

В. П. ИВАННИКОВ

ИСПОЛЬЗОВАНИЕ КЛАСТЕРОВ В ОПЕРАЦИОННОЙ СИСТЕМЕ

(Представлено академиком А. А. Дородницким 17 VI 1977)

При формировании понятия кластера ⁽¹⁾ был сделан важный шаг в способе обработки данных: объединение объекта и операций над объектом. В данной работе предполагается включить в кластер средства синхронизации, что создает возможность принципиально нового подхода к построению операционных систем.

Операционные системы (ОС) устанавливают и управляют взаимодействием различных объектов таких, как задание, задача, сообщение, файл и т. д. Поскольку ОС имеет дело с объектами, которые, вообще говоря, взаимодействуют асинхронно, необходимо, чтобы в ОС существовали средства синхронизации. Традиционный метод решения этой задачи заключается в следующем. В ядро ОС ⁽²⁾ включаются элементарные типы объектов синхронизации, например, семафор и событие. Составные объекты, взаимодействующие асинхронно, включают в себя элементарные объекты синхронизации. Хотя такой подход широко распространен, он имеет существенный дефект, заложенный в самой элементарности и отчужденности средств синхронизации от синхронизируемых составных объектов. Именно в этом кроется сложность разработки и отладки операционных систем. Это и является препятствием для удобного и простого распараллеливания вычислений. В этой связи понятен поиск новых средств синхронизации, которые бы явно содержали информацию о синхронизируемых объектах ⁽³⁾.

Говоря о взаимодействии объектов, мы неявно подразумевали взаимодействие программ управления объектом. Таким образом, более естественно рассматривать не собственно объект, а совокупность объекта и программы управления объектом. В этом мы будем следовать кластерному подходу в описании и управлении объектом.

Предполагается следующая трактовка понятия кластера. Кластер представляет собой объединение собственно объекта и программы управления данным объектом, иными словами, существует неразрывная пара: объект и программа управления, которая обеспечивает некоторый набор операций над данным объектом. Например, файл не существует сам по себе. Он находится в банке данных в паре с программой управления, которая выполняет операции над этим файлом. Двум разным файлам одного типа и, следовательно, имеющим одинаковый интерфейс (одинаковый набор операций), могут быть сопоставлены, вообще говоря, разные программы управления. Справочник, в котором описан данный кластер-файл, сам в свою очередь является кластером-файлом со своим собственным набором операций.

Кластер является основным универсальным объектом, с которым имеет дело ядро ОС. Ядро обеспечивает средства определения, активизации—деактивизации кластеров, вызова кластера другим кластером. Ядро операционной системы не вникает в детали управления объектом; для него кластер-задание, кластер-задача, кластер-терминал суть объекты одной природы. Вся специфика управления реальным объектом сосредоточена в самом кластере реального объекта.

Ядро ОС дает средства порождения кластера другим кластером. При определении кластера указываются следующие параметры: тип кластера

(Этот параметр определяет интерфейс с данным кластером); программа управления; дополнительные параметры, обрабатываемые этим кластером. Например, для кластера-файла в дополнительных параметрах содержится информация о кластере-дисковый пакет, на котором размещен файл. Кластер выполняет некоторую совокупность операций, каждой операции соответствует вход в кластер. Входы могут быть либо статическими и содержаться в описателе кластера, либо динамическими, порождаемыми при передаче параметров в обращении из одного кластера к другому.

Для своей работы кластеры могут требовать подключения других кластеров. В частности, кластер-задача должна быть связана с другими кластерами такими, как файл, терминал и т. д. Для кластера-задачи действия по подключению кластеров может выполнять интерпретатор языка управления заданиями, используя операции, предоставляемые ядром. Эти операции ядра по активизации — деактивизации кластеров следующие: «Подключить» кластер, «Отключить» кластер. Операция «Подключить» обеспечивает установление связи между двумя кластерами. Операция «Отключить» кластер вызывает отторжение кластера и разрыв связи между кластерами.

