МЕТОД ОЦЕНКИ НАДЕЖНОСТИ КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР И ОТКАЗОУСТОЙЧИВОСТИ ПРИЛОЖЕНИЙ С НЕДЕТЕРМИНИРОВАННЫМ ПОВЕДЕНИЕМ

В.А. ВОЛКОВ[1]
С.М. ЧУДИНОВ[2]

1) «НИИ ВК им. М.А. Карцева»
2) ОАО «НИИ Супер ЭВМ»
(г. Москва)

e-mail:
chudinov@supercomputer.ru

Приведены модели и получены соотношения для оценки надежности кластерных вычислительных систем. Показана важность учета надежности переключателя резерва при моделировании резервируемых структур. Предметом рассмотрения являются резервированные вычислительные системы типа «Эльбрус». Также рассмотрены вопросы обеспечения прозрачной отказоустойчивости.

Ключевые слова: кластерные системы, высокая готовность, отказоустойчивость (ОУ), проект кластерных структур, марковские процессы, коэффициент готовности (КГ), высокий уровень готовности (ВУГ) прозрачной отказоустойчивости (Transparent Fault Tolerance, TFT), механизм контрольных точек (snapshots), управляемых выполнением.

Надежность корпоративных систем обработки данных, их способность в режиме реального времени (online) обеспечивать пользователей оперативной и достоверной информацией – одно из важнейших условий эффективной работы и, в конечном счете, конкурентоспособности современных компаний. Сегодня существует множество технических решений, обеспечивающих необходимый уровень надежности и отказоустойчивости информационных систем, и один из краеугольных камней таких решений - кластеризация вычислительных систем (ВС), за счет которой поддерживается высокий уровень готовности (high availability).

Построение отказоустойчивых кластеров предполагает использование в качестве узловых машин высоконадежных процессоров (серверов) с двукратным, или многократным дублированием всех основных модулей. Только резервирование способно обеспечить значения коэффициента готовности в районе «пяти девяток» - 0,99999. В качестве типовых решений применяются отказоустойчивые системы (Fault Tolerance, FT) и системы высокой готовности (High Availability, HA). Наибольшую популярность на сегодняшний день имеют HA-кластеры.

Кластер - это объединение двух и более устройств (модулей), которые связаны между собою и функционируют как **один узел обработки информации**. Спектр предлагаемых кластерных решений весьма обширен, ниже будут рассмотрены конфигурации применительно к классу HA-кластеров, основной целю которых, является обеспечение отказоустойчивости.

Примером организации смешанной конфигурации HA-кластеров и HP-кластеров является многопроцессорный вычислительный комплекс (МВК) «Эльбрус-2» с перекрестной коммутацией межмодульных связей.

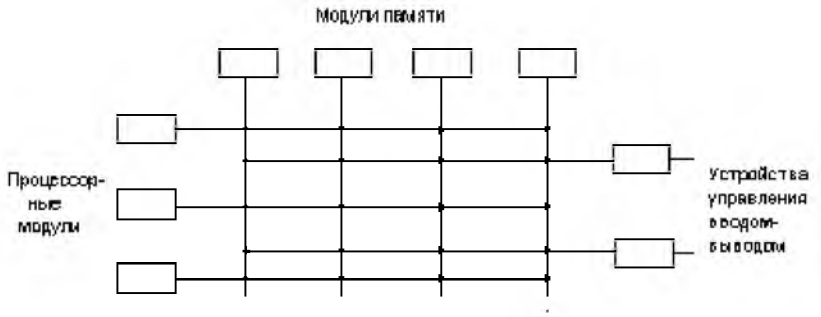

Рис. 1. Коммутатор межмодульных связей МВК «Эльбрус-2»

Глобальной проблемой оценки надежности отказоустойчивых систем, включая FT-системы, HA-кластеры и является задача прогнозирования показателей надежности разрабатываемых ВС (проектная оценка надежности). Практика показывает, что часто используются упрощенный подход к такой оценке и получение явно завышенных величин показателей надежности.

В соответствии с двумя основными существенно различными сферами использования вычислительных средств – в составе информационно-вычислительных центров и в системах реального времени – по-разному формулируется требование надежности к ним. В одном случае требования по надежности определяются коэффициентом снижения производительности информационно-вычислительного комплекса за счет отказов и сбоев аппаратуры, в другом вероятностью выполнения необходимого технологического цикла в заданный промежуток времени.

Расчет надежности дублированной группы

При расчете надежности сетевого кластера, как правило, рассматривается дублированная группа узлов. При этом отказом считается выход из строя обоих узлов. Рассмотрим модель дублированной группы с идентичными узлами, приведенную на рис 2.

