Отказоустойчивость

Мельников В.М., 2011

Содержание

1. Понятие отказоустойчивости, fail-fast модули
2. Отказоустойчивость для аппаратного обеспечения
3. Отказоустойчивость для систем хранения данных
4. Отказоустойчивость для процессов
5. Отказоустойчивость при передачи данных

“Правило самолета”

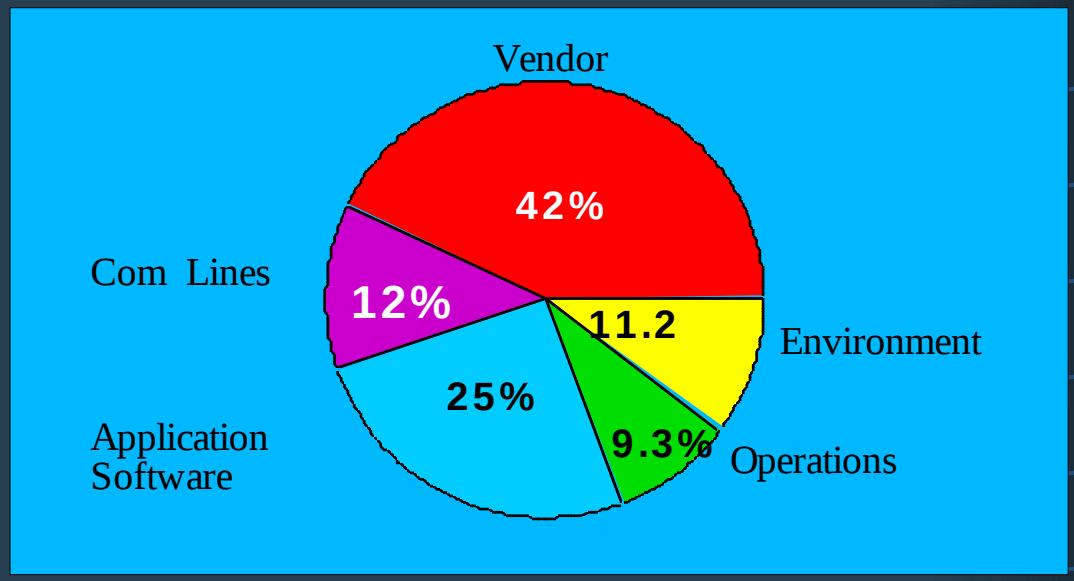
- Самолет с двуми двигателями имеет в два раза больше проблем
- Отказоустойчивость – ключевой компонент
- Маскируем и восстанавливаем ошибки

Надежность – ЭТО ...

- Достоверность (reliability)
- Целостность (integrity)
- Доступность (availability)
- Безопасность (security)

- MTTF / (MTTF + MTTR)

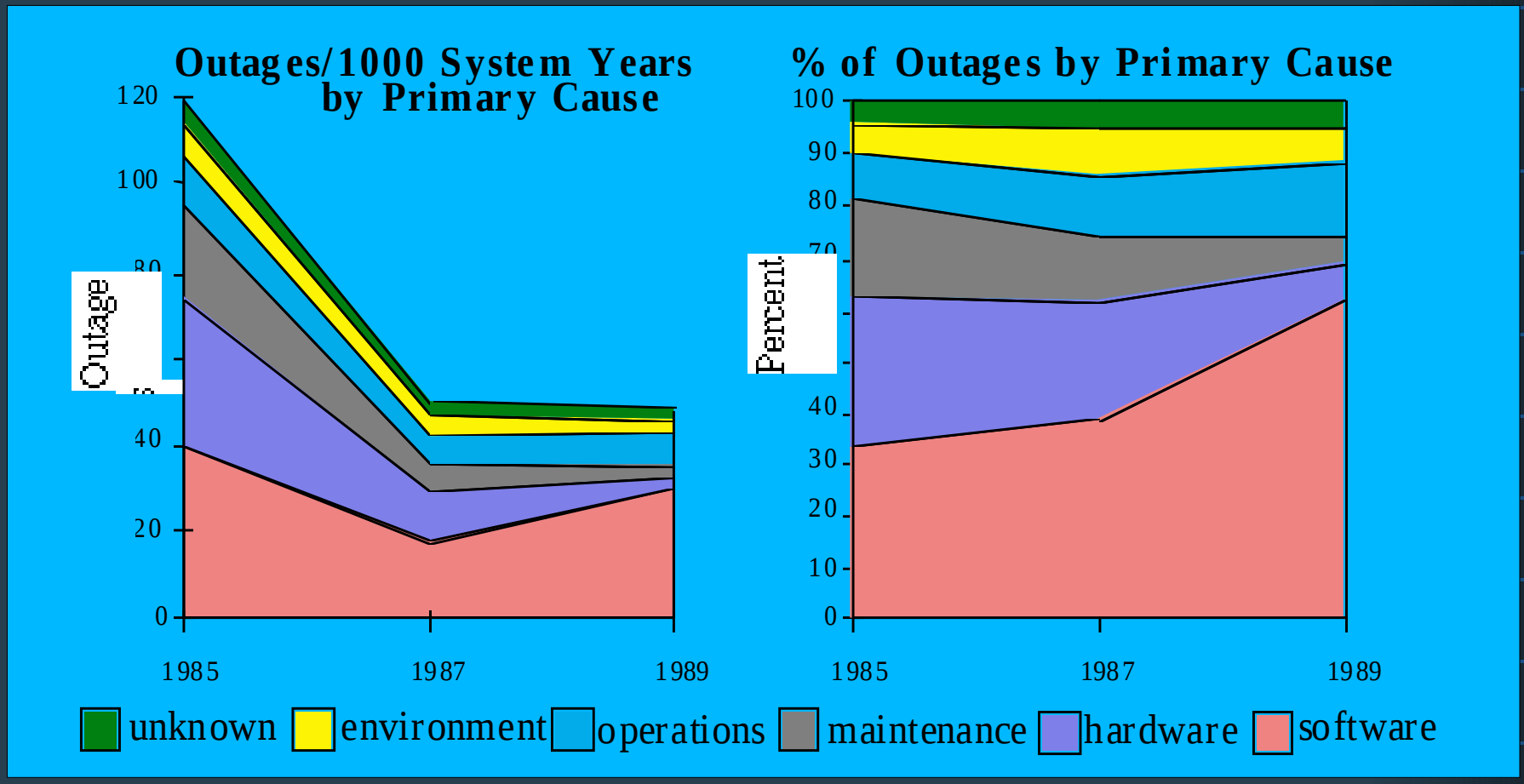
Как часто происходят ошибки



1,383 института опрошены (6/84 - 7/85)

7,517 отказов, MTTF ~ 10 недель, средняя длительность ~ 90 минут

Тенденции



Сдвиг в сторону ПО (62%) и операций(15%)

Ключевые идеи обеспечения отказоустойчивости

{ Архитектура
ПО
Распределение } Маскирует { Ошибки аппаратуры
Ошибки окружения
Ошибки поддержки }

- Программное обеспечение устраняет/автоматизирует выполнение операций
- Все проблемы граничат с ошибками программного обеспечения и дизайна
- Откозоустойчивость программного обеспечения – ключ к НАДЕЖНОЙ СИСТЕМЕ!

