Лекции по Отказоустойчивым ВС, Майданов Ю.С.

ИВ-621 Антипова Екатерина

СОДЕРЖАНИЕ:

[СОДЕРЖАНИЕ: 1](#_Toc40623050)

[1. Лекция №1 2](#_Toc40623051)

[1.1. Общие понятия ВС 2](#_Toc40623052)

[1.2. Допущение к Отказоустойчивым ВС 3](#_Toc40623053)

[1.3. Меры диагностики 3](#_Toc40623054)

[2. Лекция №2 5](#_Toc40623055)

[2.1. Термины ОУВС 5](#_Toc40623056)

[2.2. Поиск одиночных отказов в ВС 5](#_Toc40623057)

[2.2.1. Алгоритм поиска 1: парные решения с исключением пар 5](#_Toc40623058)

[2.2.2. Алгоритм поиска 2: парное решение с пост проверкой 6](#_Toc40623059)

[2.3. Графовые модели ОуВС 7](#_Toc40623060)

[3. Лекция №3 Диагностические модели 9](#_Toc40623061)

[4. Лекция №4 11](#_Toc40623062)

[4.1. Алгоритмы построения диаграммы загрузки П1 11](#_Toc40623063)

[4.2. Алгоритм построения диаграммы загрузки П2 12](#_Toc40623064)

[4.3. Алгоритм построения диаграммы загрузки П3 13](#_Toc40623065)

1. Лекция №1
   1. Общие понятия ВС

Вычислительная система называется отказоустойчивой, если она способна продолжать функционирование даже при наличии отказавших ресурсов в реальной ВС, являющийся ее носителем.

ОВС может находиться в 3-х стадиях:

1. Работоспособное отказоустойчивое состояние
2. Не отказоустойчивое состояние
3. Неработоспособна

Для обеспечения отказоустойчивости требуется избыточность.

Избыточность бывает:

1. Пассивной

Избыточные ресурсы используются только при отказе.

1. Активной

Избыточные ресурсы используются в вычислительной системе.

Этапы

1. Вычислительный процесс (решаются прикладные задачи)
2. Контроль (выявляется факт наличия неисправности)
3. Диагностика (все ресурсы делятся на исправные и неисправные)

Это самый сложный этап.

1. Реконфигурация (из исправных ресурсов формируется новая ВС)
2. Восстановление вычислительного процесса.

Отказоустойчивые вычислительные системы, типы:

1. Централизованные

Все процедуры определяются состояния и сдачи возлагаются на некий центральный узел, отказоустойчивость обеспечивается внешними средствами.

1. Децентрализованные (более сложные)

Коллектив вычислителей самостоятельно выявляет неисправные ресурсы и обеспечивает продолжение вычислительного процесса.

* 1. Допущение к Отказоустойчивым ВС

1. Рассматриваются только установленные отказы, сбои не рассматриваюстя.
2. Исправная вычислительная машина однозначно верно определяет состояние ср-й вс
3. Рассматриваются только отказы вычислительной системы, отказы связи не рассматриваются
   1. Меры диагностики

Количество элементарных машин, неисправность которых может быть выявлена самой вс, без привлечения внешних средств, при этом большое количество машин не удастся выявить

Мера диагностики определяется

1. количеством элементарных машин
2. структурой связей между ними
3. характером неисправности.

Определяется индивидуально для каждой ВС

1. Количество элементарных ВС (n)
2. Количество связей между элементами (m)
3. Мера диагностики

Режимы работы ВС

Отказ системы – это ее невозможность решать поставленные прикладные задачи.

1. Отказоустойчивый

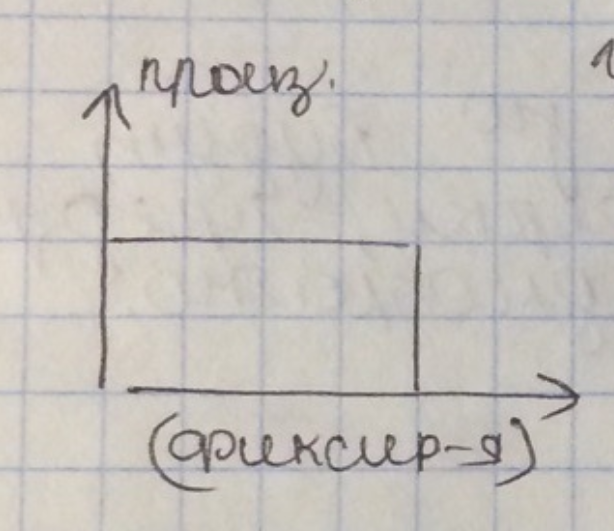


Рисунок 1 График производительности

1. Живучий

Живучая ВС способна использовать все ресурсы для обеспечения вычислительного процесса.

