандартный вывод процесса-менеджера задачи.

Интерактивная задача считается выполняющейся с момента своего запуска, а не с момента выполнения пользователем команды **mattach**. Если пользователь не выполнит **mattach**, то по истечении заказанного времени задача будет снята. В этом случае, а также тогда, когда интерактивная задача сама завершается до выполнения пользователем команды **mattach**, весь произведенный задачей вывод в потоки стандартных вывода и ошибок будет записан в файлы **output** и **errors** в каталоге стандартного ввода/вывода (так же, как и в случае обычной задачи, см. п. 10).

ВНИМАНИЕ! Так же, как и для команды **mout,** завершение команды **mattach** по нажатию **Ctrl-C** не приводит к завершению самой задачи. Команда **mattach** может быть выполнена и завершена несколько раз, при этом работа с отлаживаемой задачей каждый раз будет продолжаться с места, на котором была завершена предыдущая команда **mattach**. Вместе с этим одновременно для одной и той же задачи пользователь может выполнить лишь одну команду **mattach**.

16. Поиск задачи в системе по данным паспорта

mfind <имя_секции> <имя_параметра> <маска_значения>

Команда осуществляет поиск задачи в системе запуска по данным паспорта. Параметр **<имя_секции>** задает секцию паспорта задачи, **<имя_параметра>** - параметр в данной секции, **<маска_значения>** - диапазон значений указанного параметра. Команда **mfind** печатает на стандартный вывод полные имена задач, паспорта которых соответствуют критериям поиска. Например, получить список всех задач, которые используют ресурсы ЛДП с именами, начинающимися на букву **g**, можно с помощью следующей команды:

```
mfind Local_Disks ldm_name g*
```

17. Пользовательский конфигурационный файл системы запуска

Чтобы система запуска задач работала правильно, необходимо следить за правильными ее настройками. Эти настройки для каждого пользователя сохраняются в файле **.crunmvs**, который должен находиться в домашнем каталоге пользователя. Формат файла следующий:

```
# Это комментарий
# Следующая строка - название секции
[General]
# Далее идет содержимое секции
#
# Следующая срока определяет имя каталога
# для временных файлов, где системе можно мусорить)
# Если имя не задано или задано неправильно
# система будет использовать каталог /tmp
tmp_directory = ~/runmvs
# Следующая срока определяет имя управляющей ЭВМ
host = irc-1
#
# Следующая строка - название секции
# Данная секция определяет значения
# ключей команды mpirun, используемых  по умолчанию
```

```
[Mpirun]
# Следующая строка определяет значение кванта времени для
# фоновой задачи
quantum=0
# Задан ноль - это означает,
# что задача не является фоновой
#
# Следующая строка определяет
# максимальное время счета задачи
maxtime=300
# Задача может выполняться не более 300 минут
#
# Следующая строка определяет каталог
# стандартного ввода/вывода
stdiodir=`pwd`
# Если указано `pwd` - то создавать каталог
# стандартного ввода/вывода в текущем каталоге.
# Перед использованием этот параметр обрабатывается
# командным интерпретатором
#
# Следующая строка определяет командный файл, задающий
# распределение задач по процессорам.
# Подробнее см. п. 6 (ключи -transform).
transform=/common/runmvs/bin/t_u2s
# Следующий параметр указывается при необходимости работы
# с отладчиком totalview (см. п. 18). Значение параметра yes означает
# запуска отладчика по умолчанию, no - отладчик по умолчанию
# не запускается
totalview=no
# Следующая строка определяет необходимость рестарта
# задачи по ее окончанию
restart=0
# сигнал, посылаемый задаче для принудительного ее завершения,
# используемый по умолчанию
# 0 - нет сигнала
term_signal=0
# время на завершение задачи в минутах после получения сигнала,
# используемое по умолчанию
# 0 - нет дополнительного времени
term_time=0
# Следующая строка - название секции
# Данная секция определяет значения
# ключей команды mpirun, используемых  по умолчанию
# для работы с ресурсами ЛДП
[LDM]
# имя ресурса ЛДП, используемое по умолчанию
# " " - нет ресурса по умолчанию
ldm_name=" "
# размер в килобайтах ресурса ЛДП, используемый по умолчанию
ldm_space=1000
# число модулей в ресурсе ЛДП
ldm_nodes=10
```

18. Работа с отладчиком TotalView

Система запуска задач поддерживает работу с отладчиком TotalView, для ознакомления с которым рекомендуется изучить “Getting Started With TotalView” http://www.totalviewtech.com/Documentation/getting_started/getting_started.html

Для работы с отладчиком от пользователя требуется определить параметр **totalview** секции **[Mpirun]** пользовательского конфигурационного файла **~/.crunmvs** (см. п. 17), например:

```
[Mpirun]
# для запуска отладчика по умолчанию
totalview=yes
```

или

```
[Mpirun]
# если отладчик по умолчанию вызывать не надо
totalview=no
```

Для запуска программы под управлением отладчика (если он не используется по умолчанию) необходимо использовать ключ **-tv** команды **mpirun**, например:

```
mpirun -tv    <остальные параметры команды mpirun>
```

Передать параметры командной строки отладчику можно через переменную окружения **TOTALVIEW**. Например, для передачи параметров командной строки отладчику только в текущем запуске:

```
TOTALVIEW="totalview <параметры totalview>" mpirun -tv <параметры mpirun>
```

для передачи одних и тех же параметров командной строки в totalview при нескольких последовательных запусках программы:

```
export TOTALVIEW="totalview <параметры totalview>"
mpirun -tv  <параметры команды mpirun >
```

В приведенных примерах будет использоваться вариант отладчика с графическим интерфейсом (GUI). Для использования GUI необходимо наличие на клиентской ЭВМ работающего X-сервера (для Windows-компьютеров это может быть продукт Exeed). При старте ssh-клиента, с помощью которого производится соединение с сервером доступа МВС-1000/16, должна быть включена возможность ретрансляции (форвардинга) X-протокола.