Техники обеспечения отказоустойчивости

FAIL FAST MODULES: работай или остановись

SPARE MODULES : короткое время восстановления

INDEPENDENT MODULE FAILS путем дизайна

$MTTF_{Pair} \sim MTTF^2 / MTTR$

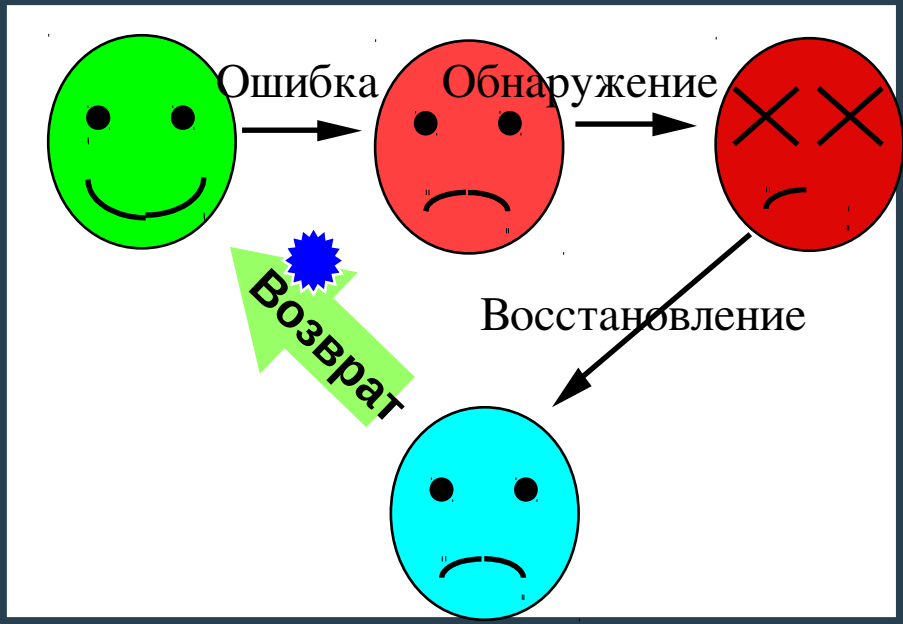
MESSAGE BASED OS: Изоляция ошибок, нет разделяемой памяти

SESSION-ORIENTED COMM: Достоверные сообщения (обнаружение пропавших и дублирующихся сообщений)

PROCESS PAIRS : Маскирование проблем аппартного и программного обеспечения

TRANSACTIONS: предоставление A.C.I.D. свойств (простая модель обработки ошибок)

Fail-fast модули

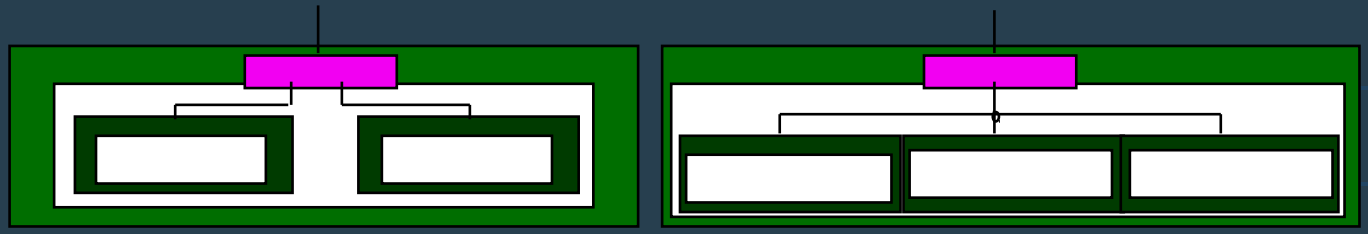


- Короткое время обнаружения ошибки
- Основная задача состоит в улучшении MTTF и MTTR

Аппаратное обеспечение. Мультиплексирование

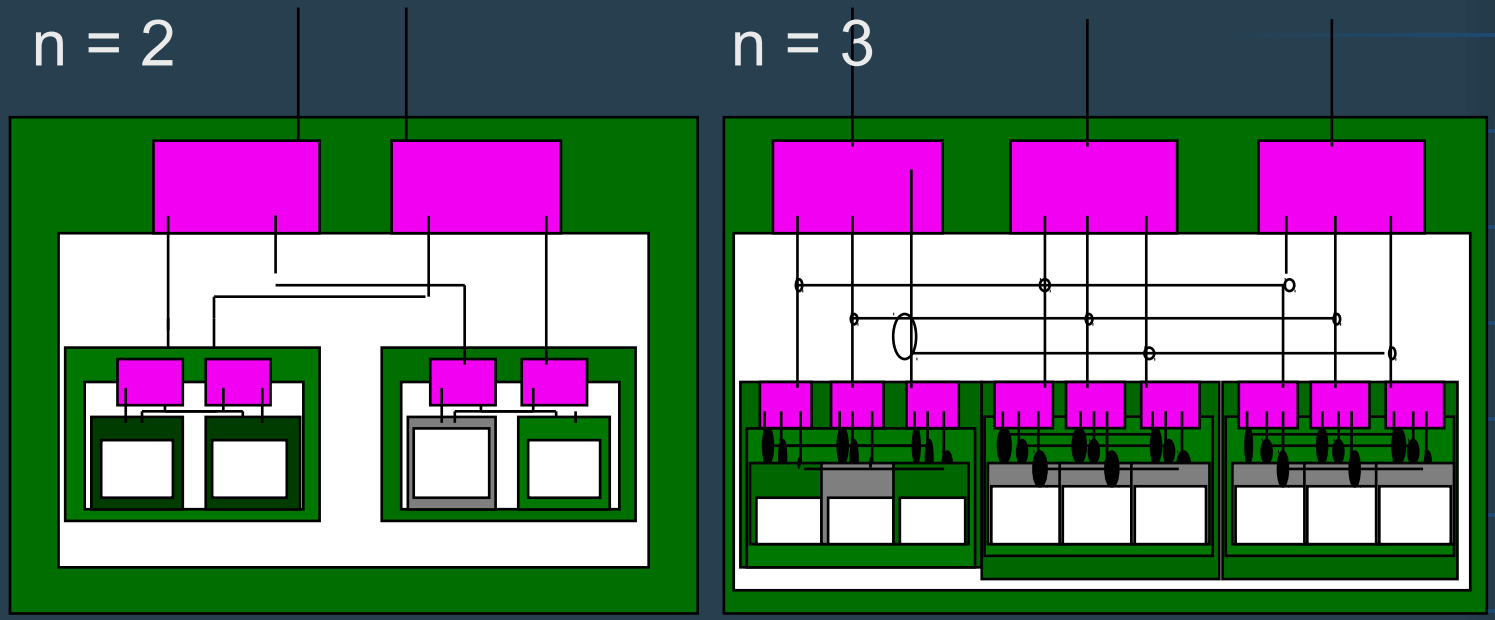
Пара

Триплекс



- Стратегии голосования
  - Fail-fast – ошибка, если не работает любой
  - Fail-soft – ошибка, если не работают все