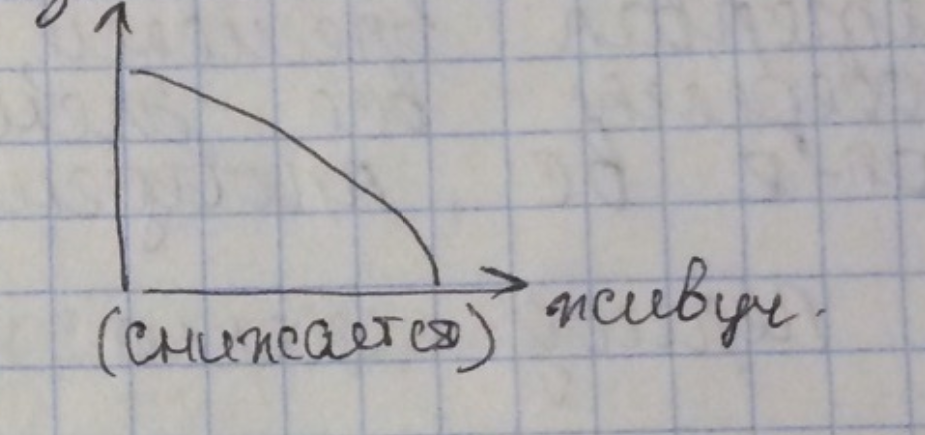


Рисунок 2 График производительности

1. Лекция №2
   1. Термины ОУВС
2. Элементарная проверка

На 2-х элементарных машинах в составе ВС распределяется одна и та же задача с последующими сравнениями результатов, при этом каждая машина делает вывод о состоянии парной.

1. Синдром ВС

Это совокупность резервов всех элементарных проведенных

1. Дешифрация синдрома

Это обработка синдрома в результате которой элементарной машине ставится в соответствие состояние: исправна/неисправна.

1. Период цикла диагностики

Этом интервал времени в течении которого будут проведены все элементарные проверки, предусмотренные ВС, проведены будут однократно.

1. Период цикла решения

Это интервал времени за который однократно будут решены все фрагменты прикладных задач, предусмотренные на 1-й итерации.

* 1. Поиск одиночных отказов в ВС
     1. Алгоритм поиска 1: парные решения с исключением пар

Все элементарные машины в составе ВС разбиваются на пары. Каждой паре назначается фрагмент прикладной задачи. После решения очередного фрагмента производится сверка результатов в паре.

Если результаты не совпадают, то вся пара исключается из вычислительного процесса, а назначенный ими фрагмент пересчитывается другой парой.

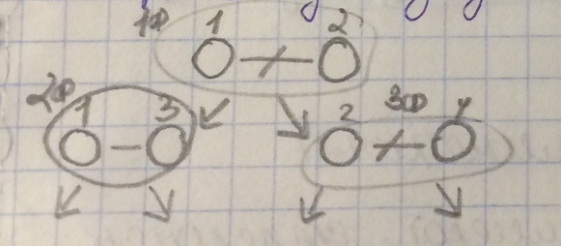
Достоинства:

* Простота реализации
* Высокая скорость работы

Недостатки:

* При любом отказе и иск-ся исправная элементарная машина
* Решаемый фрагмент требует пересчета
* Очень высокая избыточность
  + 1. Алгоритм поиска 2: парное решение с пост проверкой

Элементарные машины так же делятся на пары и так же каждой паре ставится свой что-то там. При несовпадении результатов пара разделяется и образуются 2 новые пары, на которых расчитываются новые фрагменты прикладных задач.



Если расхождение присутствует в 2х новых парах, то образуемый доступно новые 4 пары и т.д. до однозначного выявления всех неисправных машин.

Достоинства:

* Ни один из фрагментов не требует пересчета
* Из вычислительных процессоров исключаются только неисправные машины
* Простота реализации алгоритма

Недостатки:

* Высокая избыточность
* Требуется большое количество независимых фрагментов прикладных задач