Если все приведенные выше условия будут выполнены, то после прохождения через очередь и запуска задачи на экране клиентской ЭВМ появится X-окно отладчика.

Отладчик **Totalview** можно также использовать в режиме работы с командной строкой из терминального окна. В этом случае в переменную окружения **TOTALVIEW** вместо значения **totalview** необходимо занести значение **totalviewcli**, напрмер:

```
export TOTALVIEW="totalviewcli <параметры totalview>"
mpirun -tv  <параметры команды mpirun >
```

Запущенная таким образом задача автоматически получает статус интерактивной, и для работы с ней необходимо соблюдать правила работы с интерактивными задачами (см. п. 15). В частности, пользователь должен выполнить команду **mattach**.

19. Ключи выбора управляющей ЭВМ и логической подсистемы

Система запуска способна обслуживать одновременно несколько управляющих ЭВМ с несколькими вычислителями, подключенными к каждой из

них. Кроме этого, система может быть поделена на несколько логических подсистем, каждая из которых также будет уникально именована. Причем с точки зрения системы запуска каждый отдельный вычислитель также будет представлять логическую подсистему.

Выбор управляющей ЭВМ задается либо в пользовательском конфигурационном файле, либо с помощью ключа **-host** в любой из команд системы запуска, например:

```
mrun tnet –host beta
```

запустить задачу **tnet** на вычислителе, подключенном к управляющей ЭВМ с именем **beta**

```
mfree –host cascade
```

выдать, сколько свободных процессоров на вычислителе, подключенном к управляющей ЭВМ с именем **cascade**.

Если к одной управляющей ЭВМ подключено несколько многопроцессорных вычислителей, то каждый такой вычислитель трактуется как отдельная система, имя которой указывается в качестве параметра с ключом **-s**:

```
-s <имя_системы>
```

Данная опция применима ко всем командам, **ВНИМАНИЕ!** При формировании имени запущенной задачи имя системы будет добавлено в начало! Например, следующий запуск

```
mrun tnet –s sys
```

приведет к тому, что будет запущена задача с именем **sys.tnet.1**. Будьте внимательны!

По умолчанию будет использована система, имя которой прописано в качестве значения параметра **default_system** секции **[General]** пользовательского конфигурационного файла **~/.crunmvs**. Например:

```
# Это комментарий
[General]
# Далее - имя управляющей ЭВМ
host = irc-1
# Далее - имя системы на этой управляющей ЭВМ
default_system = sys
```

О существующих системах и их именах справляйтесь у системного администратора. В ближайшее время планируется разработать команду, которая будет показывать существующую многопроцессорную конфигурацию, включая имена всех управляющих ЭВМ и систем, им принадлежащих.

20. Коды возврата команд системы запуска

Все команды системы запуска по умолчанию имеют код возврата 0 (если завершение нормальное) или 1 (если авария). Указание ключа **-rcode** в команде

заставит эту команду возвращать специальный код возврата. Специальный код возврата имеет место в следующих командах:

mpirun (mrunf) – возвращает номер запущенной (поставленной в очередь) задачи (действительно для номеров меньше 128), либо значение от 128 до 255, если авария.

mqtest – возвращает 1, если задача стоит в очереди, 0 – если задача запущена, 2 - если задача завершена, либо значение от 128 до 255 – если такой задачи нет или авария.

21. Команда получения вычислительных модулей

Данная команда необходима в тех случаях, когда пользователю требуется вычислительный ресурс (один или несколько вычислительных модулей) на определенное время, но при этом нет необходимости запускать конкретную задачу. Команда получения вычислительных модулей имеет следующий формат:

```
getnodes -np <число_модулей> -maxtime <время>  <имя_запроса>
```

где **число_модулей** – число требуемых вычислительных модулей (**не процессоров!**);
время – время, на которые требуется получить указанное число модулей;
имя_запроса – имя, которое будет присвоено запросу на требуемый вычислительный ресурс.

При выполнении этой команды запрос на требуемый ресурс будет поставлен в очередь, как обычная задача. При этом к имени запроса будет через точку добавлен уникальный номер. Как и для обычной задачи будет сформирован каталог стандартного ввода/вывода. После того, как запрос отстоит в очереди, пользователю будет выделено указанное число модулей на указанное время. При этом в файл стандартного вывода будет помещена информация о выделенных модулях. С момента выделения модулей любой из них становится доступен пользователю с помощью команд **rlogin** или **rsh**. Завершить работу с модулями можно командой

```
mkill <имя_запроса.номер>
```

22. Библиотека интерфейсных вызовов (API) для организации контрольных точек (КТ)

Для облегчения работы пользователя при сохранении и восстановлении расчетов с помощью КТ предлагается использовать специальную библиотеку интерфейсных вызовов (далее - API). Подробное описание находится в документе “Библиотека интерфейсных вызовов (API) для организации контрольных точек (КТ)”.