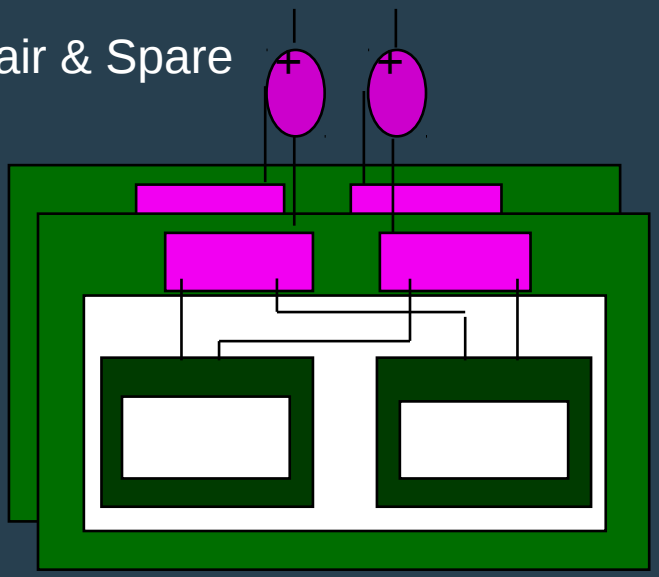
Аппаратное обеспечение. Рекурсивное построение



