МОДЕРНИЗАЦИЯ СУПЗ МВС-1000

А.В. Баранов, С.В. Смирнов, М.Ю. Храмцов, С.В. Шарф

Практически на всех отечественных высокопроизводительных вычислительных установках типа МВС-1000 (примеры - МВС-1000М, МВС 15000, МВС-100К) в качестве системы пакетной обработки используется система управления прохождением задач (СУПЗ) [1]. СУПЗ обеспечивает многопользовательский доступ к вычислительной установке с ведением очереди параллельных задач.

СУПЗ имеет развитый командный интерфейс с расширенной поддержкой запуска программ, написанных с использованием библиотеки MPI. Планирование очереди осуществляется с помощью гибкого настраиваемого алгоритма, учитывающего специфику и размер параллельных задач. Все задачи делятся на три категории - отладочные (короткие по времени и с малым числом процессоров), пакетные (большие задачи, не прерываемые системой) и фоновые (сверхбольшие задачи, выполнение которых может периодически прерываться). Планирование задач ведется совместно, но с разными условиями для разных категорий, что позволяет добиться высокой эффективности работы СУПЗ.

СУПЗ МВС-1000 непрерывно совершенствуется. Последняя модернизация системы привнесла следующие возможности:

1. Использование шаблонов для управления распределением вычислительных процессов по процессорам.
2. Многоресурсное планирование.
3. PBS-подобный командный интерфейс СУПЗ, совместимый со стандартом POSIX.

Рассмотрим новые возможности СУПЗ подробнее.

Распределение задач по вычислительным модулям решающего поля СУПЗ производит автоматически. Пользователь не может влиять на процесс распределения и заранее знать, на какие модули будет спланирована его задача. По умолчанию число процессов задачи, запускаемых на вычислительном модуле, равно числу процессоров модуля. С помощью файла-шаблона пользователь может самостоятельно задавать, на каком модуле сколько процессов должно быть запущено.

Для того, чтобы воспользоваться новой возможностью, пользователь должен предположить, что его задача будет запущена на нужном ему числе вычислительных модулей с предопределенными виртуальными именами node1, node2, node3,:,nodeN, где N - число необходимых для задачи вычислительных модулей. Распределение процессов по модулям задается с помощью специального файла-шаблона. Формат данного файла аналогичен известному формату файла описания вычислительных узлов (machine file) для MPI-программы.

После того, как задача пройдет через очередь и поступит на выполнение, СУПЗ заменит виртуальные имена вычислительных модулей файла-шаблона реальными именами модулей, выделенных задаче для счета. Процессы задачи будут распределены по реальным вычислительным модулям точно так, как указал пользователь в файле-шаблоне.

До недавнего времени СУПЗ поддерживала планирование только двух вычислительных ресурсов - процессоров и локальной дисковой памяти. Современная тенденция построения высокопроизводительных систем такова, что вычислительная установка постоянно подвергается модернизации в процессе эксплуатации. При этом редко проводится полная модернизация всей системы, часто ограничиваются наращиванием мощности лишь части системы. Такой частичный "апгрейд" приводит к неоднородности решающего поля. Разные вычислительные модули могут содержать процессоры разной мощности, разные объем оперативной памяти и коммуникационное оборудование. Если факта разнородности не учитывать при планировании, то выделение ресурсов будет производиться неэффективно. Более оснащенные вычислительные модули могут простаивать или выделяться под задачи, которым не нужны расширенные ресурсы.

Для поддержки планирования множества вычислительных ресурсов СУПЗ была соответствующим образом модернизирована. Теперь при постановке задачи в очередь (запуске) пользователь может указать, какие дополнительные вычислительные ресурсы (помимо процессоров) требуются для его задачи.

Дополнительные вычислительные ресурсы специфицируются администратором системы. При этом каждый специфицированный дополнительный ресурс имеет имя и может быть числовым или строковым. Числовые вычислительные ресурсы выделятся по принципу "не меньше, чем задано пользователем", а строковые - по принципу "точного соответствия значению, заданному пользователем". В настоящий момент многоресурсное планирование доступно на вычислительной установке МВС-100К в МСЦ РАН.

