ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ СО АН СССР

А.Д. Букин, Л.А. Миленький, В.А. Сидоров

ОРГАНИЗАЦИЯ ПАКЕТНОЙ ОБРАБОТКИ ЗАДАНИЙ

ПРЕПРИНТ 88-157

НОВОСИБИРСК

Организация пакетной обработки заданий

А.Д. Букин, Л.А. Миленький, В.А. Сидоров

Институт ядерной физики
630090, Новосибирск 90, СССР

АННОТАЦИЯ

Описаны принципы планирования очереди пакетных заданий. Подсистема ДИСПЕТЧЕР ЕС ЭВМ, созданная по этим принципам в ИЯФ СО АН СССР, позволила эффективно решить проблему разделения процессорного времени между пользователями.

1. ВВЕДЕНИЕ

Эксперименты по физике высоких энергий в качестве одной из своих составляющих включают обработку данных на вычислительных машинах. Задачи реконструкции событий по сигналам детектирующих устройств, математическое моделирование событий разных процессов взаимодействия частиц, определение параметров теоретических моделей требуют большого количества процессорного времени ЭВМ, что, как правило, приводит к дефициту вычислительной мощности. В крупных физических центрах обработку ведут большие коллективы научных сотрудников, и ликвидация возникающего социального напряжения по поводу разделения времени ЭВМ административным путем, т. е. путем самоограничения пользователей по договоренности друг с другом, сопровождается большими трудностями. Процессорного времени всегда не хватает.

Опыт использования ЭВМ в Институте ядерной физики СО АН СССР за прошедшие 20 лет, когда для обработки экспериментальных данных последовательно использовались ЭВМ Минск-22, Минск-32, ЕС-1040, ЕС-1060, ЕС-1061, показывает, что, несмотря на увеличение вычислительной мощности, дефицит не уменьшается. По-видимому, этот эффект, в основном, связан с использованием ЭВМ все большим числом независимых групп пользователей, а также с повышением роли ЭВМ в физических экспериментах. Из-за дефицита вычислительной мощности в ИЯФ с самого начала на больших ЭВМ был выбран пакетный режим исполнения заданий.

В конце 70-х, начале 80-х годов в нашем Институте много сил уделялось вопросу упорядочения очереди заданий в рамках операционной системы ОС на машинах серии ЕС. Варьировались определения классов заданий, количество инициаторов (параллельных потоков заданий), способы изменения приоритетов заданий в зависимости от времени ожидания в очереди и т. д. Дефицит времени только нарастал. Очереди заданий были близки к заданному пределу (было введено ограничение на количество заданий в очереди для каждого пользователя), многие задания находились в очереди по несколько недель. Общая схема работы счетных ЭВМ коллективного пользования на ВЦ ИЯФ СО АН СССР в то время описана в [1].

В зарубежных физических центрах дефицит машинного времени не такой острый, однако это не является следствием каких-либо достижений в вопросе организации очереди заданий. Скорее сказывается значительное опережение в мощности вычислительной техники. В основном, используются фирменные средства поддержки очереди заданий, ориентированные, как правило, на повышение производительности ЭВМ, однако встречаются случаи своих разработок [2].

В данной работе описывается новый способ организации очереди пакетных заданий, разработанный в ИЯФ СО АН СССР и ориентированный, в основном, не на эффективную загрузку счетных ЭВМ, а на эффективное разделение ресурсов этих машин как между отдельными пользователями, так и между группами пользователей.

2. ОБЩАЯ СХЕМА ПРОХОЖДЕНИЯ ЗАДАНИЙ

В большинстве крупных вычислительных центров для коллективного использования доступны несколько машин. Если есть необходимость организовать общую очередь заданий, то на практике используются два способа. При первом одна из счетных машин берет на себя дополнительную нагрузку администратора и передает другим машинам часть заданий из очереди. Этот способ имеет существенный недостаток: если выходит из строя машина-администратор, то очередь блокируется. При втором способе функции администратора возлагаются на дополнительную машину, которая используется только для организации очереди и диалога с пользователями. Естественно, когда ломается эта машина, то очередь тоже блокируется. Однако, для этой машины нет высоких требований на быстродействие процессора, поэтому ее можно подбирать именно по параметрам надежности, что существенно сглаживает эту проблему.

