

РАСШИРЕНИЕ ФУНКЦИОНАЛЬНОСТИ МЕНЕДЖЕРА РЕСУРСОВ СУПЕРКОМПЬЮТЕРА SLURM

С.Н. Леоненков

Факультет Вычислительной Математики и Кибернетики МГУ им. М. В. Ломоносова

Каждый год вместе с производительностью суперкомпьютерных комплексов по всему миру не менее быстро растет и база пользователей подобных систем. Ресурсы этих систем становятся все более востребованными для решения научных, промышленных, финансовых и прочих задач разной сложности.

Пользователи высокопроизводительных кластеров имеют различные права доступа, запрашивают от одного до многих тысяч процессоров для запуска своих задач, время выполнения которых колеблется от нескольких секунд многих часов или даже дней.

Решение задачи удовлетворения всех запросов пользователей в режиме реального времени в рамках ограниченных мощностей является краеугольным камнем для системных администраторов загруженных вычислительных систем.

Суперкомпьютер “Ломоносов”, используемый для исследований, которые проводят ученые и студенты Московского Государственного Университета имени М. В. Ломоносова, возглавляет рейтинг Top50 суперкомпьютеров, установленных на территории СНГ и показавших наибольшую производительность на тесте Linpack, и занимает 42 место в авторитетном международном рейтинге самых высокопроизводительных вычислительных систем Top500.

Система состоит из 12422 процессоров, но даже такого количества не достаточно, чтобы обеспечить все группы исследователей, работающих на суперкомпьютере “Ломоносов”. Все процессоры физически разделены на серверы по 4 или 6 штук, называемые узлами, которые логически объединены на разделы. Каждый раздел ориентирован на определенные потребности пользователей.

В текущей конфигурации “Ломоносова” доступны 8 разделов, количество узлов в каждой колеблется от 1 до 4096. В среднем в день выполняется от 200 до 400 задач, которые ставят на запуск 30-50 разных пользователей, причем каждый год активно используется около 400 аккаунтов, и эти числа не являются пределом. Чтобы справляться с ежедневно растущими нагрузками на суперкомпьютер “Ломоносов”, на нем используется система управления задачами SLURM (*Simple Linux Utility for Resource Management*).

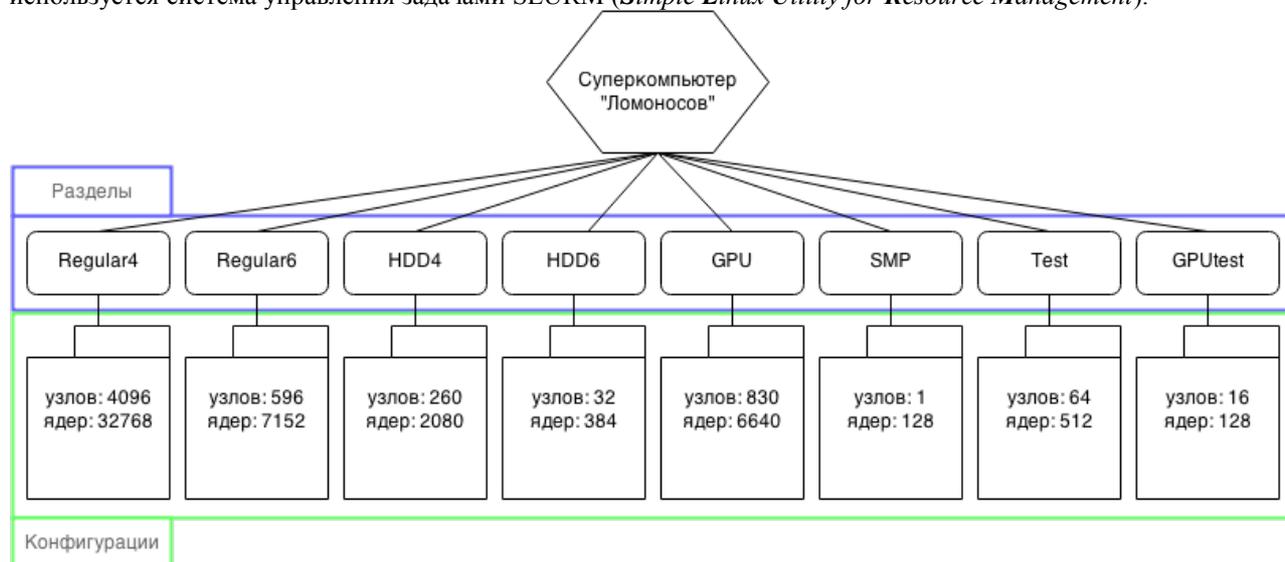


Рис. 1. Конфигурация разделов суперкомпьютера “Ломоносова”