В операции «Подключить» указываются следующие параметры: тип кластера, имя кластера-объекта, дополнительные параметры. Параметр тип кластера определяет интерфейс, который необходим кластеру, давшего операцию «Подключить». Этот тип может не совпадать с типом кластера-объекта, заданным при определении. В этом случае подключается кластер-посредник, который будет осуществлять сопряжение между кластерами. Тем самым решается задача обеспечения независимости объекта от изменения окружающей среды.

Подключаемый кластер должен быть настроен (активизирован), поскольку ему могут понадобиться другие кластеры. Над любым кластером определены операции «Активизировать», «Деактивизировать». Эти операции разрешены только для ядра. Ядро ОС при выполнении операции «Подключить» обращается к кластеру с операцией «Активизировать». Активизированный кластер может обратиться к ядру с операцией «Подключить» и т. д. Таким образом может быть порождена связанная цепочка кластеров. При активизации кластер обрабатывает дополнительные параметры, во-первых, те, которые были заданы при операции «Подключить», во-вторых, параметры, заданные при определении кластера.

При выполнении операции «Отключить» ядро обращается к кластеру по операции «Деактивизировать». Кластер выполняет ряд завершающих действий и может, в частности, обратиться к ядру с операцией «Отключить».

Выполняя активизацию, кластер может осуществить индивидуальные средства защиты (например, проконтролировать права доступа). Ядро после обращения к кластеру по активизации может получить от кластера отказ от подключения. Такой ответ обрабатывается как ошибка кластера, давшего операцию «Подключить».

По способу использования любой кластер может быть копируемым, совместным, монопольным. Информация об этом содержится в описании кластера. Для каждого кластера, требующего подключения кластера-копии, порождается экземпляр кластера. Совместный кластер в ОС существует в одном экземпляре и может быть одновременно подключенным ко многим кластерам. Монопольный кластер подключается лишь к одному кластеру, все остальные требования становятся элементами очереди на подключение кластера. Очередь разгружается на один элемент при каждой операции отключить.

Рассмотрим теперь вопросы вызова кластера кластером, способы передачи параметров при обращении и способы синхронизации. Наложим на взаимодействие кластеров следующие ограничения: каждый кластер имеет только свои собственные локальные данные; общих данных взаимо-

действующие кластеры иметь не должны; при вызове одного кластера другим в качестве параметров могут передаваться либо значения данных, либо ссылки на входы в кластеры. Проиллюстрируем способы передачи параметров на двух простейших примерах: обращение из кластера *A* к кластеру *T* (кластер *T* управляет терминалом) по печати текста на терминале и при приеме с терминала.

«Печать». *A*, обращаясь к *T*, определяет в качестве параметра сам текст.

«Прием». *A*, обращаясь к *T*, передает кластеру *T* параметр *A1* — вход в кластер *A*. При обращении из *T* по входу *A1* параметром является принятый текст. Этот текст кластер *A* размещает в своих собственных локальных данных.

Очевидно, что уже в приведенном примере «Прием» подразумевается наличие асинхронности. Действительно, кластер *A* должен уметь ждать появления на входе *A1* принятого текста и размещения его в локальных данных, поскольку кластеру *A*, как и любому кластеру, доступны только собственные локальные данные.

Итак, кластер представляет собой замкнутый объект, имеющий только входы, над которыми определен базовый набор операций. Поскольку синхронизация доступа к объекту в действительности является синхронизацией по выполнению операций над объектом, очевидно, что синхронизоваться должны операции (входы в кластер).

Любой кластер теперь может выполняться асинхронно относительно других кластеров. Синхронизация взаимодействия асинхронных кластеров осуществляется захватами входов в кластер. Любой вход в кластер может находиться в одном из двух состояний: захвачен монополюсно или захвачен совместно. При монополюсном захвате вход содержит идентификатор кластера-захватчика. Кластер может быть монополюсным захватчиком своих собственных входов. По совместному входу кластера разрешается обращение от всех кластеров, подключенных к данному кластеру, по монополюсному только кластеру-захватчику. Если некоторый кластер *A* обращается по входу кластера *B*, а этот вход захвачен монополюсно кластером *C*, то кластер *A* ставится в очередь к этому входу кластера *B*. Очередь разгружается при соответствующем изменении состояния входа.