Эксплуатация трех машин ЕС-1061 на ВЦ ИЯФ организована по второй схеме. ЕС ЭВМ получают задания через ЭВМ М-6000, которая дальше будет называться подсистемой ДИСПЕТЧЕР ЕС ЭВМ, или просто «диспетчер». Тексты заданий, подготовленные пользователями, поступают в «диспетчер» и записываются на диск типа «винчестер».

Каждому заданию в «диспетчере» ставится в соответствие целое число — приоритет, способ вычисления которого будет описан ниже, так что задания образуют упорядоченную по этому приоритету очередь. Коротко опишем порядок выборки на исполнение заданий из очереди.

В настоящее время две машины ЕС-1061 работают под управлением операционной системы OS версии 21.8F + HASP, одна машина используется для освоения операционной системы СВМ и ведения пакетной обработки заданий тоже через «диспетчер». Эта машина не полностью загружена сейчас из-за отсутствия специального математического обеспечения экспериментов. На ней планируется комбинированное прохождение заданий: короткие задания — в интерактивном режиме (что очень удобно для отладки), большие задания — в пакетном. В ближайшем будущем все три машины будут использоваться с операционной системой СВМ, однако, принцип планирования пакетной очереди изменится слабо и дальше изложение будет идти на примере ЕС под управлением OS.

Каждая из двух машин имеет восемь инициаторов (термин OS), позволяющих одновременно выполнять до восьми заданий.

Когда одна из машин ЕС-1061 готова принять очередное задание (освободился инициатор), «диспетчер» выбирает из очереди задание с наивысшим приоритетом, удовлетворяющее следующим условиям:

— нет исполняющегося задания того же пользователя;
— задание не блокировано автором;
— есть достаточный объем ОЗУ;
— есть достаточное количество свободных НМЛ;
— гарантированное количество инициаторов занято короткими/большими заданиями или свободно;
— гарантированное количество инициаторов занято заданиями с положительным приоритетом или свободно.

Предпоследний пункт введен для того, чтобы ускорить прохождение коротких заданий и обеспечить лучшую загрузку процессора ЕС ЭВМ.

Последний пункт должен обеспечить наличие свободных инициаторов к началу рабочего дня (за ночь задания с положительным приоритетом обычно выполняются).

По окончании любого задания приоритеты заданий, хранящихся в очереди, вычисляются заново. Файл с результатами задания, по желанию пользователя, направляется или на АЦПУ в машинном зале, или на магнитный диск «диспетчера», откуда он может быть прочитан по запросу пользователя.

Дополнительные условия при выборке очередного задания на исполнение слабо влияют на прохождение очереди и порядок исполнения заданий, в основном, определяется приоритетом.

3. ПРИОРИТЕТ ЗАДАНИЯ

При разработке описываемой дисциплины планирования мы исходили из того, что конечным потребителем ресурсов счетных ЭВМ выступает пользователь и что единственным значимым для планирования ресурсом является процессорное время (время CPU). Одним из основных принципов, принятых нами, был принцип непрерывности планирования: приоритет заданий не должен зависеть ни от выделенных календарных дат, ни от астрономического времени суток. Исходя из этого, мы пришли к идее текущих условных «суток», «время» в которых измеряется в процессорном времени исполненных заданий. Такой эффект достигается путем пересчета потребленного пользователями процессорного времени после окончания любого задания и введения в формулы пересчета нормировки на длительность условных «суток».

Мы решили учитывать потребление процессорного времени как за «сутки», так и за более длительный срок — «неделю», так что для каждого пользователя «диспетчер» отслеживает, соответственно, две величины a_i и b_i, где i — номер пользователя.

Введенные величины a_i и b_i пересчитываются для всех пользователей по окончании любого задания по следующим формулам:

$$a_i = a_i \cdot [1 - t_j/T] + s_j \cdot t_j \cdot \delta_{ij}, \qquad (1)$$

$$b_i = b_i \cdot [1 - t_j/(T \cdot N)] + s_j \cdot t_j \cdot \delta_{ij}/N,$$

где i — номер пользователя, для которого пересчитываются a_i и b_i; j — номер пользователя, задание которого только что закончилось; t_j — время процессора (мин), которое потребило это задание; $T = 1000$ — условная длительность «суток» в минутах; $N = 7$ — условная длительность «недели» в днях; δ_{ij} — символ Кронекера (равен 1 при $i = j$ и равен 0 при $i \neq j$); s — коэффициент срочности (см. ниже).

