УДК 004.75:004.451.42:004.052.3

Анализ средств обеспечения отказоустойчивости системы управления ресурсами PBS/TORQUE[1]

А. В. Ефимов, К. В. Павский

Работа посвящена проблеме отказоустойчивого функционирования распределенных вычислительных систем под управлением PBS/TORQUE при выполнении параллельных MPI-программ. Рассматривается влияние событий сбоев и отказов на возможность завершить выполнение параллельной программы. В первой части работы представлен теоретический анализ средств обеспечения отказоустойчивости системы управления ресурсами PBS/TORQUE. Во второй части представлена функциональная модель обработки отказов для планировщика Maui.

Ключевые слова: распределённые вычислительные системы, система управления ресурсами, PBS/TORQUE, отказоустойчивость, анализ, сбои и отказы.

1. Введение

Распределённые вычислительные системы (ВС) относятся к перспективным средствам обработки информации [1]. Основным функциональным элементом распределённой ВС является элементарная машина (ЭМ). Структура ЭМ допускает варьирование от процессорного ядра до конфигураций, включающих многоядерные процессоры и специализированные ускорители (например, GPGPU). Количество ЭМ в распределённых ВС может варьироваться в широких пределах – от десятков до сотен тысяч.

Основное назначение ВС – это решение пользовательских задач, представленных параллельными программами. Управление распределёнными ВС осуществляется специализированным системным программным обеспечением, реализующим функции управления пользовательскими задачами и ресурсами системы.

Время наработки на отказ в связи с большемасштабностью современных распределённых ВС может достигать десятков часов [2, 3]. Поэтому актуальной является проблема обеспечения надежного и живучего функционирования распределенной ВС [4, 5]. В работе [6] определены пять направлений исследований, важных для создания надежных ВС экзафлопсного уровня производительности:

- классификация аппаратных отказов;
- разработка моделей обработки отказов;
- совершенствование средств прогнозирования, локализации, обнаружения, уведомления и исправления отказов;
- программирование абстракций для отказоустойчивости;
- стандартизация средств оценки методов обеспечения отказоустойчивости.

Настоящая работа посвящена направлению по разработке моделей обработки отказов.

Под отказом будем понимать событие, при котором ЭМ теряет способность выполнять заданные функции по переработке информации [1]. Принято выделять частичный и полный

[1] Работа выполнена в рамках проекта ГЗ 0306-2016-0018 и при поддержке РФФИ (грант № 16-07-00712).

отказы (ГОСТ Р 27.002-2009). Полный отказ – это событие, в результате которого выполнение параллельной программы не может быть продолжено без выполнения процедуры восстановления. Частичный отказ аппаратурных или программных компонентов предполагает возможность корректного завершения выполнения параллельной программы, хоть и с потерей производительности. Под восстановлением будем понимать событие, в результате которого ЭМ переходит из неработоспособного состояния в работоспособное и приобретает способность выполнять заданные функции по переработке информации. Сбой – самоустраняющийся отказ.

В данной работе рассматриваются результаты анализа системы управления ресурсами (СУР) PBS/TORQUE [7] (с открытым исходным кодом до 06.2018 г.) на предмет функциональности и эффективности имеющихся средств обеспечения отказоустойчивости распределенных ВС. PBS/TORQUE является одной из популярных СУР в настоящий момент.

2. Архитектура кластера под управлением PBS/TORQUE

Рассмотрим архитектуру PBS/TORQUE с целью установить, от каких компонентов зависит корректность выполнения параллельной программы.

На рис. 1 представлена функциональная схема ВС под управлением PBS/TORQUE.

