İNFORMATİKA

КОНТРОЛЬНЫЕ ТОЧКИ И ОТКАТ-ВОССТАНОВЛЕНИЕ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

А.А.АЛИЕВ, Р.Э.ФАТУЛЛАЕВ
Бакинский Государственный Университет

В работе рассматриваются проблемы обеспечения отказоустойчивости распределенных систем. Более подробно описывается механизм отката-восстановления на основе аппарата контрольных точек. Предлагается алгоритм, который позволяет не останавливая функционирования системы, формировать глобальную контрольную точку. Алгоритм обеспечивает, чтобы ни одна подтранзакция не отвергалась. Приводится утверждение о том, что алгоритм формирования глобальных контрольных точек корректен.

Введение. Â ïîñëåäíåå âpåìÿ âû÷èñëèòåëüíûå ñèñòåìû (ÂÑ) âñå øèpå ïpèìåíÿþòñÿ â ñèñòåìàõ è ïpîöåññàõ óïpàâëåíèÿ, ÷òî âûäâèãàåò pÿä òpåáîâàíèé, êîòîpûå îòëè÷àþò ýòè ñèñòåìû îò ñèñòåì, òpàäèöèîííî èñïîëüçóåìûõ äëÿ íàó÷íî-òåõíè÷åñêèõ pàñ÷åòîâ. Ñpåäè ýòèõ òpåáîâàíèé îäíèì èç âàæíåéøèõ ÿâëÿåòñÿ âûñîêàÿ íàäåæíîñòü pàáîòû. Hàäåæíàÿ ÂÑ â ñîñòîÿíèè ãîòîâíîñòè äîëæíà ïpè ëþáîì íápàùåíèè ê íåé íáåñïå÷èâàòü ïîëüçîâàòåëåé àäåêâàòíîé èíôîpìàöèåé. Ñàìî æå ñîñòîÿíèå ãîòîâíîñòè äîëæíî îõâàòûâàòü âîçìîæíî áîëüøèé ïpîöåíò âpåìåíè.

Â ïîñëåäíåå âpåìÿ â pàìêàõ îáùåé ïpîáëåìû íàäåæíîñòè âîçíèêëî íîâîå íàïpàâëåíèå – îòêàçîóñòîé÷èâîñòü ÂÑ. Îòêàçîóñòîé÷èâûìè íàçûâàþòñÿ òàêèå ÂÑ, êîòîpûå ñïîñîáíû ïpîäîëæàòü ôóíêöèîíèpîâàíèå ïpè âîçíèêíîâåíèè pàçíîîápàçíûõ îòêàçîâ êîìïîíåíòîâ. Â ñâÿçè ñ pàñøèpåíèåì ïîòpåáíîñòåé â îòêàçîóñòîé÷èâûõ ÂÑ è çíà÷èòåëüíûì ñíèæåíèåì ñòîèìîñòè ýëåêòpîííûõ êîìïîíåíòîâ â áëèæàéøåì áóäóùåì îæè-äàåòñÿ, ÷òî ïîäîáíûå ñèñòåìû íàéäóò øèpîêîå ïpèìåíåíèå. Âîçìîæíûìè îáëàñòÿìè ïpèìåíåíèÿ îòêàçîóñòîé÷èâûõ ñèñòåì ÿâëÿþòñÿ: óïpàâëåíèå ñïóòíèêîâîé ñâÿçüþ, àâèàpåéñàìè, ïpîìûøëåííûìè pîáîòàìè, àâòîìàòèçèpîâàííûìè ýëåêòpîñòàíöèÿìè è ò.ä. Â áîëüøèíñòâå ïîäîáíûõ ïpèìåíåíèé äëÿ îáåñïå÷åíèÿ ýôôåêòèâíîãî ôóíêöèîíèpîâàíèÿ ÂÑ íåîáõîäèìà èõ ïîëíàÿ óñòîé÷èâîñòü ê íåáîëüøèì îòêàçàì. Ïpè ýòîì äîïóñêàåòñÿ íå-êîòîpîå óìåíüøåíèå ïpîèçâîäèòåëüíîñòè ÂÑ. Â íàñòîÿùåå âpåìÿ pàáîòà â îáëàñòè îòêàçîóñòîé÷èâîñòè ÂÑ âåäåòñÿ âåñüìà èíòåíñèâíî ñ öåëüþ ïîëó÷åíèÿ íîâûõ òåîpåòè÷åñêèõ påçóëüòàòîâ è ïpàêòè÷åñêèõ pàçpàáîòîê.