Коэффициент срочности назначается автором задания, как целое число 1, 2 или 3 (по умолчанию полагается равным 1), и приводит к выигрышу в приоритете данного задания, но ухудшает прохождение последующих заданий, так как время выполнившегося задания засчитывается автору в увеличенном размере.

Динамику изменения величин a_i и b_i можно понять на примере предельных случаев. Если i-й пользователь не ставит заданий в очередь и выполняются только «чужие» задания, то обе величины экспоненциально стремятся к нулю, только a_i уменьшается в e раз за время процессора, равное суткам, а b_i — за неделю. Соответственно, если работает только i-й пользователь, то его характеристики выработки экспоненциально стремятся к T, только с разной скоростью.

Достаточно сложным является вопрос, как делить время ЭВМ — между пользователями или между группами пользователей. Для каждого пользователя более комфортными являются условия, когда ему выделяется индивидуальная норма времени, однако тогда суммарное время группы используется не всегда рационально. В нашей системе принят следующий порядок: введено три уровня администраторов, каждому администратору его вышестоящим администратором выделяется квота процессорного времени в минутах, которую он может свободно перераспределять между подчиненными ему группами пользователей и отдельными пользователями по своему усмотрению. В итоге каждый пользователь имеет квоту времени q_i минут, которую он может в любой момент узнать, выдав запрос «диспетчеру». Администратор группы в соответствии с планами работ группы имеет возможность перераспределять машинное время между членами группы, что способствует более рациональному использованию времени группы.

Теперь можно ввести приоритет на основе величин a_i, b_i и квоты q_i. К ранее введенным величинам следует добавить заказ на время CPU в минутах r_i в задании, которому требуется приписать приоритет.

Основным (и очевидным) требованием при выборе формулы

для приоритета является монотонная зависимость от a_i и b_i. Чем больше a_i или b_i при прочих равных условиях, тем меньше должен быть приоритет задания.

Простота формулы тоже имеет большое значение. Необходимо помнить, что формула часто используется для пересчета приоритетов. Пользователей на ВЦ — несколько сотен, заданий поэтому тоже может быть несколько сотен. Чем проще формула и чем нагляднее механизм ее действия, тем лучше.

Также необходимо предусмотреть, чтобы пользователь не мог занять монопольно машину на долгий срок, даже если у пользователя величины a_i и b_i по каким-то причинам занулились (занимался другими делами, был в отпуске и т. д.). Это стимулирует равномерную работу пользователей и облегчает планирование собственной работы на ЭВМ.

В соответствии с этими требованиями мы выбрали приоритет как ближайшее целое число к величине

$$p = 1000 \cdot \left(1 - \frac{a_i + r_i/N}{s_i q_i} \; \frac{q_i + 2b_i}{2q_i + b_i}\right). \qquad (2)$$

Сумма квот по всем пользователям не обязана совпадать с длительностью «суток» T, однако примерное равенство желательно для эффективного использования всего диапазона приоритетов. Наибольшее значение приоритета в формуле (2) равно 1000, наименьшее определяется предельными ресурсами пользователя. Максимально возможное время задания определено на нашем ВЦ равным $t_{max} = 100$ минутам, максимальное значение для a_i и b_i примерно равно $T = 1000$ минут. Минимальная квота равна 1 минуте. Если подставить все эти числа в формулу (2), то получим число, примерно равное -2000000.

Основная зависимость приоритета от квоты и предыдущего использования ЭВМ пользователем заложена в первой дроби в формуле (2). Зависимость приоритета от заказанного времени r_i сделана несколько ослабленной по следующим соображениям.

На первый взгляд, кажется логичным совсем не включать параметр r_i в приоритет, так как после исполнения задания реально потребленное время учтется в a_i и b_i и повлияет на приоритеты последующих заданий. Однако, у пользователей с малыми квотами в этом случае открывалась бы возможность несанкционированно потреблять большое время CPU, пропуская большие задания. «Диспетчер» при этом за неделю «забудет» об этом злоупотреблении и можно будет при любой малой квоте иметь среднее потребление времени около t_{max}/N минут. К тому же величина t_{max} — максимально допустимое время задания, получила бы неоправданно большое влияние на потребление времени.