Аппаратное обеспечение. Запасная пара

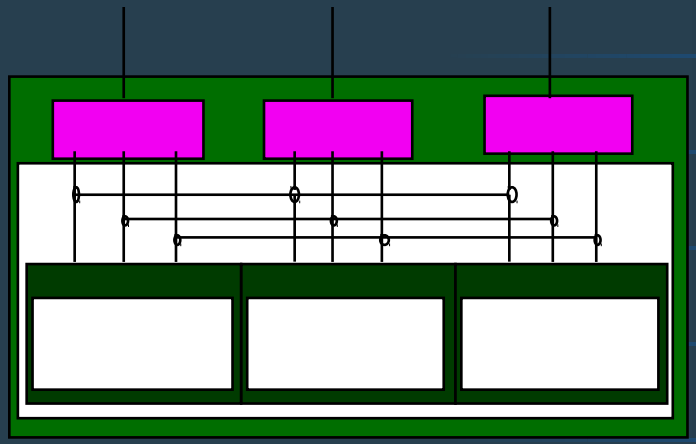
n = 2

Pair & Spare



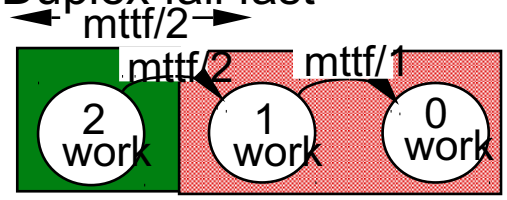
n = 3

Triple Modular Redundancy

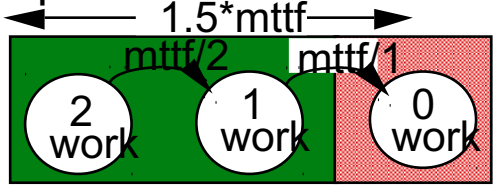


Доступность модуля без восстановления

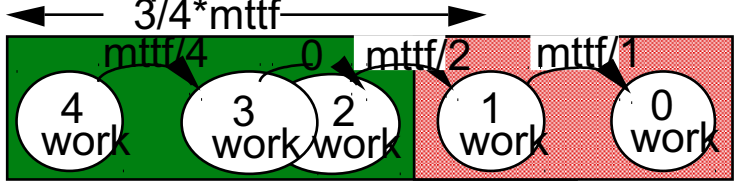
Duplex fail fast



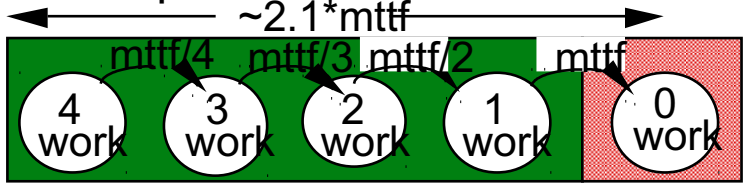
Duplex: fail soft



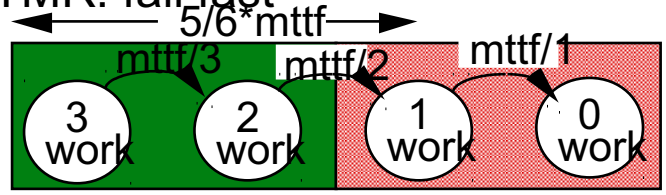
Pair & Spare: fail fast



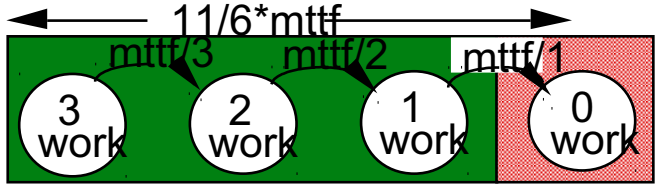
Pair & Spare: fail soft



TMR: fail fast

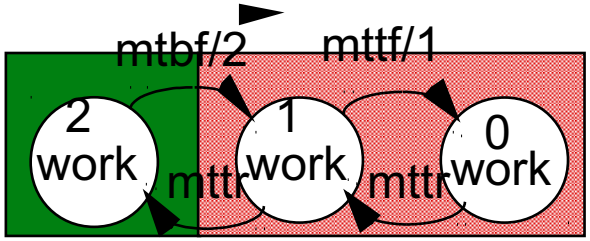


TMR: fail soft

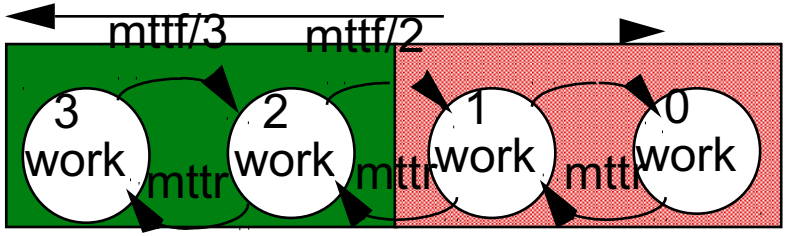


Доступность модуля с восстановлением

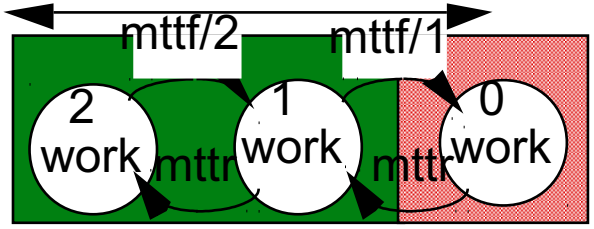
Duplex: fail fast: **mttf/2**



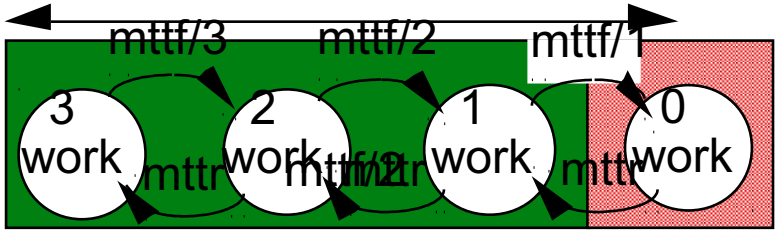
TMR: fail fast **10^4 mttf**



Duplex: fail soft **10^4 mttf**



TMR: fail soft **10^5 mttf**



Доступность модуля с восстановлением

Availability estimates 1 year MTTF modules 12-hour MTTR			MEQUATION MTTF
SIMPLEX	1 year	MTTF	1
DUPLEX: FAIL FAST	~0.5 years	≈MTTF/2	2+ε
DUPLEX: FAIL SOFT	~1.5 years	≈MTTF(3/2)	2+ε
TRIPLEX: FAIL FAST	.8 year	≈MTTF(5/6)	3+ε
TRIPLEX: FAIL SOFT	1.8 year	≈1.8MTTF	3+ε
Pair and spare: FAIL-FAST	~.7 year	≈MTTF(3/4)	4+ε
TRIPLEX WITH REPAIR	$>10^5$	$MTTF^3/3MTTR^2$	3+ε

Серверные решения

- Дублирование питания
- Дублирование сетевых карт
- Дублирование на уровне модулей

Отказоустойчивость для систем хранении данных

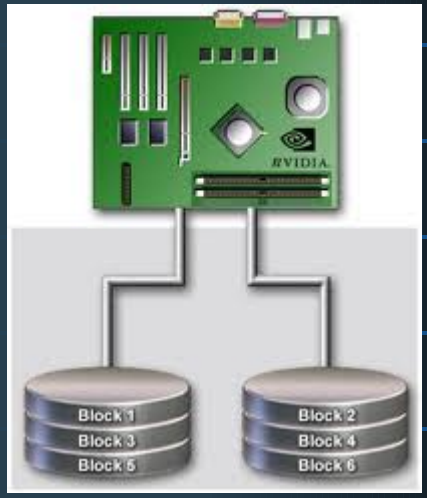
- Операции READ/WRITE
- Параметры
  - Адрес блока (цилиндр, поверхность, дорожка, сектор)
  - Содержимое (набор байт)
- Ошибки
  - Нарушение целостности
  - Ошибка адресации
  - Физическое повреждение

Необходимая избыточность

- Для данных
  - Контрольная сумма
  - Адрес вместе с данными
  - Копия данных
- В логике
  - Скрытое контрольное чтение после записи
  - Скрытое восстановление во время чтения

RAID

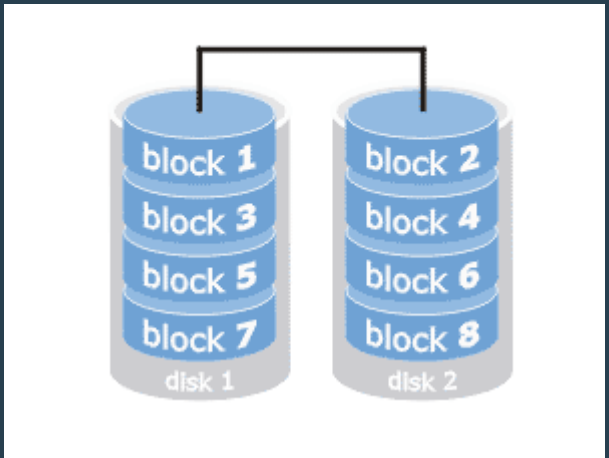
Redundancy
Array of
Inexpensive/Independent
Disks



- RAID 0 – нет дублирования, параллельные операции
- RAID 1 – зеркалирование данных
- RAID 2 – избыточный код для восстановления
- RAID 3,4 – контроль четности
- RAID 5 – чередующийся блок четности
- RAID 6 – двойной блок четности
- RAID 01, 10, 03, 30, 53 – комбинированные схемы