Îòêàçîóñòîé÷èâîå ôóíêöèîíèpîâàíèå ñèñòåìû îáåñïå÷èâàåòñÿ çà ñ÷åò ââåäåíèÿ pàçíîîápàçíûõ ôîpì èçáûòî÷íîñòè: àïïàpàòíîé, ïpîãpàììíîé è âpåìåííîé, îáåñïå÷èâàåìîé êàê íà àïïàpàòíîì, òàê è íà ïpîãpàììíîì ópîâíÿõ. À âñÿêàÿ èçáûòî÷íîñòü, êàê èçâåñòíî ïîâûøàåò ñòîèìîñòü ñèñòåìû. Ïpè ïpàâèëüíîì

ïpîåêòèpîâàíèè ñèñòåì íåîáõîäèìî ñòpåìèòüñÿ ìèíèìèçèpîâàòü èõ ñòîèìîñòü è ìàêñèìèçèpîâàòü ýôôåêòèâíîñòü.

Ñâîéñòâîì îòêàçîóñòîé÷èâîñòè îáëàäàþò ìíîãèå òåõíè÷åñêèå ñèñòåìû, íî êîìïüþòåpíûå ÿâëÿþòñÿ íàèáîëåå õàpàêòåpíûìè, òàê êàê îíè ñîïñîáíû äàäïòèpîâàòüñÿ ê èçìåíÿþùèìñÿ óñëîâèÿì, ò.å. ïåpåñòpàèâàòü àëãîpèòìû ñâîåãî ôóíêöèîíèpîâàíèÿ â øèpîêîì äèàïàçîíå. Ñpåäè êîìïüþòåpíûõ ñèñòåì ñâîéñòâî îòêàçîóñòîé÷èâîñòè â íàèáîëüøåé ñòåïåíè ïpèñóùå, âî âñÿêîì ñëó÷àå ïîòåíöèàëüíî, pàñïpåäåëåííûì ñèñòåìàì (PC), ôóíêöèîíèpóþùèì íà îñíîâå êîìïüþòåpíîé ñåòè [1]. Â ÐÑ ìîæíî îáåñïå÷èòü óñòîé÷èâîñòü ê îòêàçàì îòäåëüíûõ ñpåäñòâ, äàæå â óñëîâèÿõ pàñïàpàëëåëèâàíèÿ ïpîöåññîâ îápàáîòêè èíôîpìàöèè.

Механизмы обеспечения отказоустойчивости. Ìåõàíèçìû ïîâûøåíèÿ îòêàçîóñòîé÷èâîñòè pàçíîápàçíû. Îñíîâíîé ñëîæíîé ÿâëÿåòñÿ çàäà÷à pàçpàáîòêè òàêèõ ìåõàíèçìîâ îòêàçîóñòîé÷èâîñòè, êîòîpûå ïîçâîëÿò ïîñpàíèòü öåëîñòíîñòü pacпределенных баз данных (ÐÁÄ) [2]. Â îáùåì ñëó÷àå îòêàçîóñòîé÷èâîñòü íåâîçìîæíî ñ ïîìîùü ñëåäóþùèõ ìåõàíèçìîâ:

- îáíàpóæåíèå îòêàçà â ñèñòåìå;
- äèàãíîñòèpîâàíèå îòêàçàâøåãî óñòpîéñòâà;
- óñòpàíåíèå âëèÿíèÿ îòêàçàâøåãî óñòpîéñòâà (påîpãàíèçàöèÿ ñèñòåìû);
- âîññòàíîâëåíèå íîpìàëüíîãî ôóíêöèîíèpîâàíèÿ ÂÑ.

Âñå ýòè ìåõàíèçìû ÿâëÿþòñÿ íåîòúåìëåìûìè ÷àñòÿìè îòêàçîóñòîé÷èâîé ñèñòåìû è ìîãóò pàáîòàòü àïïàpàòíûì, ïpîãpàììíûì èëè ñìåøàííûì ïpîãpàììíî-àïïàpàòíûì ñïîñîáîì.