SLURM

SLURM – это высокомасштабируемый, отказоустойчивый менеджер кластеров и планировщик заданий для больших систем вычислительных узлов. Исходный код SLURM находится в свободном доступе в соответствии с GNU General Public License. В основе этой системы реализована иерархическая архитектура управления суперкомпьютером.

Главным агентом менеджера является основной контроллер, который содержит управляющий демон slurmctld. В некоторых архитектурах для повышения отказоустойчивости менеджера основной контроллер может дополнять резервный. Основной задачей slurmctld является выделение ресурсов под задачи, который

были поставлены пользователями. Для управления каждым отдельным узлом на них развернуты демоны slurmд. Они служат для запуска и управления заданием на самом узле (получение задания от основного контроллера, запуск уже непосредственно на ядрах) и мониторинга его состояния.

Одним из основных преимуществ SLURM является модульность, доступны десятки дополнительных плагинов. SLURM включает в себя стандартные планировщики, один из которых (sched/backfill) используется для оптимизации работы суперкомпьютера “Ломоносова”. При возникновении ситуации, что стандартные алгоритмы планирования полностью не устраивают системного администратора, можно легко ввести в эксплуатацию собственный плагин, используя стандартные интерфейсы (wiki и wiki2), которые предоставляет SLURM.

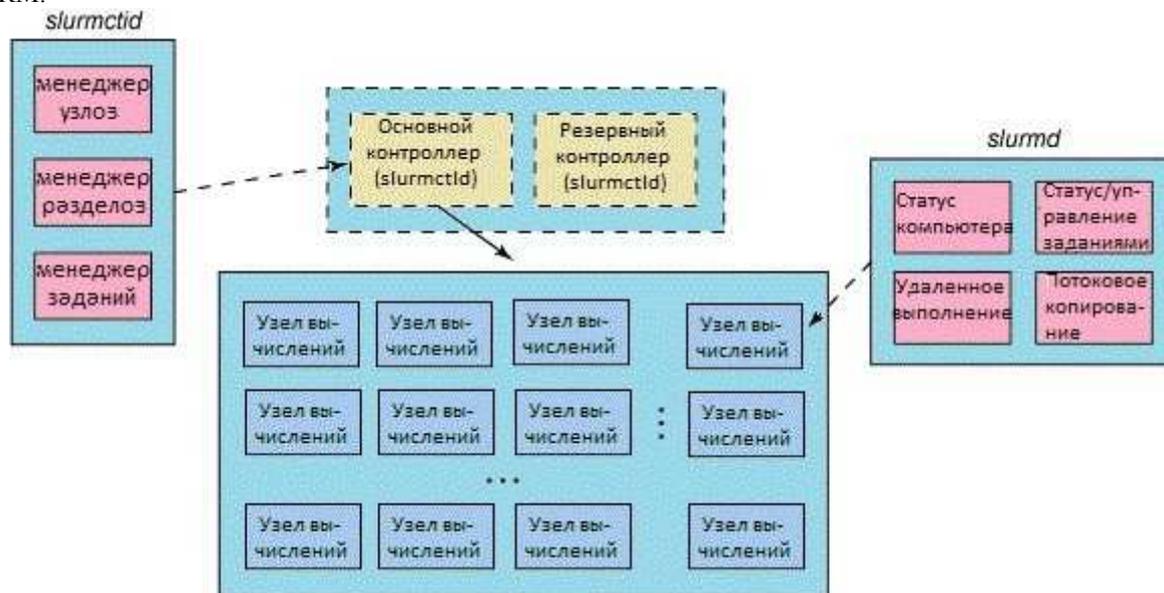


Рис. 2. Схема структурной организации SLURM

Системный администратор может достаточно гибко настраивать логическую конфигурацию вычислительной системы, которую будет поддерживать SLURM, и легко варьировать множество параметров кластера путем изменения конфигурационного файла или использования соответствующих команд.