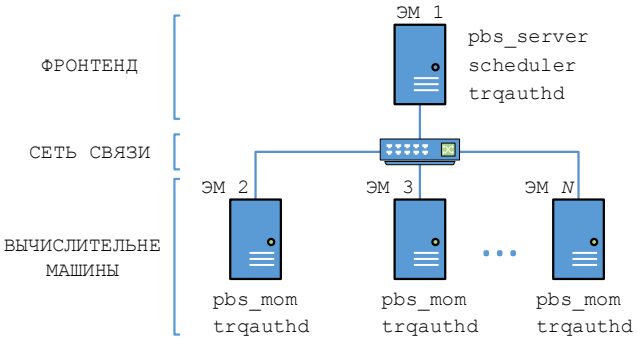

Рис. 1. Функциональная схема СУР PBS/TORQUE

В PBS/TORQUE предусмотрены центральный и локальные компоненты. Центральный компонент `pbs_server` устанавливается на головной ЭМ (FrontEnd, фронтент) и обслуживает очередь задач, взаимодействует с планировщиком ресурсов scheduler, а также при помощи локальных компонентов осуществляет координацию работы и мониторинг состояния ЭМ ВС. В качестве планировщика ресурсов в СУР используется либо встроенный компонент `pbs_sched`, либо внешний, например, Maui Cluster Scheduler [8]. Каждая вычислительная ЭМ ВС оснащается локальным компонентом `pbs_mom` – Machine Oriented Miniserver, который осуществляет управление ресурсами данной ЭМ. Локальный компонент trqauthd выполняется на всех ЭМ ВС и используется для авторизации подключений пользователей к `pbs_server`.

На рис. 2 представлена архитектура ВС под управлением PBS/TORQUE.

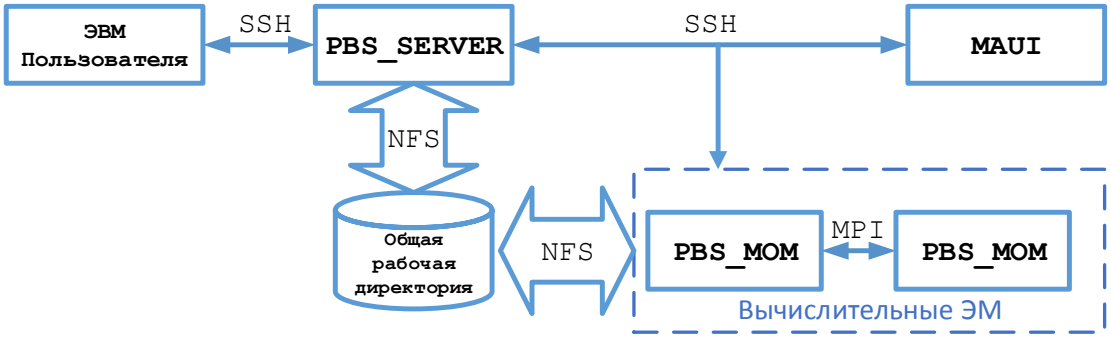

Рис. 2. Архитектура кластера под управлением PBS/TORQUE

В СУР PBS/TORQUE для выполнения и восстановления программы в случае отказа требуется обработка служебных файлов, которые располагаются в специализированной директории (обычно `/var/spool/torque`). В частности, в служебных файлах PBS/TORQUE находится информация о наборе ЭМ, выделенном СУР для выполнения параллельной программы, и об отображении процессов параллельной программы по ЭМ набора. Также выполнение программы зависит от доступности файлов входных и выходных данных параллельной программы. В связи с чем ошибки выполнения параллельной программы могут быть связаны со сбоем файловой системы, доступностью общей рабочей директории, исправностью сетевого интерфейса и т.п.

3. Теоретический анализ средств обеспечения отказоустойчивости

В данном разделе приведены результаты анализа исходных кодов PBS/TORQUE версии 6.1.2, Maui версии 3.3.1 и руководства администратора [9, 10]. Рассмотрены вопросы отказоустойчивости центральных и локальных компонентов, подсистемы обслуживания и возобновления задач, а также сопутствующей инфраструктуры ОС Linux.

Как правило, информация о сбоях и отказах хранится в файлах журнала логов `/torque/server_log` и `/torque/mom_log` соответственно для `pbs_server` или `pbs_mom`. В случае если в логах отсутствует информация об ошибках, рекомендуется запустить `pbs_server` или `pbs_mom` в GDB (GNU Debugger) [11]. Для этого необходимо экспортировать переменную окружения `PBSDEBUG=yes` и запустить программу в `gdb`.