Ñpåäè ìåõàíèçìîâ îáåñïå÷åíèÿ îòêàçîóñòîé÷èâîñòè ÂÑ îáÿçàòåëüíûìè ÿâëÿþòñÿ ìåõàíèçìû âîññòàíîâëåíèÿ, êîòîpûå çàïóñêàþòñÿ ñèãíàëàìè îò àëãîpèòìîâ îáíàpóæåíèÿ íåèñpàâíîñòé, è çàäà÷åé êîòîpûõ ÿâëÿåòñÿ âîçpàò ñèñòåìû ê íîpìàëüíîìó ôóíêöèîíèpîâàíèþ. Âûâîp ìåòîäîâ âîññòàíîâëåíèÿ îpèåíòèpîâàí íà îòêàçû îïpåäåëåííîãî òèïà, à ïîñëåäîâ íîñüìà pàçíîápàçû, òî ñóùåñòâóåò è ìíîãî pàçíûõ ìåòîäîâ âîññòàíîâëåíèÿ. Âîññòàíîâëåíèå ñîñòîèò èç âñåõ òåõ äåéñòâèé, êîòîpûå îñóùåñòâëÿþòñÿ ïî ïîéó÷åíèè ñèãíàëà íåèñpàâíîñòè.

Håâûïîëíåíèå ìåõàíèçìû âîññòàíîâëåíèÿ ñâîéñòâ ôóíêöèé ïpèâîäèò ê îòêàçó ñèñòåìû. Âîññòàíîâëåíèå âîçìîæíî ïîñpåäñòâîì îòêàòà, èñïpàâëåíèÿ, ìàñêèpîâàíèÿ îòêàçà è påêîíôèãópàöèè.

Одним из ìåòîäîâ âîññòàíîâëåíèÿ íîpìàëüíîãî ôóíêöèîíèpîâàíèÿ ñèñòåìû ïîñëå îáíàpóæåíèÿ îøèáêè ÿâëÿåòñÿ îòêàò, îñóùåñòâëÿþùèé âîçâpàò ñîñòîÿíèÿ ñèñòåìû ê èñïpàâíîìó ñîñòîÿíèþ, ïpåäøåñòâóþùåìó åå ïåpåõîä â íåèñpàâíîå ñîñòîÿíèå, ïpè÷åì ïîñëå äàëüíåéøåãî ôóíêöèîíèpîâàíèÿ çàâèñèò îò ïpèìåíÿåìîãî ìåòîäà îápàáîòêè îòêàçà. Ýòîò ìåòîä çàâèñèò îò ñîïñîáíî påàëèçàöèè òî÷êè âîçâpàòà, ò.å. ïpîãpàììí-àïïàpàòíé êîíñòpóêöèè, ñ ïîìùüþ êîòîpîé ñîñòîÿíèå ñèñòåìû ìîæåò áûòü çàôèêñèpîâàíî è ïîçäíåå âîññòàíîâëåíî.

Òàêèì îápàçîì, ñpåäè ìåòîäîâ âîññòàíîâëåíèÿ ñàìûì ïpîñòûì, óíèâåpñàëüíûì è óäîáíûì ÿâëÿåòñÿ îòêàò. Ñëåäîâàòåëüí, ìæíî ñêàçàòü, ÷òî îäèí èç îñíîâíîõ ìåõàíèçìîâ îáåñïå÷åíèÿ îòêàçîóñòîé÷èâîñòè ñèñòåìû ÿâëÿåòñÿ ìåõàíèçì îòêàò-âîññòàíîâëåíèÿ. Ýòîò ìåõàíèçì ïîçâîëÿåò âåpíóòü ñèñòåìó â íåêîòîpîå èñïpàâíîå ñîñòîÿíèå ïîñëå îáíàpóæåíèÿ îøèáêè â pàáîòå ñèñòåìû, à çàòåì âîññòàíîâèòü

åå ñîñòîÿíèå, èìåâøåãñÿ ê ìîìåíòó îáíàðóæåíèÿ îøèáêè è ïðîäîëæàòü íîðìàëüíîå ôóíêöèîíèðîâàíèå.

Будем говорить, что система непротиворечива, если значения ее объектов удовлетворяют всем ограничениям целостности. Ограничения целостности – это множество формальных правил, устанавливающих и регламентирующих связи между объектами РБД. Они представляют собой некие предикаты, которым должны удовлетворять данные. Ограничения целостности в той или иной мере реализуются в узлах и проверяются, как правило, на этапе загрузки баз данных. Однако стоимость такого контроля велика и стремительно возрастает с ростом его полноты. Поэтому, как правило, выбирают компромиссный путь: контроль ограничен, но во всех узлах на запоминающее устройство периодически копируются локальные базы данных (ЛБД) и накапливаются локальные контрольные точки (ЛКТ) с тем, чтобы при обнаружении ошибки в работе компьютера быстро произвести откат к этой точке [3]. При этом нужно решить две проблемы:

- обеспечение максимально возможной уверенности в правильности ЛКТ и в отсутствии отказов при ее накоплении;
- обеспечение уверенности в том, что за период времени между возникновением отказа, который произошел после накопления последней ЛКТ и обнаружением ошибки неверные данные не распространились в другие узлы.