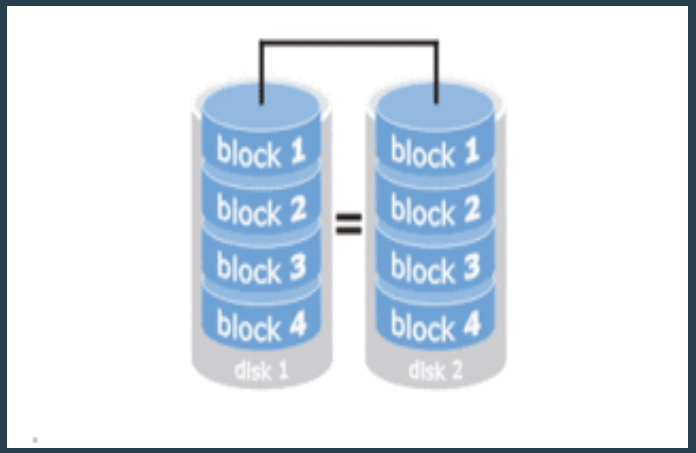
RAID 0

- Дисковый массив с чередованием без отказоустойчивости/чётности



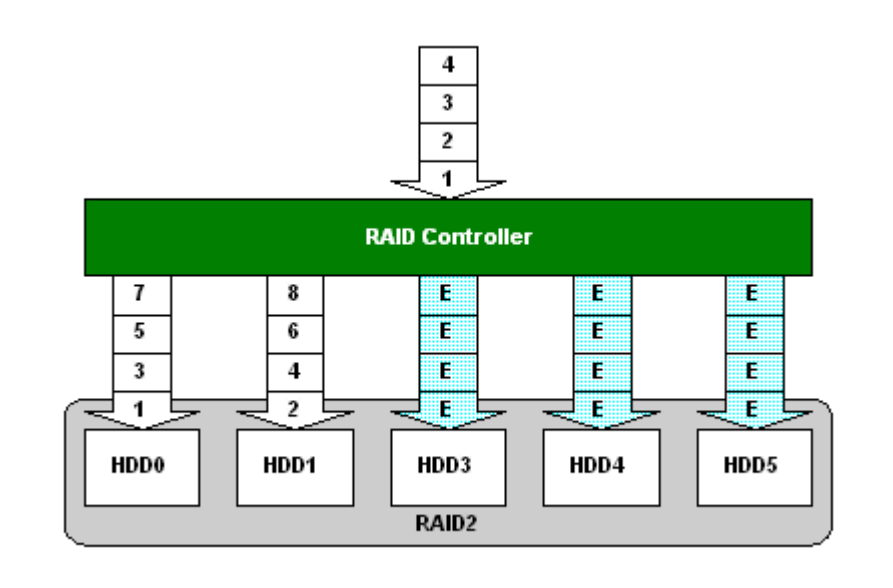
RAID 1

- Зеркалирование



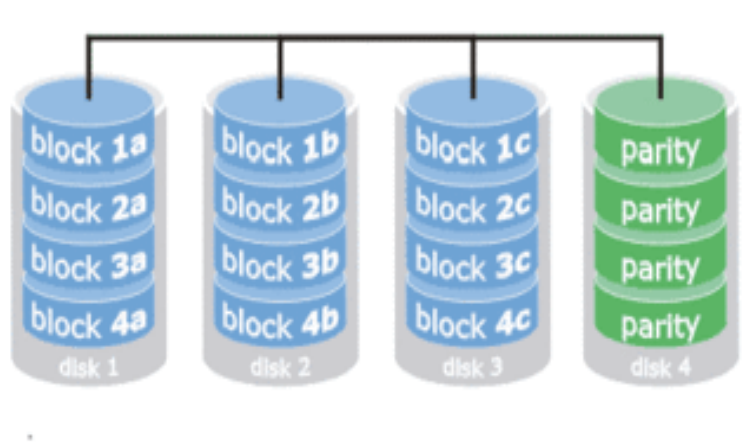
RAID 2

- Массив с использованием помехоустойчивого кода Хемминга (ECC память)



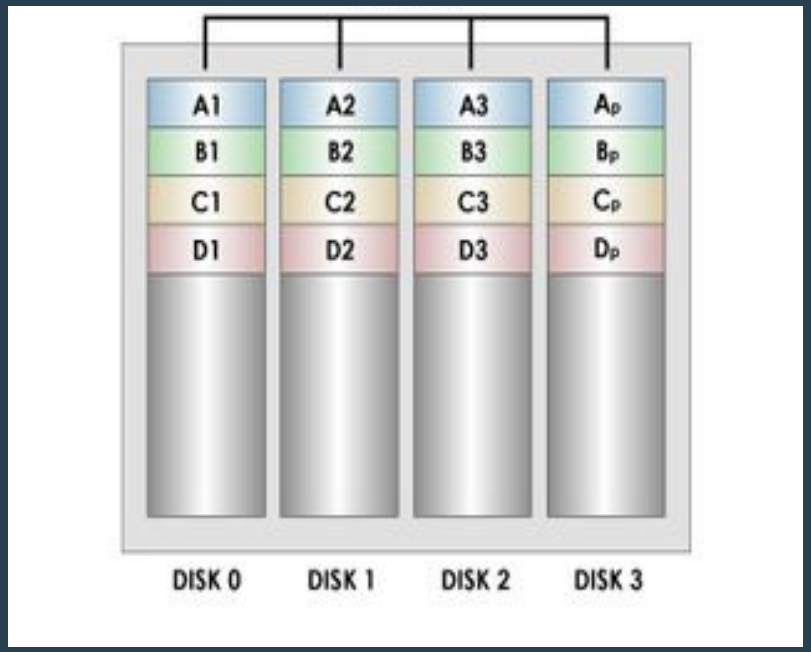
RAID 3

- Отказоустойчивый массив с битовым чередованием и чётностью



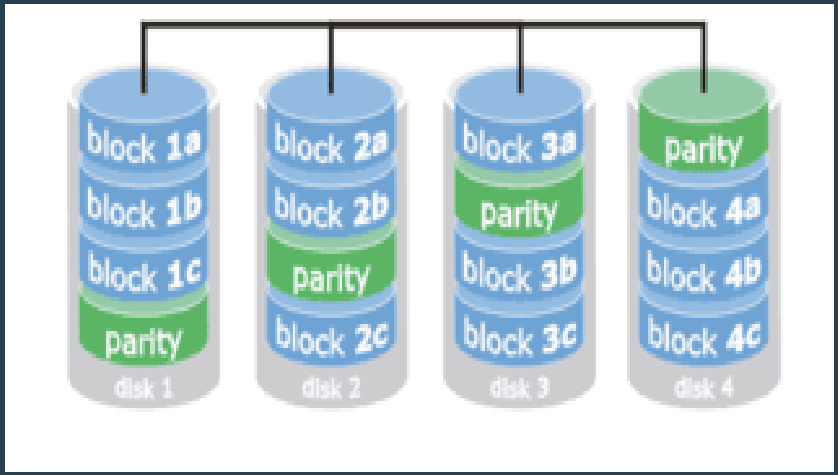
RAID 4

- Отказоустойчивый массив с блочным чередованием и чётностью



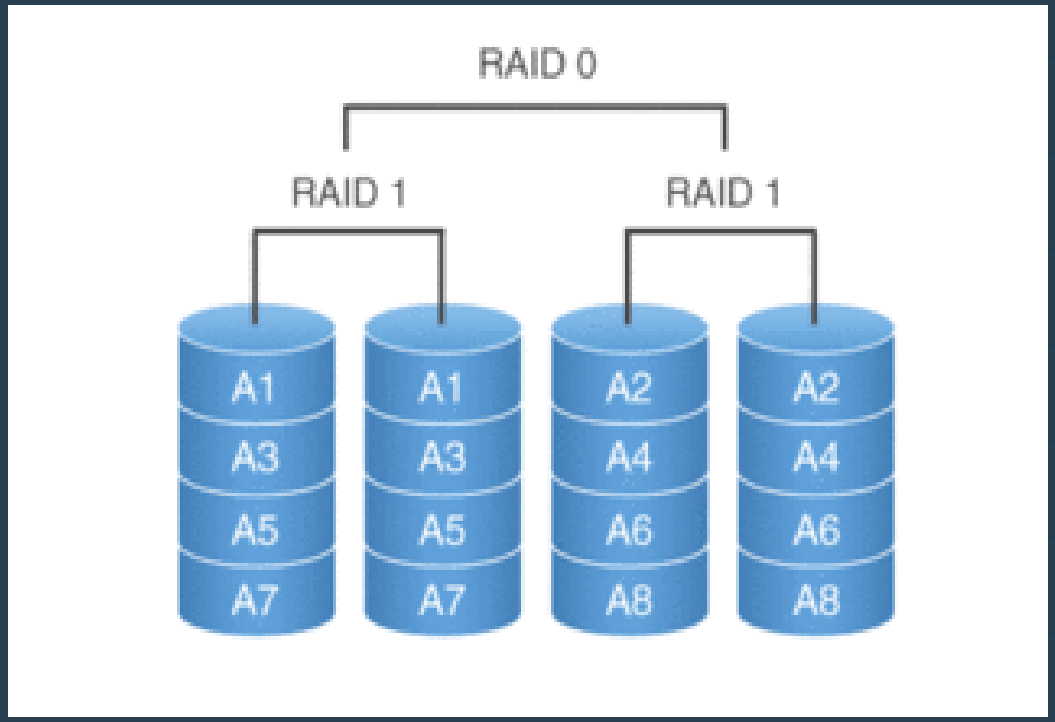
RAID 5

- Дисковый массив с чередованием и распределённой чётностью



RAID 10

- Кобинированная схема



Отказоустойчивость программного обеспечения

- Обязательно хорошее и корректное проектирование, кодирование и тестирование
- Техники обеспечения отказоустойчивости:
  - Модульность (изоляция, ограничение распространения ошибок)
  - Разнородный дизайн
  - Программирование N-версий (разная реализация)
  - Защитное программирование (проверка параметров и данных)
  - Аудитор: Проверка структур данных в фоновом режиме
  - Транзакции (восстановление состояния после ошибки)
- Необходимо fail-fast программное обеспечение