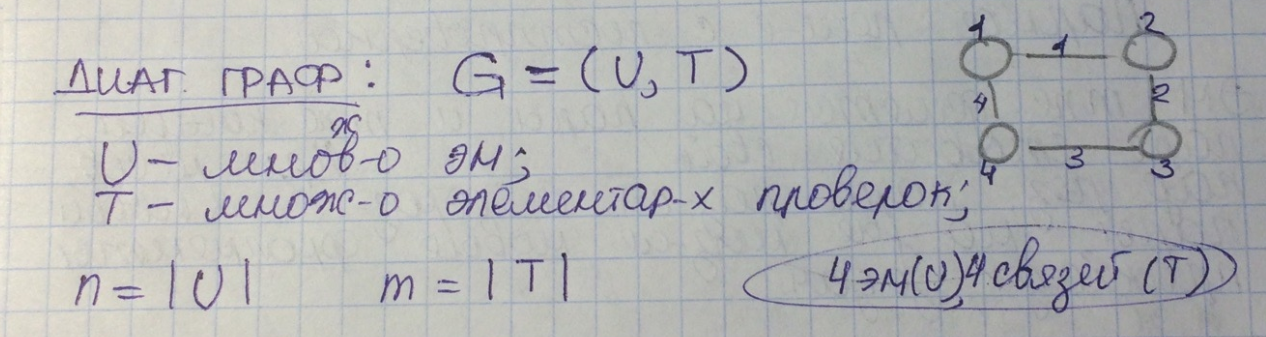
Подходит для систем реального времени.

* 1. Графовые модели ОуВС

Для описания ряда процессов удобно использовать графовые представления.

В ОуВС используется 2 графа:

1. Диагностический

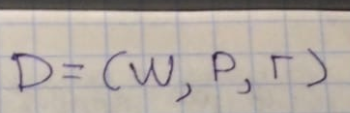


* 1. Симметричный

В симметричном графе связи не имеют стрелочек и это означает: проведение взаимной проверки, обе машины являются проверяемыми и проверяющими.

* 1. Несимметричный

В несимметричном графе четко выделено проверяющий и проверяемый.

1. Информационных связей 

Показывает фрагменты прикладных задач и связей между ними

W – множество фрагментов в прикладных задачах

Р – множество слож-ть фрагментов прикладных задач.

Г – множество информационных связей между фрагментами

Граф информационных связей всегда направленный

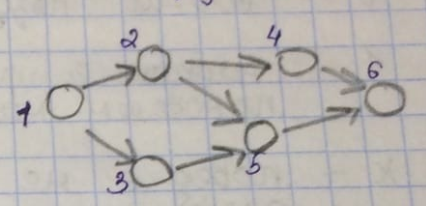


Рисунок 3 зависимость по данным

* 1-ый такт – 1 фрагмент (1)
* 2-ой такт – 2 фрагмента (2, 3)
* 3-ий такт – 3 фрагмента (4, 5, 5)
* 4-ый такт – 2 фрагмента (4, 5)

Граф используется при планировании работы ВС и ОУ в частности

1. Лекция №3 Диагностические модели

Диагностическая модель описывает предположения о результате взаимной проверки в зависимости от состояния ЭМ участвующих в этой проверке.

Диагностическая модель описывает четверкой переменных (F1, F2, F3, F4)

F1 – предположение о результате проверки, если обе элементарные машины исправлены

F2 – предположение о результате элементарной проверки, если проверяющая исправна, а проверяемая неисправна

F3 – предположение о том, что проверяющая машина неисправна, а проверяемая исправна.

F4 – если проверяющая машина не исправна и проверяемая тоже.

Каждая из F может принимать значения 0,1.

0 – совпадение результатов, соответственно исправности элементарной машины

1 – расхождение результатов, соответственно неисправность проверяемой машины

х – поведение не определено (нельзя предсказать оценку)

F1 – всегда 0

F2 – всегда 1

F3, F4 – 0, 1, х.

Таким образом количество диагностических моделей 9 (3^2)

Если ничего неизвестно о ВС, то ее относят к модели (0, 1, x, x)/

В процессе эксперимента или происходив можно сделать предсказание о нове-ие исправной машины.

В этом случая диагностическая модель будет отлична от (0, 1, х, х).

Присваивание дм-ли позволяет использовать ускоренно делифрац синдрома и существенно снизить затраты на \_\_\_\_ диагностику операций. Если F3 = 1, то диагностическая модель называется симметричной.

По факту осуществляется взаимная элементарная проверка и неисправность будет выявлена независимо от того, какая из 2-х машин неисправна.

Если F3 = 0, то диагностическая модель называется несимметричной

Если F3 == x, то модель нельзя отнести ни к симметричным, ни к несимметричным.

Таким образом синхронизация моделей 3.

Дешифрация синдрома для несимметричных моделей быстрее.

Диагностическая модель (0, 1, 0, 0)

Расхождение резервов возможно только в случае, если проверяемая модель исправна, а проверяемая модель неисправна.

Диагностическая модель (0, 1, 0, 1)

Появление единицы говорит о том, что либо проверяемая машина неисправна, а состояние проверяющей неизвестно.