Первая проблема решается средствами внутреннего контроля при накоплении ЛКТ (повторные просчеты, сравнения и т.д.). Вторая – скоординированным откатом ЛБД в узлах, в которые отказавший узел при подозрении на отказ посылал сообщения.

После отката система инициирует процесс восстановления. Восстановлением называется приведение системы из непротиворечивого состояния, полученного в результате отката, в непротиворечивое состояние, близкое тому, при котором был обнаружен отказ.

Система должна повторно инициировать и выполнять те транзакции, результаты работы которых были аннулированы при откате, и те, которые поступили в систему во время реализации отката и повторного выполнения указанных выше транзакций. Необходимо отметить, что система возвращается в состояние, близкое тому, при котором был обнаружен отказ, а не точно в то же состояние, ведь повторный процесс с некоторого момента времени (соответствующего моменту возникновения отказа первичного процесса) не повторяет первичный процесс, так как отказ уже не возникает.

Введение контрольных точек (КТ) требует дополнительных ресурсов, как временных, так и ресурсов памяти, что снижает производительность системы. Но, с другой стороны, КТ позволяют сократить время восстановления баз данных в случае ее разрушения. С увеличением числа КТ снижается общая производительность, но уменьшается время восстановления, и наоборот, с увеличением интервала между соседними КТ умень-

шается потеря производительности, но увеличивается время восстановления. Поэтому, очень важна задача определения рационального числа КТ и временного интервала для проведения защиты протоколов изменений. При этом должны учитываться такие параметры как частота изменения БД и надежность работы системы. Для определения моментов проведения защиты протоколов изменений, дополнительно нужно учитывать размер файла системного журнала. Чем меньше его размер, тем чаще должна быть защита, так как она не может быть реже, чем момент полного заполнения файла, о чем система выдает соответствующие подсказки. Процедура записи КТ помещает в системный журнал специальную запись, содержащую текущее состояние всех системных компонентов. При рестарте эта информация используется для восстановления состояния системы.

Однако существует потенциальная возможность, что в связи с откатом, т.е. возвратом к некоторому старому, но целостному состоянию, возникает подозрение в целостности в контрольной точке. Это будет иметь место в том случае, если процесс получил сообщение до момента накопления КТ. При неудачно сложившейся последовательности сообщений, которыми обменивались процессы, после первой серии откатов может начаться следующая, и в худшем случае откаты могут продолжаться до тех пор, пока все процессы не перейдут в первоначальное состояние. Такое явление, описанное впервые В.Ранделлом, называется “эффектом домино”.

Одним из способов предотвращения «эффекта домино», является формирование глобальных контрольных точек (ГКТ), которые являются обобщением понятия КТ вычислительного процесса на распределенную вычислительную систему. ГКТ служит не только для рестарта РС при отказах оборудования, но и является основой решения некоторых специфических прикладных задач в РС, например для ревизии счетов распределенной банковской системы, переписи населения и др.

Процесс формирования ГКТ инициируется одним из видов транзакций. Для того чтобы транзакция формирования ГКТ дала значимый результат, отдельные этапы формирования ГКТ должны быть четко скоординированы с этапами выполнения других транзакций [4]. Все ГКТ нумеруются по порядку, т.е. ГКТ=r , где r = 1,2,... Все транзакции и порожденные ими подтранзакции принадлежат одной и только одной ГКТ= r.

Ниже приводится алгоритм формирования глобальных контрольных точек.

Алгоритм

В начальном состоянии будем считать, что все узлы РС находятся в состоянии ГКТ=r-1. Алгоритм формирования ГКТ, аналогичен протоколу двухфазной фиксации.

Первая фаза:

Шаг 1. Узел-инициатор отправляет всем узлам распределенной системы сообщение о формировании ГКТ=r, в том числе и в узел-инициатор (*T(r)*).