Отказоустойчивость программного обеспечения

- Процессы с хранением состояния
- Пара процессов

Процесс с сохранением состояния

- Состояние процесса:
  - Локальные и глобальные переменные
  - Содержимое стека
  - Регистры
  - Текущая выполняемая команда
- Постоянная запись состояния во внешную память
- Считывания состояния при перезапуске
- Перезапуск при сбое

Процесс с сохранением состояния

- Печать лотерийных билетов

```
if (restart)
        read ( f, i);
else
        i = 1;
while ( i < =N ) {
        write ( f, i );
        PRINT_TICKET( i );
        i = i + 1;
}
```

Процесс с сохранением состояния

- Печать лотерийных билетов

```
if (restart)
        read ( f, i);
else
        i =  0;
while ( i < N ) {
        i = i + 1;
        write ( f, i );
        PRINT_TICKET( i );
}
```

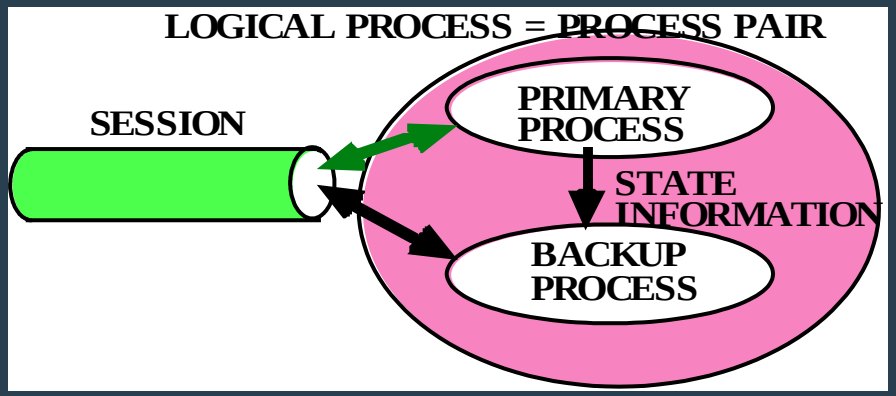
Процесс с сохранением состояния

- Достоинства
  - Простая реализация
- Недостатки
  - Медленная работа
  - Проблема синхронизации логики приложения с последним состоянием

Пара процессов

ОСНОВНОЙ + ЗАПАСНОЙ = ЛОГИЧЕСКИЙ ПРОЦЕСС

- Как обнаружить сбой?
- Как продолжить работу?

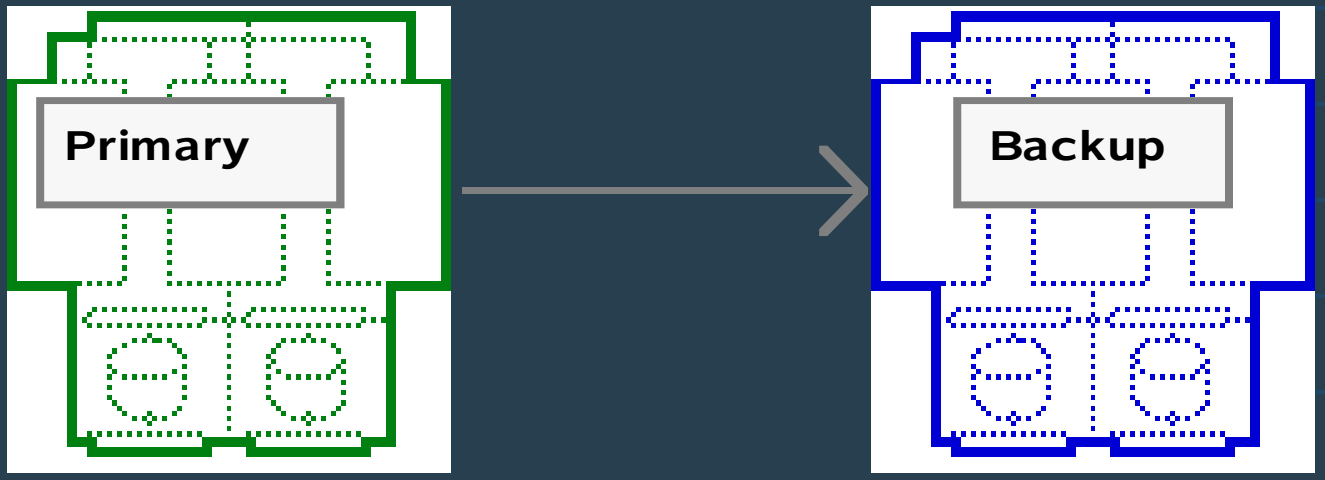


Блок схема работы процессов

Пара процессов

- Достоинства
  - Высокая скорость переключения
  - Возможно масшабирование
- Недостатки
  - Высокая нагрузка на сеть

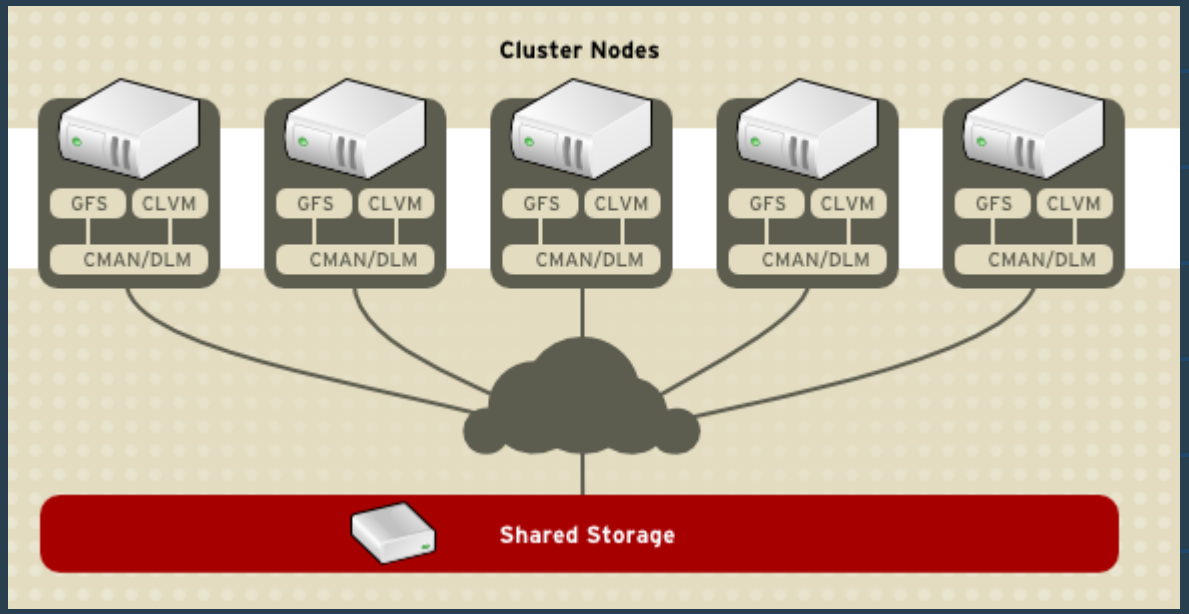
Отказоусточивость систем. Общая концепция