В 1994г. был принят стандарт POSIX 1003.2d Batch Environment Standard на системы пакетной обработки для многопроцессорных вычислительных систем. Цель подхода POSIX [2] состоит в том, чтобы обеспечить воз-

можность решения проблемы переносимости прикладных программ между различными компьютерными платформами.

Существующий командный интерфейс СУПЗ МВС-1000 разрабатывался, начиная с 90-х годов 20-го века, и сложился исторически. Существующий командный интерфейс СУПЗ МВС-1000 не соответствует стандарту POSIX, что ограничивает его функциональность в свойствах открытости, на которые нацелен POSIX. Следование рекомендациям POSIX позволит решить следующие задачи:

1. Интеграция информационных систем из компонент различных изготовителей. Например, существуют грид-системы, поддерживающие POSIX-интерфейс, наличие такого интерфейса в СУПЗ МВС-1000 позволит ее интегрировать в грид с минимальными затратами.

2. Обеспечение эффективности реализаций и разработок, благодаря точности спецификаций и соответствию стандартным решениям, отражающим передовой научно-технический уровень.

3. Обеспечение эффективности переноса прикладного программного обеспечения, благодаря использованию стандартизованных интерфейсов и прозрачности механизмов реализации сервисов систем. Известно, что пользователи при работе с POSIX-совместимой системой OpenPBS накопили большое количество командных файлов для запуска и сопровождения задач. При переходе пользователей на СУПЗ МВС-1000 желательно, чтобы они могли использовать ранее созданное ПО.

При создании POSIX-совместимого командного интерфейса СУПЗ было решено ориентироваться на известную систему пакетной обработки OpenPBS/Torque [3], которая соответствует спецификации POSIX 1003.2d Batch Environment Standard.

Был произведен анализ несоответствий команд пользовательского интерфейса СУПЗ МВС-1000 системе OpenPBS/Torque, все выявленные несоответствия разделены на три группы.

К первой группе относятся команды, которые лишь по своему синтаксису не схожи с командами пользователя системы OpenPBS. Ко второй группе относятся команды, которые дополняют и изменяют функциональность СУПЗ. Третью группу составляют команды, затрагивающие принципы построения СУПЗ МВС-1000. К ним относятся команды OpenPBS qmove, qalter, qselect (с ключом -q) и qsub (с ключами -q и -u). С одной стороны трудоемкость их реализации эквивалентна созданию с "нуля" всей СУПЗ, с другой - указанные команды и ключи не являются основными для большинства пользователей. Поэтому команды третьей группы не подлежали реализации в рамках производимой модернизации СУПЗ.

При реализации POSIX-совместимых команд для СУПЗ авторы преследовали цель обеспечить полный цикл работы пользователя в многопроцессорной вычислительной системе. Обычный порядок работы пользователя следующий:

1. Пользователь подключается к серверу доступа системы, редактирует программы и данные, транслирует программы в исполнительные файлы и запускает задачи. Для запуска используется команда qsub.

2. Запущенные задачи ставятся в очередь. Пользователь может просматривать очередь задач, смотреть статус задач. Это обеспечивает команда qstat.

3. Пользователь производит манипуляции своими задачами: удаление задачи из очереди, снятие задачи со счета, а также перезапуск задачи. Этим действиям соответствуют команды qdel, qrerun. Имеется возможность выбрать задачи, удовлетворяющие заданным условиям, с помощью команды qselect.

В результате был реализован следующий набор POSIX-совместимых команд: команда qsub с ключами [c], [-C], [-e], [-l], [-N], [-o], [-p], [-q], [z]; команда qstat с ключами [B], [B f], [q], [-Q], [-Q -f]; команда qdel, команда qrerun, команда qselect с ключами [-l], [-N], [-u].

Перечисленный набор команд обеспечивает проведение полного цикла работы пользователя с системой и, по нашим оценкам, покрывает до 80% потребностей пользователя.

ЛИТЕРАТУРА:

1. Руководство пользователя СУПЗ МВС-1000. М.:ИПМ им. М.В. Келдыша РАН. 2002.
2. Коваленко В., Коваленко Е. Пакетная обработка заданий в компьютерных сетях: Открытые системы // http://www.osp.ru/os/2000/07-08/178074/_p1.html
3. Portable Batch System - User Guide // http://car.labf.usb.ve/pbs/homepage.html