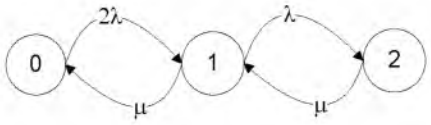

Рис. 2. Граф состояний кластера двумя узлами и идеальной системой контроля

Отказ дублированной группы наступает тогда, когда во время восстановления одного из узлов откажет второй узел. Возможные состояния:

«0» – оба узла исправны;

«1» – отказ в одном узле;

«2» – отказ в обоих узлах.

Таким образом, состояния исправности системы «0», «1», отказа «2».

В случае отказа одного из элементов группы, отказавший узел ремонтируется (заменяется) без остановки системы и после восстановления через случайный промежуток времени, распределенный по экспоненциальному закону с параметром μ, включается в состав дублированной группы: $\mu=1/T_в$, где $T_в$ - среднее время восстановления. Одновременно может восстанавливаться один узел.

Для ненагруженного режима резервирования с ограниченным восстановлением получим формулу расчета коэффициента готовности:

$$Кг = \frac{\lambda\mu + \mu^2}{\lambda^2 + \lambda\mu + \mu^2}.$$

Аналогично для дублированной системы с нагруженным резервом и неограниченным восстановлением (две ремонтные бригады) получим:

$$Кг = \frac{2\lambda\mu + \mu^2}{\lambda^2 + 2\lambda\mu + \mu^2},$$

и с ненагруженным резервом и неограниченным восстановлением:

$$Кг = \frac{2\lambda\mu + 2\mu^2}{\lambda^2 + 2\lambda\mu + 2\mu^2}.$$

Приведенные выше формулы для расчета дублированной группы сетевых узлов являются наиболее распространенными. При исходных данных интенсивности отказов λ = 0,00005 1/ч (наработка на отказ составляет 20 000 часов) и интенсивно-

сти восстановления μ = 0,25 1/ч (4 часа восстановления) значение $K_г$ =0,999 999 92 (семь девяток). Подчеркнем, что взятая наработка 20 тыс. часов – является нижней планкой MTBF (MeanTime Before Failure, средняя наработка на отказ) серверных платформ, обычно для серверов приводятся значения 50-100 тыс. часов и, следовательно, получаются еще более «хорошие» результаты.

Состояние необнаруженного отказа

В каждом элементе могут быть скрытые отказы. Модель, приведенная на рис., не учитывает вероятность обнаружения отказа, надежность «переключателя» резерва, задержку при переключении резервов и другие, учет которых возможен при введении дополнительных параметров модели.

В дополнение к множеству состояний традиционной модели, представленной на рис. 2, в модернизированной модели, приведенной на рис. 3, добавляется состояние «3» – необнаруженного средствами внутреннего (внутрикластерного) контроля отказа. Таким образом, состояниями отказа системы являются «2» и «3».

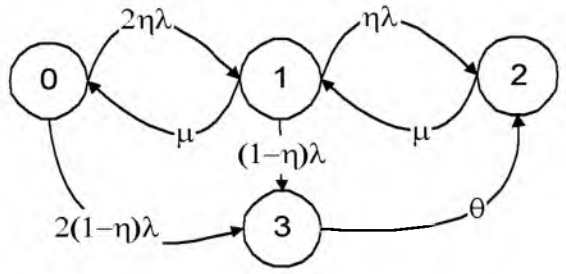

Рис. 3. Граф состояний кластера с двумя узлами и неидеальной системой контроля

Отметим, что для учета ненагруженности резерва и наличия нескольких ремонтных бригад применяются подходы к выводу зависимостей $K_г$, аналогичные приведенным выше при расчете подобных модификаций модели графа рис. 2.

С позиции контролируемости кластер представляется как дублированная структура с непрерывным неполным контролем (внутренними средствами кластера), заданным η, и периодическим – внешним полным контролем работоспособности узлов, заданным θ, причем отказ узла с вероятностью η обнаруживается мгновенно, а с вероятностью 1-η обнаружение отказа задерживается на время $1/\theta$ (в среднем). Время задержки обнаружения скрытых отказов имеет экспоненциальное распределение с параметром θ.

Так же важным вопросом надежности является прозрачная отказоустойчивость.

Под прозрачной отказоустойчивостью (Transparent Fault Tolerance, TFT) сервера обычно понимается такое его поведение при возникновении аппаратных или программных сбоев, либо сбоев в сети, при котором:

- сбой не вызывает потери или искажения данных, находящихся в базе данных сервера;
- сервер продолжает нормально функционировать несмотря на имевший место сбой или сбои;
- клиенты сервера «не замечают» произошедших сбоев. Единственным допустимым отклонением от нормального поведения с точки зрения клиента является, возможно, увеличенное время обслуживания (например, на несколько секунд).