Специфика использования менеджера на суперкомпьютере “Ломоносов” предполагает расширение функциональности программного комплекса SLURM для повышения эффективности системы.

Ключевые проблемы, с которыми приходится сталкиваться при использовании SLURM для управления суперкомпьютером “Ломоносов”: невозможность ограничить пользователя так, чтобы он не мог занять все ресурсы одной очереди на длительное время, слабая предсказуемость приоритетов задач и трудность учёта данных внешними средствами. Для оптимизации работы планировщика SLURM было решено внести следующие дополнения в функционал:

1. Учет и контроль процессорочасов, запрошенных каждым пользователем;
2. Создание прозрачной системы приоритетов с возможностью настройки в режиме реального времени;
3. Реализация возможности использования узлов из разных разделов для пользователей с определенным приоритетом;
4. Добавление квот по времени: определённое число процессорочасов в неделю/месяц/год;
5. Добавление внешних обработчиков событий очереди и учёта заданий.

РЕАЛИЗАЦИЯ

Согласно требованиям к ПО, предъявляемым в Суперкомпьютерном центре МГУ, на техническое решение были наложены некоторые ограничения. Периодически команда разработчиков SLURM выпускает новую версию менеджера, и поэтому наши нововведения должны быть переносимым.

Все новые функции, которые были внесены в программный код SLURM, должны за разумное время быть переносимы в новую версию менеджера. Это требование является основным, из него вытекает множество побочных ограничений, с которыми пришлось столкнуться. Например, изменения не могут вноситься в ядро системы: фоновые программы slurmд и slurmctld, потому что это крайне затормозит процесс обновления, намного практичнее изменять внешние модули системы. Ориентированность ядра SLURM на внешние плагины не позволяет создавать свои плагины, не ориентируясь на интерфейсы wiki и wiki2.

Другим затрудняющим разработку фактором является отсутствие возможности добавления новых “состояний” задач и их описаний не затрагивая ядро SLURM. Основной цикл планировщика описан в файле job_scheduler.c, который является часть утилиты slurmctld.

Если все же добавить новые “состояния” и их описания во внешних файлах, то они будут неправильно восприняты или проигнорированы slurmctld. Последней серьезной проблемой является отсутствие возможности использования встроенных методов хранения настроек новых дополнений. Встает необходимость написания своего хранилища данных, которые будут использованы для корректной работы новых дополнений.

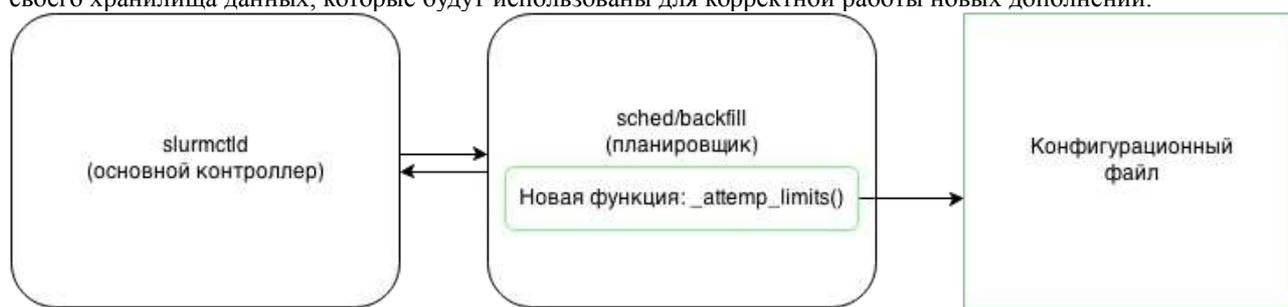


Рис. 3. Архитектура решения на примере контроля лимита процессорочасов.

Рассмотрим процесс реализации дополнения на примере счетчика учета и контроля процессорочасов, запрошенных каждым пользователем. Удобным инструментом управления распределением ресурсов вычислительного комплекса является счетчик выделенных процессорочасов каждому пользователю.