Состояния входов изменяют операции над кластерами «Пуск» и «Стоп», которые будут рассмотрены ниже.

Любое обращение от кластера к кластеру контролируется ядром ОС. Кластер может находиться в одном из трех состояний: выполняется, ждет (т. е. находится в очереди ожидания), остановлен (готов принять обращение от других кластеров). Обращение к кластеру происходит только, если кластер находится в состоянии «остановлен». Если некоторый кластер *B* находится в состоянии «выполняется» или «ждет», ядро ОС, осуществляя взаимное исключение, ставит в очередь все кластеры, обратившиеся к кластеру *B* по любым входам. Очередь разгружается ядром ОС при переходе в состояние «остановлен».

Обращение от кластера к кластеру осуществляется операцией «Пуск», обрабатываемой ядром. Операция может содержать следующие параметры: совокупность значений данных, ссылки на входы в кластер с проставлением захватов, ссылку на вход вызываемого кластера. Вызывающий кластер продолжает свое выполнение. Вызванный кластер (если он находился в состоянии «остановлен») начинает выполняться асинхронно. Параметры пересылаются ядром в локальную область параметров вызванного кластера.

Выполнение кластера завершается операцией «Стоп». Эту команду дает выполняющейся кластер. Операция обрабатывается ядром ОС и содержит следующие параметры: ссылки на входы кластеров с проставлением захватов.

Рассмотрим примеры, демонстрирующие операции над входами асинхронных кластеров.

Пример 1. Прием с терминала. Кластер A дает команду «Пуск» T , передавая параметр $A1$, монополюно закрепляя его за кластером T . Кластер A начинает выполняться асинхронно с кластером T . Выполнение A заканчивается операцией «Стоп» без параметров. Кластер T дает команду «Пуск» $A1$, передавая тест и возвращая параметр $A1$. Кластер A переписывает текст в свои локальные области данных и продолжает выполнение. Кластер T завершается командой «Стоп» без параметров. Если кластер T обратится по входу $A1$ до момента выдачи A команды «Стоп», то кластер T будет поставлен в очередь к A до тех пор, пока A не перейдет в состояние «остановлен».

Пример 2. Обработка файла. Кластер Φ обеспечивает доступ к файлу в двух режимах: в режиме модификации записей файла и в режиме чтения. В режиме чтения с кластером Φ могут взаимодействовать несколько кластеров, в режиме модификации только один. Кластер Φ имеет следующие входы: Открыть M — открыть файл для модификации, Открыть $Ч$ — открыть файл в режиме чтения, Закрыть, Читать, Писать.

Первоначально входы в кластер находятся в следующем состоянии: Открыть M и Открыть $Ч$ — в совместном; Закрыть, Читать, Писать — монополюно с захватчиком Φ .

Если произошло обращение по входу Открыть M , то кластер Φ , завершая выполнение по Открыть M командой «Стоп», указывает в параметрах входы Закрыть, Читать, Писать, закрепленными монополюно за обратившимся кластером, а входы Открыть M , Открыть $Ч$ захватывает монополюно сам.

После обработки Закрыть кластер Φ командой «Стоп» переводит свои входы в начальное состояние.

Аналогично происходит управление входами в режиме чтения. При входе по Открыть $Ч$ кластер Φ захватывает вход Открыть M и открывает для совместного использования входы Закрыть, Читать. При входе Закрыть кластер Φ сравнивает число открытий и закрытий и, если они совпали, командой «Стоп» переводит входы в начальное состояние.

Данная трактовка понятия кластера и введение понятия асинхронного кластера дает способ структурированного построения ОС, дает удобный способ распараллеливания взаимодействий не только объектов, традиционно входящих в сферу ОС, но и для распараллеливания вычислений вообще.

Поступило
10 VI 1977

ЛИТЕРАТУРА

- ¹ B. Liskov, S. Ziller, SIGPLAN Notices, № 9 (1974). ² P. B. Hansen, Commun. ACM, v. 13, № 4 (1970). ³ C. A. R. Hoare, *ibid.*, v. 17, № 10 (1974).