3.1. Отказоустойчивость центрального компонента

Отказоустойчивость центрального компонента PBS/TORQUE обеспечивается возможностью одновременного выполнения на разных ЭМ нескольких резервных экземпляров `pbs_server` в режиме высокой готовности (high availability mode). Настройки сервера и состояние очередей задач для всех экземпляров `pbs_server` сохраняются в каталоге `torque/server_priv`, расположенном в общей файловой системе. Имена всех ЭМ, на которых выполняется `pbs_server`, должны быть перечислены в файле `torque/server_name` через «**,** » на всех ЭМ ВС.

Все экземпляры `pbs_server` должны быть запущены с ключом «`--ha`» командной строки, которая позволяет серверам работать одновременно. Стартовавший первым `pbs_server` становится основным (primary) сервером и блокирует файл `torque/server_priv/server.lock`. Все резервные серверы после запуска будут крутиться в цикле и проверять доступность файла блокировки. В случае отсутствия активности основного сервера блокировка с файла `server.lock` снимается и он становится доступным для резервных серверов, один из которых перехватывает управление ВС.

Интервалы обновления и проверки файла блокировки можно настроить с помощью опций сервера `lock_file`, `lock_file_update_time` и `lock_file_check_time`. Опция `lock_file` позволяет администратору ВС изменить расположение файла блокировки `server.lock`. Опция `lock_file_update_time` позволяет установить время обновления файла блокировки основным сервером (по умолчанию – 3 секунды). Опция `lock_file_check_time` позволяет установить интервал времени проверки файла блокировки резервными серверами (по умолчанию – 9 секунд).

Резервирование планировщика может быть реализовано с помощью ключа «`-l`» при запуске `pbs_server` из командной строки. Ключ «`-l`» может использоваться несколько раз с аргументами `имя_хоста_планировщика` и `номер_порта`, разделенными через «`:`». Например, `pbs_server -l <scheduler_host1:port> -l <scheduler_host2:port>`.

3.2. Отказоустойчивость локальных компонентов

В PBS/TORQUE имеется возможность запускать несколько локальных компонентов `pbs_mom` (multi-mom) на одной ЭМ. Такая возможность позволяет имитировать ВС с большим количеством ЭМ и тестировать компоненты PBS/TORQUE. Использование multi-mom для обеспечения отказоустойчивости не предусмотрено.

В PBS/TORQUE предусмотрена возможность проверки работоспособности каждой ЭМ. Данная функция настраивается через файл `torque/mom_priv/config` с помощью параметров `node_check_script` и `node_check_interval`. В первом необходимо указать полный путь до файла, в котором находится сценарий (скрипт) для проверки работоспособности узла. Во втором параметре указываются временной интервал между проверками в секундах (по умолчанию – 45 сек.) и/или через «,» события `jobstart` (проверка перед началом выполнения программы) или `jobend` (проверка после завершения программы).

Сценарий проверки работоспособности выполняется локальным компонентом `pbs_mom` с правами администратора `root`. Файл со сценарием должен быть доступен для выполнения и может быть скриптом или скомпилированной исполняемой программой. Кроме того, локальный компонент `pbs_mom` блокируется до завершения проверки работоспособности и не имеет встроенного тайм-аута. Поэтому рекомендуется сократить время выполнения сценария проверки и убедиться, что он не будет блокироваться даже в случае сбоя или отказа.

Сценарий может выполнять любые необходимые системные вызовы и любую комбинацию системных утилит, но не должен выполнять команды СУР. В случае если сценарий обнаруживает сбой или отказ, в стандартный поток вывода `stdout` записывается ключевое слово `ERROR` с последующим сообщением (до 1024 символов) об ошибке. Локальный компонент `pbs_mom` сохраняет сообщение об ошибке в атрибут `message` параметра `status` ЭМ.

Разработкой bash-сценария проверки работоспособности ЭМ с открытым исходным кодом Node Health Check (NHC) [12] занимается коллектив лаборатории Lawrence Berkeley National Laboratory. Сценарий предлагает простой способ выполнения некоторых из наиболее распространенных задач проверки работоспособности ЭМ, таких как проверка функциональности сети и файловой системы. Более подробная информация доступна на странице проекта [12].