По умолчанию такого счётчика нет в SLURM, важно реализовать возможность ограничения использования ресурсов разделов каждым пользователем. Это реализуется с помощью лимита одновременно запрошенных процессорочасов: все одновременно работающие задачи пользователя в сумме не могут использовать более заданного числа процессорочасов.

Это значит, например, что пользователь сможет занять много узлов раздела, но не надолго, либо, наоборот, использовать небольшое число узлов постоянно. Таким образом, один пользователь не сможет ущемлять интересы других.

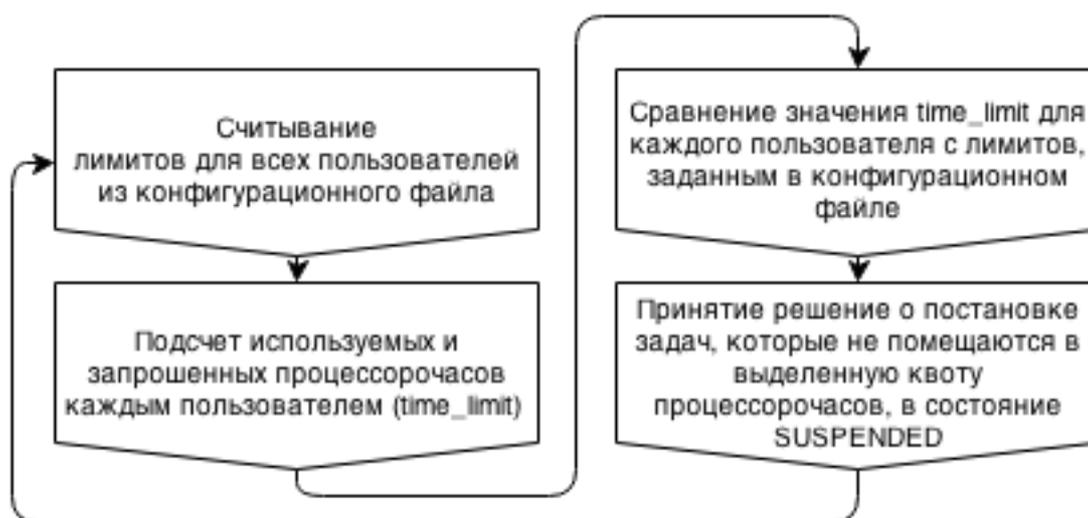


Рис. 4. Flowchart дополнения учета лимитов процессорочасов.

Оценив все вышеперечисленные проблемы, было решено изменять стандартный планировщик sched/backfill. В основной цикл планировщика был добавлен процесс пересчета процессорочасов для каждого отдельного пользователя.

Путем подсчета суммарного времени и количества процессоров, запрошенных для выполнения задач во всех очередях всех разделов, каждому пользователю ставится в соответствие параметр запрошенных и используемых в данный момент процессорочасов time_limit. В рамках работы sched/backfill считываются лимиты установленные для каждого пользователя.

Все задачи, которые “не поместились” в заданный лимит, ставятся в состояние SUSPENDED. Как только количество “свободных” процессорочасов становится достаточным для запуска задач пользователей, которые уже были поставлены в состояние SUSPENDED, то SLURM вновь возвращает их в очередь на выполнение.

Для разбора конфигурационного файла использовалась специальная библиотека libconfig. Ниже приведена архитектура конфигурационного файла.

```

limits =
{
  default = ({ limit = 101; }
  
```

```

    );
    users = ( { user = "root";
              limit = 60000; },
            ...
<ПРОЧИЕ ПОЛЬЗОВАТЕЛИ>
            ...
              { user = "stud";
                limit = 60000; }
            );

    groups = ( { group = "students";
               limit = 12; },
             ...
<ПРОЧИЕ ГРУППЫ ПОЛЬЗОВАТЕЛЕЙ>
             ...
               { group = "workers";
                 limit = 1000; }
             );
};

```

Значение default - значение порога процессорочасов по умолчанию. Значение users - значение порога процессорочасов для каждого пользователя. Значение groups - значение порога процессорочасов для каждой группы пользователей. Самый высокий приоритет имеет значение users, затем groups и default. Тип конфигурационного файла - "CFG" (.cfg).

