

МЕТОДЫ ДОСТИЖЕНИЯ ВЫСОКОЙ ОТКАЗОУСТОЙЧИВОСТИ ПРОГРАММНЫХ ПРОДУКТОВ ПРОМЕЖУТОЧНОГО УРОВНЯ

Е.С. Омелюсик, Д.А. Хлебест,
Научный руководитель – Ролич О.Ч.
канд.техн.наук, доцент

**Белорусский государственный университет
информатики и радиоэлектроники**

Широкое распространение получило мнение, что для достижения высокой отказоустойчивости существует уже готовое технологическое решение, единожды применив которое, можно достичь желаемого результата. Однако подобная технология – это достаточно небольшая часть мер обеспечения высокой отказоустойчивости.

В действительности высокая отказоустойчивость достигается комплексным воздействием специалистов, подходов и технологий. Большинство специалистов определяют отказоустойчивость как:

$$A = \frac{F-(D+R)}{F},$$

где A – отказоустойчивость,

F – время между сбоями,

D – время, необходимое для выявления сбоя и выбора контрмер,

R – время восстановления после сбоя.

Единственный фактор, не поддающийся контролю, - это промежуток времени между сбоями, так как невозможно предсказать время сбоя системы.

Основной упор в достижении высокой отказоустойчивости делается на инвестирование ресурсов в персонал и методы, сокращающие время выявления и принятия решения, и инвестирование в технологии, сокращающие время выявления и восстановления.

Наивысший процент отказоустойчивости можно достичь с помощью трех методов: кластеризации, перемещения журналов и репликации. Хотя строить решения высокой отказоустойчивости можно и без одного из этих компонентов, внимание ко всем трем позволяет обеспечить полномасштабной отказоустойчивостью всю информационную систему предприятия целиком. Также к аспектам отказоустойчивости относят и быстроту восстановления после сбоя.

Повышенная готовность является стратегией управления системой, при которой осуществляется быстрое восстановление основных служб в случае аварий системы, компонента или приложения. Целью является минимальное время прерывания службы, то есть сохранение работоспособности при отказе [1].

Кластер – слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений, и представляющихся пользователю единой системой. Кластер - это защитный механизм только на уровне аппаратного обеспечения. Кластеризация не защищает от сбоев на уровне данных, сбоев приложений логического характера и ошибок пользователей. При таком подходе запросы распределяются по всему кластеру [2].

Перемещение журналов предоставляет дополнительную гибкую возможность для защиты против аппаратных сбоев. При перемещении журналов файлы первичного сервера копируются на вторичный, затем файлы восстанавливаются на вторичном сервере. Перемещение журналов как технология спроектировано таким образом, что дает первичному и вторичному серверам независимость от компонентов аппаратного обеспечения.

Репликация обеспечивает отказоустойчивость по транзакциям. В отличие от перемещения журналов, восстанавливающего резервные копии журналов транзакций с первичного сервера на вторичном сервере, репликация работает с журналом транзакций и посылает индивидуальные транзакции на вторичный сервер по мере их поступления. Благодаря этой детализирующей функции репликация в состоянии обеспечить максимально быстрое переключение с первичного сервера на вторичный.

На основании всего вышесказанного, можно сделать вывод, что для достижения высокой отказоустойчивости не существует идеального решения. Каждый вариант повышения отказоустойчивости способен обеспечить дополнительную защиту данных, а использование комплексного подхода способно обеспечить бесперебойную работу. Основной задачей специалиста, желающего повысить отказоустойчивость программного продукта промежуточного уровня является понимание того, как, комбинируя различные технологии, преодолеть недостатки каждой из них.

Библиографический список

1. Программное обеспечение повышенной готовности промежуточного уровня в Linux [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=2620> – Дата доступа: 06.10.2019

2. Бернс, Б. Распределенные системы. Паттерны проектирования/ Б. Бернс. – СПб. : Питер, 2019. – 224 с.