Количество инструкций в скрипте фиксировано и определяется администратором, поэтому сложность проверки работоспособности ЭМ определяется алгоритмом постоянного времени – $T_{NHC} = O(1)$.

Центральный компонент `pbs_server` опрашивает состояние ЭМ с интервалом `node_ping_rate` (в секундах). Данный параметр настройки сервера определяет максимальный интервал между последовательными запросами, которые `pbs_server` отправляет `pbs_mom` для определения состояния ЭМ. Локальный компонент `pbs_mom` отправляет серверу статус и состояние ЭМ, полученные в результате последнего выполнения скрипта проверки работоспособности. Параметр настройки центрального компонента `node_check_rate` позволяет указать минимальную продолжительность времени (в секундах), в течение которого `pbs_server` может не получить обновление состояния ЭМ и при этом не изменять её состояние на неисправное (`down`). Таким образом проверяется исправность локального компонента `pbs_mom`.

Диагностика состояния ВС требует обращения ко всем ЭМ, поэтому сложность может быть определена как $T_{CheckNodes} = O(n + T_{Request})$, где n – число ЭМ в ВС, $T_{Request}$ – время передачи запросов и ответов о состоянии ЭМ с применением древовидного алгоритма $O(\log n)$.

На центральном и локальном компонентах ЭМ возможно использовать параметр настройки `down_on_error`, который может иметь значение `TRUE` или `FALSE`. Если параметр установлен в значение `TRUE` на локальном компоненте, то в случае сбоя или ошибки по результатам выполнения сценария проверки работоспособности `pbs_mom` самостоятельно изменяет состояние ЭМ на неисправное (`down`). Если параметр установлен в значение `true` на

центральном компоненте, то решение об исправности ЭМ и изменении состояния осуществляет pbs_server. По умолчанию параметр down_on_error задан на pbs_server и имеет значение TRUE.

На основе анализа исходных кодов планировщика Maui составлен псевдокод обработки сообщений об ошибках ЭМ.

```
function MPBSClusterQuery()
for each Node in NodesList
    for each Node_Attributte in NodeAttributeLine
        if (Node_Attribute == message && Attribute_Value == "ERROR:")
            SetNodeState(down)
```

В настоящее время PBS/TORQUE игнорирует сообщения об ошибках, сохраненные в атрибуте message. Эти сообщения поступают в планировщик, но ни pbs_sched, ни maui не имеют алгоритмов обработки тех или иных ошибок. У коммерческих планировщиков, таких как Moab [13], имеется возможность передавать информацию об ошибках администраторам ВС, но обработка сообщений об ошибках требует разработки соответствующих алгоритмов и программных модулей.

3.3. Отказоустойчивость инфраструктуры ВС

Важную роль для отказоустойчивости ВС играет доступность сервера сетевой файловой системы NFS, через которую обеспечивается доступ к рабочим директориям пользователей и системным файлам PBS/TORQUE в случае выполнения нескольких экземпляров pbs_server. Существуют рекомендации по настройке высоконагруженного сервера сетевой файловой системы [14].

Корректная работа центральных и локальных компонентов PBS/TORQUE зависит от возможности выполнять прямой и обратный поиск доменных имен для всех ЭМ ВС. Таким образом, недоступность файла /etc/hosts ЭМ или службы, отвечающей за разрешение имен (DNS или NIS), может приводить к сбоям или отказам.

При наличии брандмауэров (iptables или firewalld) на ЭМ ВС необходимо на всех центральных и локальных компонентах PBS/TORQUE разрешить подключения на все порты, необходимые для функционирования СУР. Центральный компонент pbs_server и локальный компонент pbs_mom по умолчанию используют TCP и UDP порты 15001–15004. Проблемы, связанные с брандмауэром, часто связаны со сбоями связи между pbs_server и pbs_mom и сообщениями, такими как «преждевременный конец сообщения» в файлах журнала.