Алгоритм дешифрации тот же.

Диагностическая модель (0, 1, 0, х)

Алгоритм тот же

Диагностическая модель (0, 1, 1, 0)

Появление единицы, возможно только тогда, когда одна из машин неисправна, а вторая исправна

Диагностическая модель (0, 1, 1, 1)

Отнесение ВС к конкретной Диагностической модели – маз-ем существенно сократить время на дешифрацию, но требует более тщательного анализа работы ВС.

1. Лекция №4

Планирование отказоустойчивых вычислений состоит из 3-х этапов

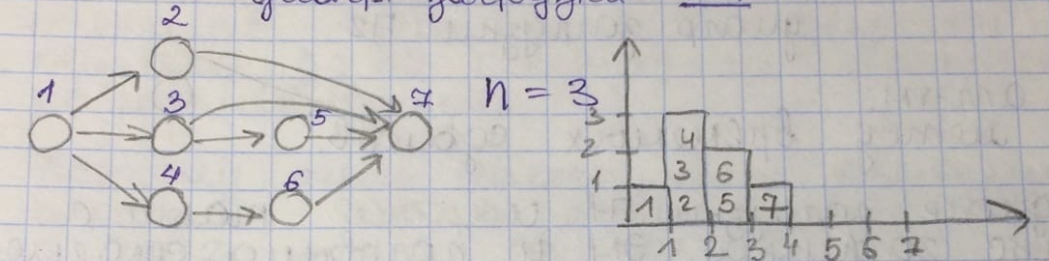
1. Построение диаграммы загрузки П1 целью которой является минимизация времени решения, т.е. диагностические операции не учитываются
2. Построение диаграммы загрузки П2. Цель: минимизация количества элементарных машин, задействованных в решениях прикладной задачи, на каждом этапе решения, при заданном максимальном времени решения.
3. Построение диаграммы загрузки П3

Цель: предусматривает реализацию всех элементарных проверок, предусматривает диагностики графом.

Итоговая диаграмма П3 является планом решения прикладных задач с учетом операции, обеспечения отказоустойчивых вычислений.

Исходными данными являются:

1. для диаграммы П1: граф информации связей и количество элементарных машин
2. для диаграммы П2: диаграмма П1 и максимальное время решений
3. для диаграммы П3: диаграмма П2 и диагностика графа
   1. Алгоритмы построения диаграммы загрузки П1



Все алгоритмы сводятся к выбору альтернатив, когда они ест. Если появляются альтернативы, соответственно

1. Алгоритм: случайный выбор
2. (самый эффективный) Алгоритм: выбирается вершина с максимальным количеством исходных связей, делаем предположение, что в дальнейшем будет больше альтернатив.
3. Алгоритм: выбирается вершина с минимальным удалением от начальных
4. Алгоритм: выбирается вершина имеющая максимальный путь до конечной (он более трудоемкий)
5. комбинация вышеперечисленных алгоритмов с разным приоритетом

Все перечисленные алгоритмы – эвристические, т.к. точный алгоритм требует полного перебора всех вариантов, но на практике не используется.

* 1. Алгоритм построения диаграммы загрузки П2

1. Алгоритм: метод временных сдвигов.

В диаграмме загрузки П1 ищется такт с максимальной загрузкой элементарных машин, и половина фрагментов этого такта переносятся на следующий такт. Далее повторяется ситуация либо до достижения максимально допустимого времени, либо до снижения загруженности элементарных машин ниже половины от всех доступных.

Дальнейшее снижение загрузки приведет к простою ЭМ.

1. Алгоритм: метод подбора количества доступных элементарных машин.

На каждой итерации строится диаграмма загрузки П1, сравнив время решений с максимально допустимым и если время решения меньше максимального, то количество доступных машин снижается и итерация повторяется. Если вышли за пределы допустимого времени, то осуществляется возврат предыдущей итерации и если количество не превышает половины, то алгоритм завершается.

1-ый алгоритм обладает меньшей трудоемкостью, позволяет использовать все доступные временные ресурсы, но итоговая диаграмма загрузки П2 обладает высокой неравномерностью.

2-й алгоритм обладает большей трудоемкостью и не всегда позволяет использовать всю доступную временную избыточность, однако диаграмма загрузок П2 получается с большей равномерностью.

Поэтому очень часто используется комбинирование: на 1-ом этапе используется 2-ой алгоритм, а на 2-ом этапе используется 1-ый алгоритм.

* 1. Алгоритм построения диаграммы загрузки П3

При построении диаграммы загрузки П3 необходимо реализовать все связи диаграммы графа.