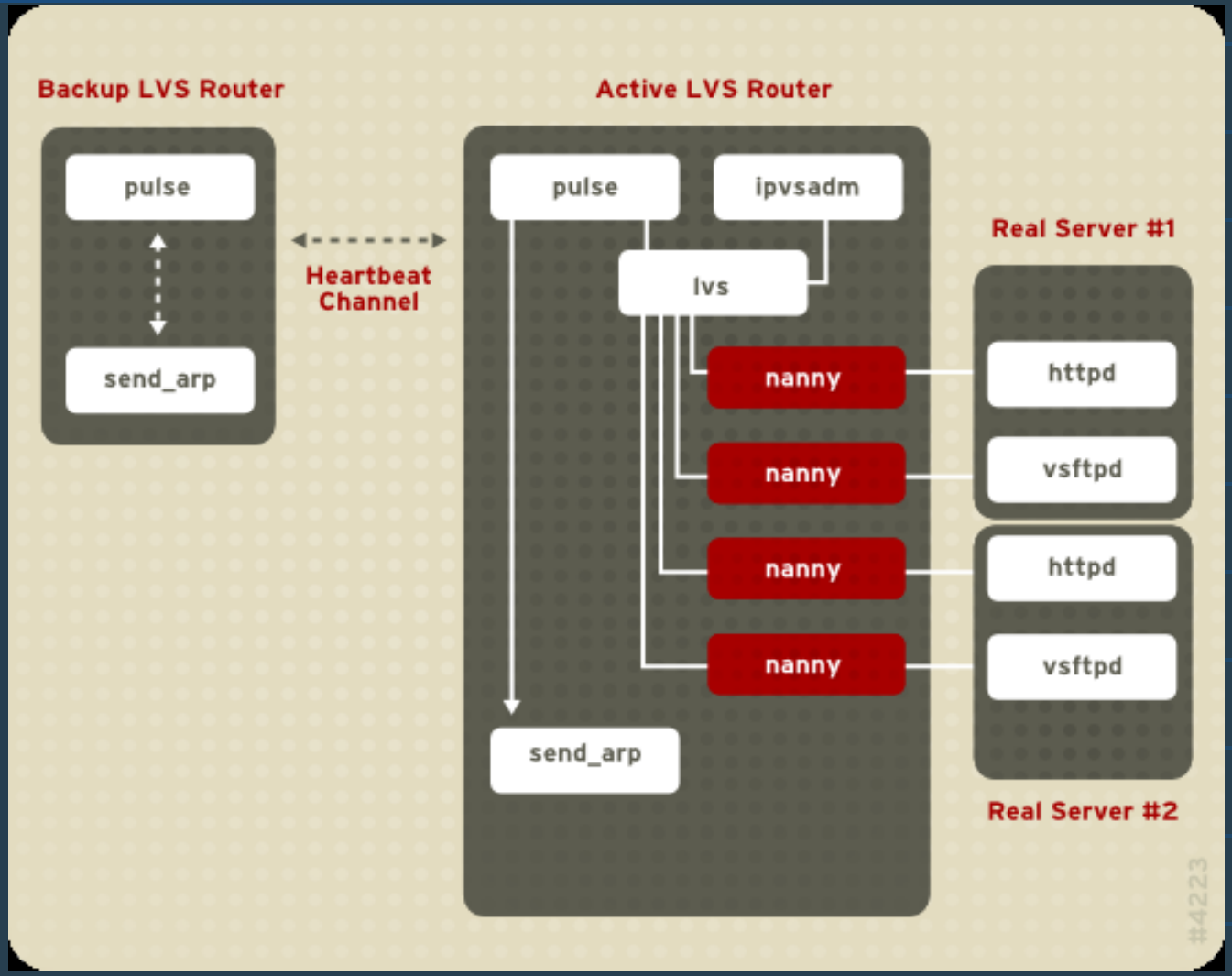
- Программы, д анные,процессы реплицируются в 2-х узлах
- Пара ведет себя как единая система
- Система становится логической концепцией как процесс
- Сохранение журнала транзакций на Backup узле
- Переключение на Backup в случае сбоя

RedHat Cluster Suite

- Global File System
- Cluster Logical Volume Manager
- Global Network Block Device
- Linux Virtual Server