3.4. Отказоустойчивость параллельных программ

Отказоустойчивость (Fault Tolerance) применительно к задаче как заданию ВС на выполнение параллельной программы может иметь несколько значений. В первом случае отказоустойчивость подразумевает лояльность к ошибкам, например, когда игнорирование одной неудачной попытки опроса состояния может быть исправлено при следующей попытке. Во втором случае отказоустойчивость подразумевает способность ВС возобновлять выполнение параллельной программы после отказа ЭМ с сохранением промежуточных результатов.

Опция «-f» команды qsub позволяет сделать задачу, устанавливаемую в очередь СУР, лояльной к ошибкам. Такой способ рекомендуется для задач, требующих большого числа ЭМ для решения. Если задача является лояльной к ошибкам, то она не будет отменена на основе неудачного опроса состояния работоспособности ЭМ решающей подсистемы (независимо от того, сколько ЭМ не сообщает об ошибке).

Администраторы ВС могут сделать все задачи лояльными к ошибкам по умолчанию. Для этого необходимо в файле torque.cfg (по умолчанию находится в директории

`/var/spool/torque`) параметр `FAULT_TOLERANT_BY_DEFAULT` установить в значение `TRUE`. В этом случае изменяется поведение команды `qsub` на тех хост-машинах, на которых находится файл с соответствующей настройкой.

Отказоустойчивое выполнение параллельных программ [15] возможно при условии периодического сохранения состояния вычислительных процессов в контрольной точке (КТ). При создании КТ выполняется сохранение состояния процессов параллельной программы в файл образа контрольной точки.

В настоящее время в PBS/TORQUE не реализована функция формирования КТ, однако допускается возможность сопряжения со специализированным программным обеспечением. Наиболее популярными программными продуктами в области создания и управления КТ являются пакеты Berkeley Lab Checkpoint/Restart (BLCR) [16] и Distributed MultiThreaded CheckPointing (DMTCP) [17].

Средства формирования КТ позволяют сохранить и возобновить вычислительный процесс только на подсистеме из исправных ЭМ. В функции BLCR и DMTCP не входит обеспечение отказоустойчивости – обнаружение неисправных ЭМ, исключение или замена отказавшей ЭМ из рабочей подсистемы, адаптация вычислительных алгоритмов к отказам исполняющих ЭМ.

Средства сопряжения с MPI-библиотеками (Message Passing Interface) [18] позволяют создавать КТ параллельных программ на распределенных ВС без изменения исходного кода. Большинство реализаций MPI-библиотек имеют встроенную поддержку BLCR, например, Open MPI, MPICH и др. Список ЭМ, на которых необходимо выполнить параллельную программу, передается в среду времени выполнения (runtime environment) подсистемы MPI через переменную среды окружения `$PBS_NODEFILE`. В этой переменной содержится полный путь до файла, содержащего построчно имена хостов ЭМ, назначенных для выполнения параллельной программы. Изменение содержимого файла в течение выполнения параллельной программы не предусматривается.

4. Функциональная модель обработки отказов

Предлагается модель обеспечения отказоустойчивого функционирования ВС под управлением СУР PBS/TORQUE и планировщика Maui за счёт структурной избыточности.

Расчет числа ЭМ структурной избыточности может быть выполнен на основе математических моделей функционирования ВС [19, 20]. Особенность предлагаемого подхода заключается в сокращении накладных расходов на поиск ЭМ для восстановления вычислений из КТ.

ЭМ структурной избыточности могут входить как в общий резерв всей ВС (рис. 3а), так и в локальный резерв подсистемы исполнения параллельной программы по запросу пользователя (рис. 3б).

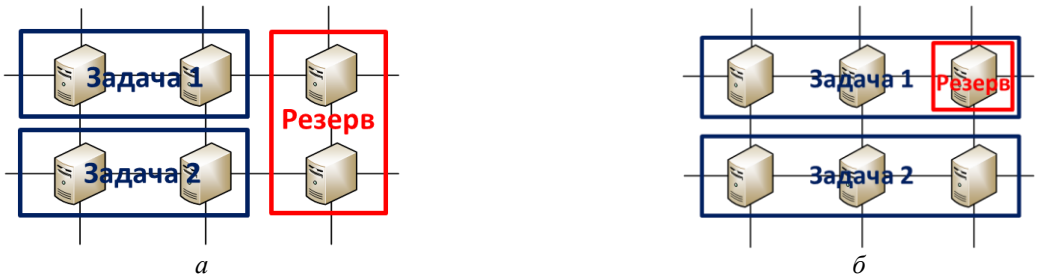

Рис. 3. Модель функционирования ВС:
а – с общим резервом; *б* – с локальным резервом по требованию