Для преодоления последствий отказов была разработана технология, которая опирается на механизм контрольных точек (snapshots. В соответствии с этой техно-

логией в системе должна присутствовать стабильная память, для которой гарантируется, что состояние памяти не меняется при отказах. Соответствующие программные средства периодически сохраняют информацию о состоянии процессов приложения в стабильной памяти. В случае отказа записанная информация используется для того, чтобы повторить вычисления с момента, когда была записана эта информация, то есть выполнить откат назад по времени. Минимальные данные, сохранение которых позволяет выполнить такой откат, называются контрольной точкой или снимком. В качестве стабильной памяти может использоваться дисковая память, энергонезависимая оперативная память, память другого узла или узлов кластера (в последнем случае узел, которому требуется сохранить информацию, пересылает ее через быстрый канал связи на другой узел). Используется также комбинация нескольких типов памяти. Стабильная память после отказа одного из узлов должна продолжать быть доступной тому узлу, на котором делается повтор. К сожалению, механизм контрольных точек не может быть непосредственно применен для обеспечения прозрачной отказоустойчивости из-за недетерминированного поведения сервера приложений. Для того чтобы все-таки добиться прозрачной отказоустойчивости в этом случае, можно применить один из двух методов, описанию и сравнению которых посвящена настоящая работа. Все примеры, которые далее рассматриваются, относятся к операционной системе Solaris 10.

Методы достижения прозрачной отказоустойчивости

Мы будем считать, что аппаратная платформа сервера приложений представляет собой многоузловой кластер, в котором все узлы кроме одного являются основными, а один – резервным. Каждый из узлов имеет свою файловую систему, но эти файловые системы используют для размещения данных общие (разделяемые) тома дисковой памяти. На основных узлах происходит оригинальное выполнение приложения. В случае отказа одного из основных узлов резервный узел берет на себя функции отказавшего узла.

Метод shapshot/restore

Этот метод опирается на механизм контрольных точек. При методе snapshot/restore основной узел периодически фиксирует состояние приложения в стабильной памяти, то есть изготавливает его снимок. Одновременно с этим изготавливается снимок (snapshot) файловой системы (описание снимков файловой системы и ее клонов см., например, в [8]). Для хранения снимков используются общие тома дисковой памяти. После изготовления снимков узел продолжает обычную работу, но с клоном файловой системы, сделанным на основании последнего ее снимка. Между двумя последовательными снимками основной узел ведет истории всех ресурсов, влияющих на детерминированность поведения. История отдельного ресурса – это журнал с именем ресурса, размещенный в стабильной памяти. В качестве стабильной памяти можно использовать оперативную память резервного узла кластера. После отказа основного узла на резервном узле делается восстановление состояния (restore), то есть восстанавливается последнее зафиксированное состояние приложения, операционная среда приводится в состояние, которое соответствует моменту изготовления снимка, а файловая система начинает работу с последнего ее снимка. Далее на резервном узле осуществляется повторное выполнение действий, которые выполнил основной узел от момента изготовления последнего снимка до отказа и которые зафиксированы в журналах ресурсов. Это повторное выполнение мы далее называем управляемым выполнением.Создание снимков состояния приложения и восстановление на основании снимка аналогичного состояния на резервном узле представляет собой сложную задачу. В частности, необходима виртуализация операционной среды, в которой работает приложение. В отсутствие виртуализации приложение при создании некоторого ресурса получает от среды, в которой оно работает, идентификатор, далее называемый системным. Этот идентификатор среда сама присваивает ресурсу, открываемому приложением. Далее приложение при об-

ращениях к ресурсу, им созданному, ссылается на ранее полученный системный идентификатор. При переносе приложения в другой экземпляр среды (среду резервного узла) тот же ресурс может получить в новой среде другой системный идентификатор. При наличии же виртуализации операционной среды для сохранения работоспособности приложения после переноса поступают следующим образом. С каждым подобным ресурсом связывается виртуальный идентификатор (псевдоним), который получает приложение, однако операционная среда по-прежнему продолжает работать с системным идентификатором, который она назначила. Поэтому при восстановлении работы приложения в другом экземпляре среды необходимо, изменив системные идентификаторы, сохранить соответствующие им псевдонимы.

Примерами ресурсов, идентификаторы которых подлежат виртуализации, являются процессы и наборы SVR4-семафоров.