Шаг 2. Каждый узел, получив такое сообщение, блокирует ресурс, т.е. ЛБД *(lock R)*. Копирует содержимое ЛБД в стабильную память, т.е. записывает текущее состояние ресурса в стабильную память *(write-SM)*. Эту копию будем называть промежуточной ЛКТ=r. Далее освобождает ресурс *(unlock R)*.

Шаг 3. Если какой-нибудь узел получил подтранзакцию с номером ГКТ=r-1 *(T(ГКТ=r-1))*, то:

а) если промежуточная ЛКТ=r не установлена, то подтранзакция с номером ГКТ=r-1 выполняется *(comply);*

б) если промежуточная ЛКТ=r уже установлена, то подтранзакция с номером ГКТ=r-1 выполняется над ЛБД *(comply)*, а не над промежуточной ЛКТ=r, и записывается в упорядоченный список q *(inqueu (T/q))*.

Шаг 4. Если какой-нибудь узел получил подтранзакцию с номером ГКТ=r *(T(ГКТ=r))*, то:

а) если промежуточная ЛКТ=r уже установлена, то подтранзакция с номером ГКТ=r выполняется над ЛБД *(comply);*

б) если промежуточная ЛКТ=r пока не установлена, то подтранзакция с номером ГКТ=r записывается в упорядоченный список Q *(inqueu (T/Q))*. Узел ожидает установления промежуточной ЛКТ=r *(wait(ЛКТ=r)),* после чего данные подтранзакции из Q выполняются над ЛБД *(comply)*, а из списка Q удаляются *(delete(T/Q))*.

Шаг 5. Каждый узел после установления промежуточной ЛКТ=r сообщает узлу-инициатору об этом *(ready)*.

Вторая фаза:

Шаг 6. Получив от всех узлов сообщение об установлении промежуточных ЛКТ=r *(ready)*, узел-инициатор отправляет всем узлам сообщение о том, чтобы фиксировать промежуточные ЛКТ=r постоянными *(commit)*.

Шаг 7. Каждый узел выполняет над своей постоянной ЛКТ=r все подтранзакции из списка q *(comply((ЛКТ=r)/q))*.

Совокупность всех полученных после шага 7 локальных состояний образует ГКТ=r.

Утверждение. Предложенный алгоритм формирования глобальных контрольных точек корректен.

Заключение. Таким образом, предложенный в работе алгоритм позволяет не останавливая функционирования системы, определить глобальную контрольную точку. Этот алгоритм обеспечивает, чтобы ни одна подтранзакция будь то *T(ГКТ=r-1)* или *T(ГКТ=r)* не отвергалась. Совокупность всех постоянных ЛКТ=r и всех списков q образует глобальное состояние распределенной системы.

ЛИТЕРАТУРА

1. Flavin Cristan Understanding Fault-Tolerant Distributed Systems // Comm. ACM. – 1991. – 43, N2. – p.56-78.
2. M.T.Ozsu, P.Valduries. Principles of Distributed Database Systems, Prentice-Hall, 1999.
3. Li X., Eich M., Joseph V., Gulzar Z. Checkpoint and recovery in partitioned main memory databases. In Proc. IASTED/ISMM International Conference on Intelligent Information Management Systems, Washington, DC. (June 1995).
4. Leu P. And Bhargava B. A model for concurrent checkpointing and recovery using transactions// Proc. 9 th Intern. Conf. On Distributed Computing Systems. – 1989. – p.423-430.

PAYLANMIŞ SİSTEMLƏRDƏ YOXLAMA NÖQTƏLƏRİ VƏ GERİQAYITMA-BƏRPA

Ə.Ə.ƏLİYEV, R.E.FƏTULLAYEV

ANNOTASİYA

İşdə paylanmış sistemlərin nasazlığa davamlılığı məsələlərinə baxılır. Yoxlama nöqtələri əsasında bərpa mexanizmi təsvir olunur. Sistemin fəaliyyətini dayandırmadan qlobal yoxlama nöqtələrinin qurulması alqoritmi təklif olunur. Bu alqoritmin korrekt olması hökmü verilir.

CHECKPOINTS AND ROLLBACK-RECOVERY IN DISTRIBUTED SYSTEMS

A.A.ALIYEV, R.E.FATULLAYEV

ABSTRACT

In the given paper the problem of ensure faulttolerance in distributed systems is considered. Description for mechanism roolback – recovery proposed on the basis of apparatus chekpoints. Proposed algorithm determine the global chekpoint, does not stop functions in distributed system.