ТЕСТИРОВАНИЕ

Тестирование было проведено на виртуальном вычислительном кластере, который состоял из 1200 узлов. Была смоделирована как ситуация без использования нового функционала, так и с использованием.

Первый тест — без использования нового функционала. Один пользователь (root) ставит большое количество задач разных размеров на исполнение (682, 683, 685, 694), причем его запросы превышают лимит процессорочасов, заданный на кластере, другие пользователи (stud) через некоторое время попытаются ставить задачи разных размеров на этом же кластере (700, 701, 702, 703), но из-за больших запросов первого пользователя они не имеют возможности воспользоваться даже частью вычислительных узлов. На рисунке 5 можно наблюдать описанную ситуацию.

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
662	def	test.sh	stud	PD	0:00	100	(Resources)
663	def	test.sh	stud	PD	0:00	100	(Priority)
622	def	test.sh	root	R	5:08	500	dummy [1-500]
623	def	test.sh	root	R	5:08	500	dummy [501-1000]
626	def	test.sh	root	R	3:41	100	dummy [1001-1100]
661	def	test.sh	stud	R	0:00	100	dummy [1101-1200]
627	def	test.sh	root	S	0:00	0	
628	def	test.sh	root	S	0:00	0	
629	def	test.sh	root	S	0:00	0	
630	def	test.sh	root	S	0:00	0	
631	def	test.sh	root	S	0:00	0	

Рис. 5. Типичная ситуация: пользователь (root) монополично использует ресурсы.

Включив режим учета процессора часов и повторив все действия первого теста, были получены улучшенные результаты - ресурсы стали доступны для большей группы пользователей (результаты теста на рис. 6). Первый пользователь (root) превысил свой лимит процессорочасов, поэтому не все его задачи были поставлены на выполнение (а только 622, 623 и 626). Это дало возможность другому пользователю (stud) получить доступ к кластеру (на выполнение задачи 661, причем раньше, чем задачи stud 627-631).

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
700	def	test.sh	stud	PD	0:00	100	(Resources)
701	def	test.sh	stud	PD	0:00	100	(Resources)
702	def	test.sh	stud	PD	0:00	100	(Resources)
703	def	test.sh	stud	PD	0:00	100	(Resources)
682	def	test.sh	root	R	11:38	500	dummy [1-500]
683	def	test.sh	root	R	11:38	500	dummy [501-1000]
685	def	test.sh	root	R	10:47	100	dummy [1001-1100]
694	def	test.sh	root	R	0:40	100	dummy [1101-1200]

Гибкая настройка параметров лимитов процессорочасов как для каждого пользователя в отдельности (например, исходя из его предыдущих успехов в работе с кластером), так и сразу для групп пользователей дает неоспоримые преимущества перед вариантом работы менеджера без учета этого параметра.

ЗАКЛЮЧЕНИЕ

Работа выполняется при поддержке гранта РФФИ 13-07-00750-А. В дальнейшие планы входит реализация остальных описанных дополнений к функционалу программного комплекса SLURM для использования в Суперкомпьютерном центре МГУ.

ЛИТЕРАТУРА:

1. M. Jette and M. Grondona, SLURM: Simple Linux Utility for Resource Management. Proceedings of ClusterWorld Conference and Expo, Сан-Хосе, Калифорния, Июнь 2003.
2. Slurm Workload Manager // [Электронный ресурс]. URL: <http://slurm.schedmd.com/slurm.html>
3. Lomonosov — T-Platforms // Официальный сайт рейтинга TOP500. [Электронный ресурс]. URL: <http://www.top500.org/system/177421>
4. М.Т. Джонс, «Оптимизация управления ресурсами суперкомпьютеров с помощью SLURM» // [Электронный ресурс]. URL: <http://www.ibm.com/developerworks/ru/library/l-slurm-utility/>
5. Суперкомпьютер «Ломоносов». Общая характеристика. // [Электронный ресурс]. URL: <http://parallel.ru/cluster/lomonosov.html>
6. D. Lipari, The SLURM Scheduler Design // http://slurm.schedmd.com/slurm_ug_2012/SUG-2012-Scheduling.pdf