Общий резерв может быть создан по желанию администратора ВС на основе интенсивностей отказов и восстановлений ЭМ ВС. В листинге ниже представлен вариант настройки интерфейса центрального компонента СУР PBS/TORQUE.

```
$ qmgr -c 'p s'
#
# Set server attributes.
#
   set server scheduling = True
   set server managers = admin@admin
   set server default_queue = batch
   set server redundancy = True
   ...
```

Запрос на создание локального резерва может быть создан пользователем в паспорте задачи. Например, как показано в листинге ниже.

```
$ cat test.job
#PBS –N Job_Name
#PBS –q Batch_Name
#PBS –l nodes=4:ppn=8
#PBS –R redundancy_nodes

mpiexec ./mpi_prog ...
```

Ниже представлен псевдокод функции по обработке отказов и восстановлению подсистемы выполнения параллельной программы за счет ЭМ из структурной избыточности.

```
/* function MPBSWorkloadQuery() */
for each Job in JobList
    for each Node in JobNodesList
        if (Node->state == Drained) SetJobState(job_id, Suspended)

/* function QueueScheduleSuspendedJob() */
if (RestoreJobNodeListFromReserve(job_id) == succes)
    ResumeJobFromCheckPoint(job_id)
```

5. Заключение

В работе рассмотрена проблема отказоустойчивого функционирования распределенных вычислительных систем под управлением PBS/TORQUE при выполнении параллельных MPI-программ. Исследовано влияние событий сбоев и отказов на возможность завершить выполнение параллельной программы.

В результате теоретического анализа средств обеспечения отказоустойчивости системы управления ресурсами PBS/TORQUE с модульным планировщиком Maui выявлены нижеследующие проблемы:

1) в планировщике Maui отсутствуют алгоритмы и программные модули для обработки сообщений об отказах на ЭМ, которые задействованы в выполнении параллельной MPI-программы;

2) в СУР PBS/TORQUE отсутствуют алгоритмические и программные средства обеспечения отказоустойчивого функционирования за счет резервирования избыточных ЭМ в соответствии с классическими подходами теории надежности;

3) возобновление выполнения параллельной программы в случае отказов исполняющих её ЭМ требует повторной постановки в очередь, что может приводить к существенным накладным расходам.

Во второй части работы предложена функциональная модель обработки отказов для планировщика Maui.