RedHat Cluster Suite



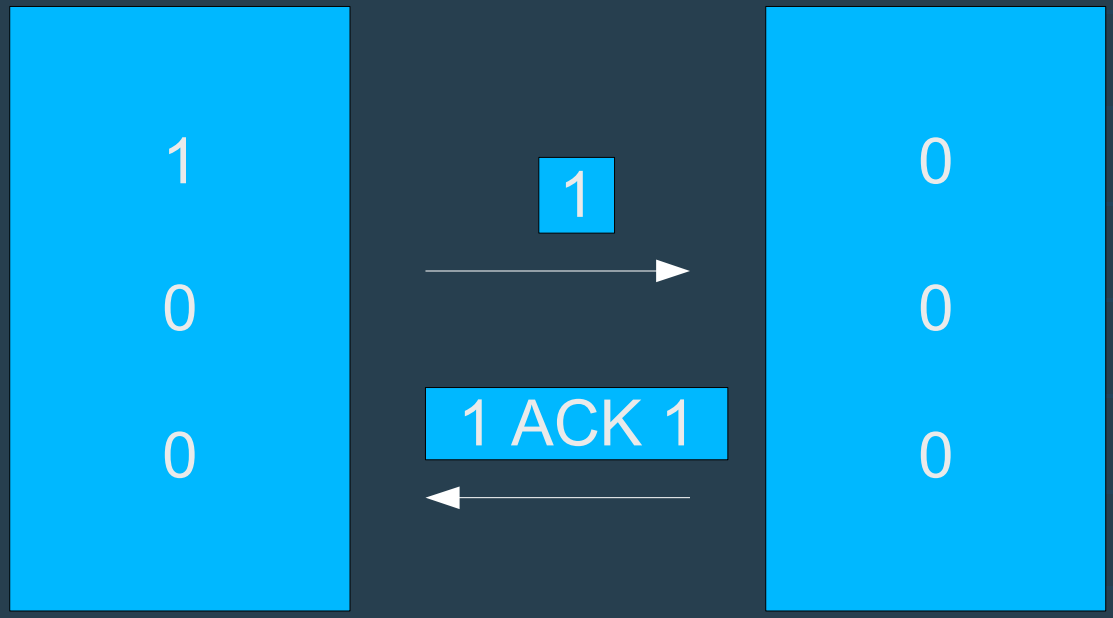
Надежная передача данных

- Общая схема работы
- Протоколы TCP, IPX

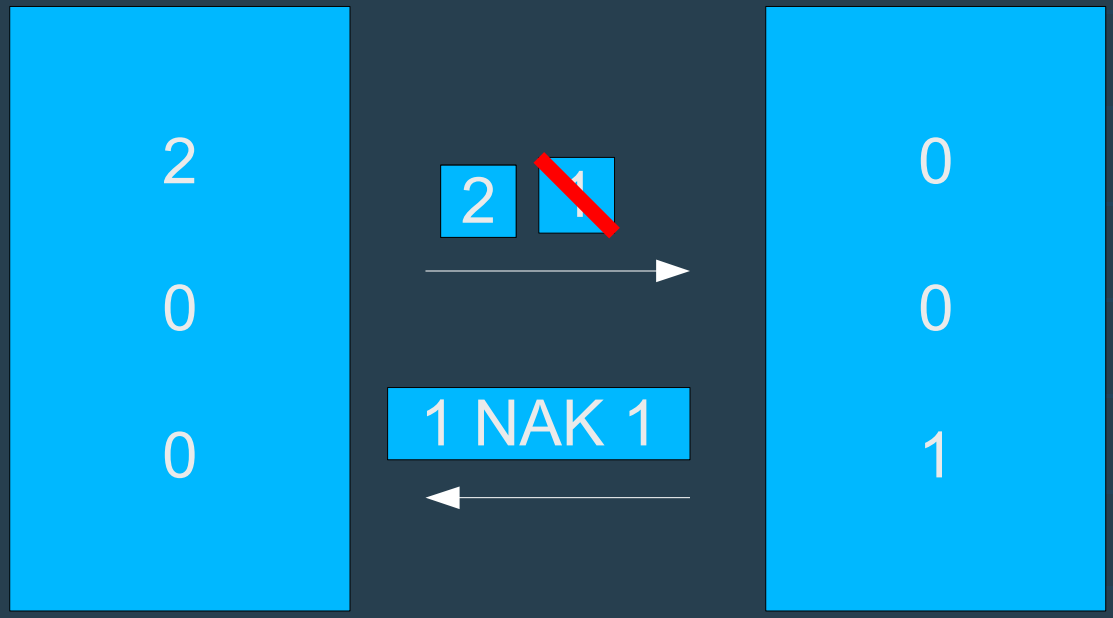
Нумерация пакетов

- Узел А, Узел В
  - Номер последнего переданного пакета
  - Номер последнего подтвержденного пакета
  - Номер последнего полученного пакета

Отправление пакета



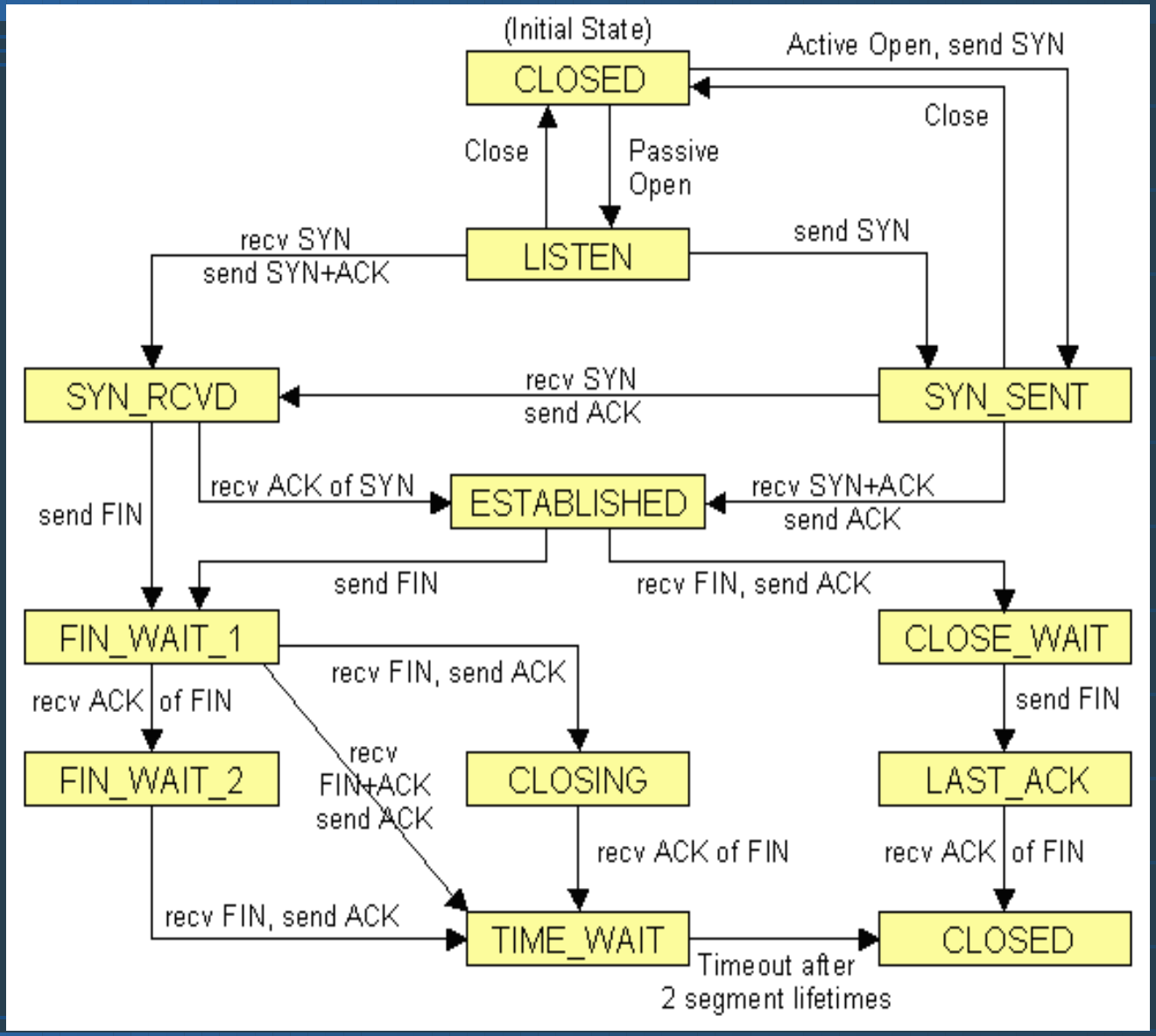
Неподтверждение пакета



Возможная оптимизация

- Выборочное подтверждение
  - “Не получены 1,6,7 пакет”
- Групповое подтверждение
  - “Получены все пакеты до 5-го включительно”
- Динамическое изменение окна протокола (кол-во неподтвержденных пакетов до ожидания)

Протокол TCP



Вопросы?