С другой стороны, использовать при вычислении приоритета сумму $(a_i + r_i)$, как бы считая задание уже выполнившимся, тоже нерационально, так как длинное задание может долго стоять в очереди, а затем в самом начале закончиться аварийно, что часто и происходит.

В нашей формуле выбран некоторый промежуточный случай, ограничивающий возможности злоупотребления со стороны пользователей с малыми квотами и не понижающий чрезмерно приоритет задания из-за еще не использованного времени.

При регулярной работе пользователя его средне-суточная и средне-недельная выработка должна колебаться около величины его квоты q. При этом условии и коэффициенте срочности $s = 1$ приоритет заданий примерно равен 0.

Вторая дробь в формуле (2) учитывает средне-недельную дневную выработку b_i. Если пользователь по каким-то причинам смог использовать очень большое время CPU, так что величина b_i стала существенно больше его квоты, то вторая дробь будет иметь значение, близкое к 2, что можно представить себе, как деление пополам квоты q_i в первой дроби. Это означает, что пользователь в течение недели будет иметь половинную квоту.

Аналогично, пользователь с нулевой средне-недельной выработкой будет в течение недели иметь удвоенную дневную квоту.

Естественно, это качественные рассуждения. Все зависимости приоритета от использованных параметров плавные и никаких резких изменений в возможностях пользователя не происходит.

Система является самонастраивающейся. Если какая-либо группа пользователей почему-то не работает на ЭВМ, то их неиспользованное процессорное время делится между работающими пользователями пропорционально их квотам. Это качественно отличает описываемые принципы планирования от «квази-денежных» принципов, где у работающих пользователей может исчерпаться их бюджет и планирование для них перестанет работать.

4. ЗАКЛЮЧЕНИЕ

Подсистема ДИСПЕТЧЕР ЕС ЭВМ реализована и осуществляет планирование очереди заданий по описанным правилам с января 1986 года. За время эксплуатации, по сравнению с первоначальным проектом, внесено мало изменений, в основном, непринципиального характера. Например, интервал значений приоритета от нуля в сторону отрицательных чисел подвергается монотонному преобразованию, такому, что нулевой приоритет переходит также в нулевой приоритет, а приоритет минус бесконечность переходит в -10000:

$$\tilde{p}=p/(1-p/10000), \quad p<0. \tag{3}$$

Это сделано для удобства представления отрицательных приоритетов, порядок заданий в очереди при этом не меняется. В очень короткий срок после введения этой системы планирования очереди заданий исчезло социальное напряжение. Времени, конечно, больше не стало, но теперь все управляется одним параметром — квотой. Если администратору какой-либо группы требуется добавить время кому-либо из своей группы, то он может выполнить перераспределение времени между членами группы самостоятельно, не согласовывая свои действия с вышестоящим администратором.

Существенно уменьшилась зависимость пользователей друг от друга. Теперь не требуется разбираться в каких-либо деталях прохождения отдельных заданий, с нарушителями «конвенций» и т. д. Мощным рычагом в сглаживании проблемы обеспечения пользователей вычислительной мощностью стала индивидуальная заинтересованность каждого в экономии машинного времени. Можно сказать, что экономия времени ЭВМ стала общим делом.

ЛИТЕРАТУРА

1. *Букин А.Д., Дворников Н.С., Романов А.В., Сидоров В.А., Сысолетин Б.Л.* Организация использования ЭВМ ЕС-1040. — Препринт ИЯФ 82-13. Новосибирск, 1982.
2. *Chadwick K.* Design, Implementation, and Operation of a Class Based Batch Queue Scheduler for VAX/VMS. — Preprint FNAL FERMILAB-Conf-88/43, Batavia, 1988.

А.Д. Букин, Л.А. Миленький, В.А. Сидоров

Организация пакетной обработки заданий

Ответственный за выпуск С.Г.Попов

Работа поступила 28 ноября 1988 г.
Подписано в печать 26.11.1988 г. МН 08623
Формат бумаги 60×90 1/16 Объем 0,6 печ.л., 0,5 уч.-изд.л.
Тираж 190 экз. Бесплатно. Заказ № 157

Набрано в автоматизированной системе на базе фотонаборного автомата ФА1000 и ЭВМ «Электроника» и отпечатано на ротапринте Института ядерной физики СО АН СССР,
Новосибирск, 630090, пр. академика Лаврентьева, 11.