Отметим, что, вообще говоря, для некоторых классов ресурсов обязательно требуется виртуализация, но можно обойтись без протоколирования, так как работа с ними не сказывается на детерминированности проведения приложения. Примером служит разделяемая память (shared memory object). И, наоборот, для некоторых классов (например, классов бинарных семафоров POSIX-mutex's и POSIX-rwlock's) не требуется виртуализация, но обязательно требуется протоколирование.

Недостатки метода snapshot/restore

Основной недостаток этого метода связан с тем, что он опирается на механизм контрольных точек. Пока отсутствуют доступные реализации, которые обеспечивали бы возможность использования в приложениях всего набора средств, предусмотренных стандартами POSIX . Это касается, например, наиболее полной системы BLCR, разработанной Национальной Лабораторией в Беркли . Эта система ориентирована на вычислительные приложения, использующие для обмена данными и промежуточными результатами специальный протокол, названный MPI (Message-Passing Interface). Поэтому она не поддерживает приложения, которые работают с TCP/UDP-сокетами, асинхронным вводом-выводом, средствами System V IPC. Но средства, редко используемые в научных приложениях (например, TCP/UDP-сокеты) могут широко использоваться в серверах приложений, которые нас интересуют. Поэтому требуется доработать систему BLCR, пополнив спектр поддерживаемых средств. Это вполне возможно, поскольку BLCR имеет открытый код.

Второй недостаток есть следствие первого: при изготовлении снимка сервер приложений на некоторое время «замирает», то есть перестает обрабатывать поступающие запросы пользователей. Конечно, запросы не теряются, просто их обработка на время откладывается. Это время имеет порядок десятых долей секунды или даже секунд.

При методе snapshot/restore увеличение размеров оперативной памяти, в которой резервный узел хранит истории ресурсов, приводит к увеличению пропускной способности основного узла, поскольку это дает возможность снизить частоту изготовления снимков.

Заключение

1. Моделирование надежности кластерных структур марковскими процессами позволяет отразить в макромодели изменения процесса отказа элементов во времени, а также временные условия осуществления других событий (профилактические и ремонтные работы);

2. Механическое увеличение глубины дублирования за счет аппаратных средств не приводит к линейному росту надежности, а в некоторых случаях приводит к ее снижению;

3. Динамическое моделирование кластерных структур марковскими процессами не позволяет учитывать наиболее опасные, нестационарные отказы, а именно

сбои вычислительной процедуры по причине самогенерации случайных помех в межсхемных соединениях оборудования модулей;

4. Параметры максимальной помехи монтажа (амплитуда, длительность, момент появления) являются кодозависимыми, а, следовательно, носят случайный характер и имеют только верхнюю, расчетную оценку.

Несмотря на привлекательность метода lock-step, связанную с его большей простотой, отдать ему предпочтение мы не может из-за отсутствия статистических данных. С точки зрения надежности эти методы являются равнозначными. Однако мы предполагаем, основываясь на предварительных оценках, что этот метод вызовет значительно большее снижение пропускной способности, чем более сложный метод snapshot/restore.

Список литературы

1. Беляев Ю.К., Богатырев В.А., Болотин В.В. и др. / под ред. Ушакова И.А. Надежность технических систем: справочник. – М.: Изд-во «Радио и связь», 1985.

2. Бурцев В.С., Характеристики надежности многопроцессорных комплексов и анализ надежности МВК «Эльбрус-2», ИВВС РАН, 1998.

3. Шпаковский Г.И., Верхотуров А.Е., Серикова Н.В. Руководство по работе на вычислительном кластере: учеб. пособие. – М.: БГУ, 2004.

METHOD OF ESTIMATION OF RELIABILITY OF CLUSTER CALCULABLE STRUCTURES AND OTKAZOUSTOYCHIVOSTI OF APPENDIXES WITH NONDETERMINISTIC CONDUCT

V.A. VOLKOV [1]
S.M.CHUDINOV [2]

[1] *NII of VK the name of M.A. Karceva*

[2] *SuperCOMPUTER Research Institute*

Models are resulted and got correlation for the estimation of reliability of the cluster computer systems. Importance of account of reliability of switch of reserve is shown at the design of the reserved structures. The article of consideration are the reserved computer systems of type «El'brus».Oaeaea the questions of providing of transparent otkazoustoychivosti are considered.

Key words: Cluster systems, high readiness, otkazoustoychivost' (OU), project of cluster structures, markovskie processes, coefficient of readiness (KG), high level of readiness (VUG). transparent otkazoustoychivost'yu (Transparent Fault Tolerance, TFT), mechanism of control points (snapshots), by the guided implementation.