Литература

1. *Хорошевский В. Г.* Архитектура вычислительных систем. М.: МГТУ им. Н. Э. Баумана, 2008. 520 с.
2. *Schroeder B., Gibson Garth A.* A large-scale study of failures in high-performance computing systems // Proceedings of the International Conference on Dependable Systems and Networks (DSN2006), 2006. 10 p.
3. *Gupta S., Patel T., Engelmann C., Tiwari D.* Failures in large scale systems: long-term measurement, analysis, and implications // Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017. P. 1–12.
4. *Каляев И. А., Коробкин В. В., Мельник Э. В., Малахов И. В.* Отказоустойчивый управляющий вычислительный комплекс машины перегрузочной атомного реактора типа ВВЭР // Мехатроника, автоматизация, управление. 2003. № 3. С. 143–146.
5. *Korobkin V., Melnik E., Klimenko A.* Fault-tolerant architecture for the hazardous object information control systems // 2015 IEEE conference “Application of information and communication technologies” (IEEE catalog number CFPI556H-PRT). P. 274–276.
6. *Cappello F., Geist A., Gropp W., Kale S., et all.* Toward Exascale Resilience: 2014 update // Supercomputing frontiers and innovations. 2014. V. 1, № 1. P. 1–28.
7. Torque Resource Manager [Электронный ресурс]. URL: http://www.adaptivecomputing.com/products/torque/ (дата обращения: 12.10.2018).
8. Maui Cluster Scheduler [Электронный ресурс]. URL: http://www.adaptivecomputing.com/support/download-center/maui-cluster-scheduler/ (дата обращения: 12.10.2018).
9. Torque Resource Manager Administrator Guide 6.1.2 [Электронный ресурс]. URL: http://docs.adaptivecomputing.com/torque/6-1-2/adminGuide/torqueAdminGuide-6.1.2.pdf (дата обращения: 12.10.2018).
10. Maui Administrator's Guide [Электронный ресурс]. URL: http://docs.adaptivecomputing.com/maui/pdf/mauiadmin.pdf (дата обращения: 12.10.2018).
11. GDB: The GNU Project Debugger [Электронный ресурс]. URL: http://www.gnu.org/software/gdb/ (дата обращения: 12.10.2018).
12. LBNL Node Health Check [Электронный ресурс]. URL: https://github.com/mej/nhc (дата обращения: 12.10.2018).
13. Moab Cloud HPC Suite [Электронный ресурс]. URL: http://www.adaptivecomputing.com/moab-hpc-basic-edition/ (дата обращения: 12.10.2018).
14. An active/passive NFS server in a red hat high availability cluster [Электронный ресурс]. URL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_administration/ch-nfsserver-haaa (дата обращения: 12.10.2018).
15. *Elnozahy E. N., Alvisi L., Wang Y. M., Johnson D. B.* A survey of rollback-recovery protocols in message-passing systems // ACM Computing Surveys. 2002. V. 34, № 3. P. 375–408.
16. *Duell J., Hargrove P., Roman E.* The Design and Implementation of Berkeley Lab's Linux Checkpoint/Restart // Berkeley Lab Technical Report. 2002. 17 p.
17. *Ansel J., Arya K., Cooperman G.* DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop // IEEE International Parallel and Distributed Processing Symposium, 2009. 12 p.
18. Message Passing Interface [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Message_Passing_Interface (дата обращения: 12.10.2018).

19. *Pavsky V. A., Pavsky K. V., Paznikov A. A.* Mathematical models and calculation of reliability indices of scalable distributed computer systems under full restoration // Proceedings of XIV International scientific-technical conference “Actual Problems of Electronic Instrument Engineering” (APEIE-2018). NSTU. V. 1, Part 4. Novosibirsk. 2018. P. 502–505.
20. *Павский В. А., Павский К. В.* Расчет показателей потенциальной живучести для распределенных вычислительных систем при групповом восстановлении отказавших машин // Материалы 4-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии», 19–24 сентября 2016 г., Ростов-на-Дону. Т. 2. С 86-89.

Статья поступила в редакцию 06.11.2018;
переработанный вариант – 14.12.2018.

Ефимов Александр Владимирович
к.т.н., доцент кафедры вычислительных систем СибГУТИ (630102, Новосибирск, ул. Кирова, 86), ведущий инженер лаборатории вычислительных систем ИФП СО РАН (630090, Новосибирск, проспект Академика Лаврентьева, 13), тел. (383) 269-82-93, e-mail: alexandr.v.efimov@sibguti.ru.

Павский Кирилл Валерьевич
д.т.н., доцент, профессор кафедры вычислительных систем СибГУТИ, заведующий лабораторией вычислительных систем ИФП СО РАН, тел. (383) 333-21-71, e-mail: pkv@isp.nsc.ru.

Analysis of resource management system PBS/TORQUE robustness tools

A. Efimov, K. Pavsky

The paper is devoted to the problem of distributed computing systems robustness with PBS/TORQUE when running parallel MPI programs. The impact of faults and failures events on the ability to terminate a parallel program is considered. The theoretical analysis of potential possibilities of tools for providing fault tolerance of PBS/TORQUE resource management system is presented in the first part of the paper. The results of the analysis on the base of experimental study using the resources of the existing distributed computing system of the SibSUTIS center of parallel computing technologies and the ISP SB RAS laboratory of computing systems are presented in the second part.

Keywords: distributed computing systems, resource management system, PBS/TORQUE, robustness, analysis, fault and failure.