

Отказоустойчивость

Мельников В.М., 2020

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”

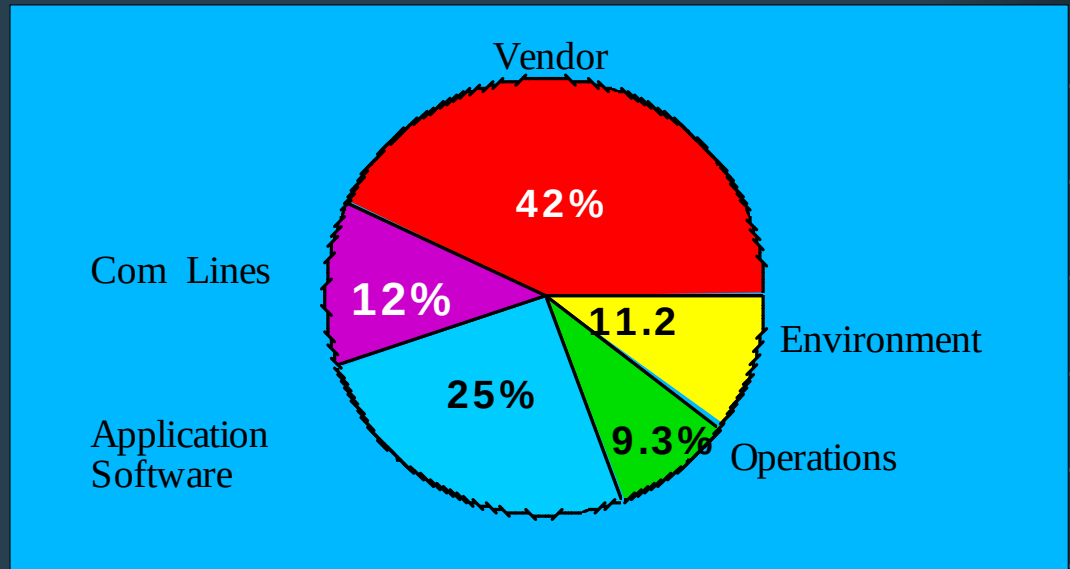
- ❶ Самолет с двумя двигателями имеет в два раза больше проблем
- ❷ Отказоустойчивость – ключевой компонент
- ❸ Маскируем и восстанавливаем ошибки

Надежность – ЭТО ...

- ❑ Достоверность (reliability)
 - ❑ Целостность (integrity)
 - ❑ Доступность (availability)
 - ❑ Безопасность (security)
-
- ❑ $\text{MTTF} / (\text{MTTF} + \text{MTTR})$



Как часто происходят ошибки

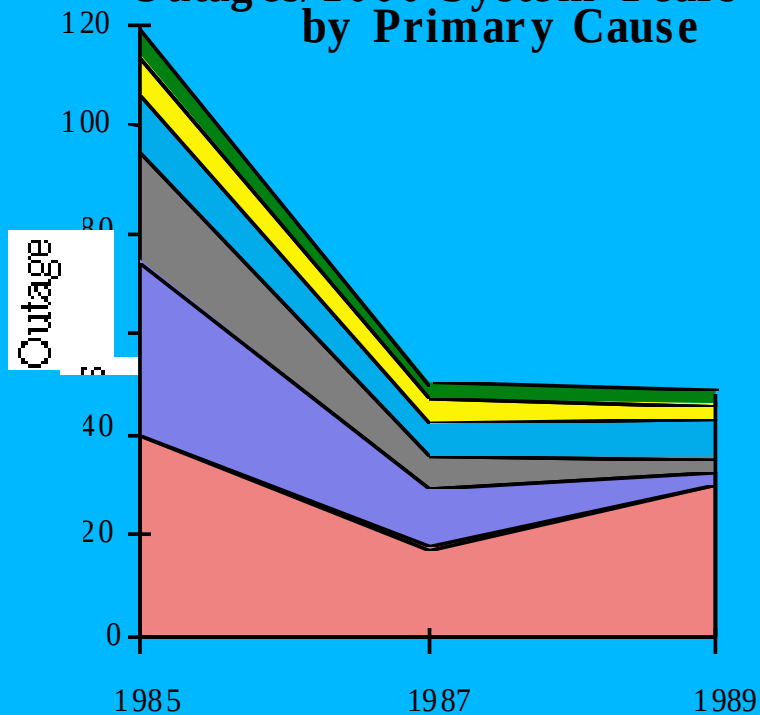


1,383 института опрошены (6/84 - 7/85)

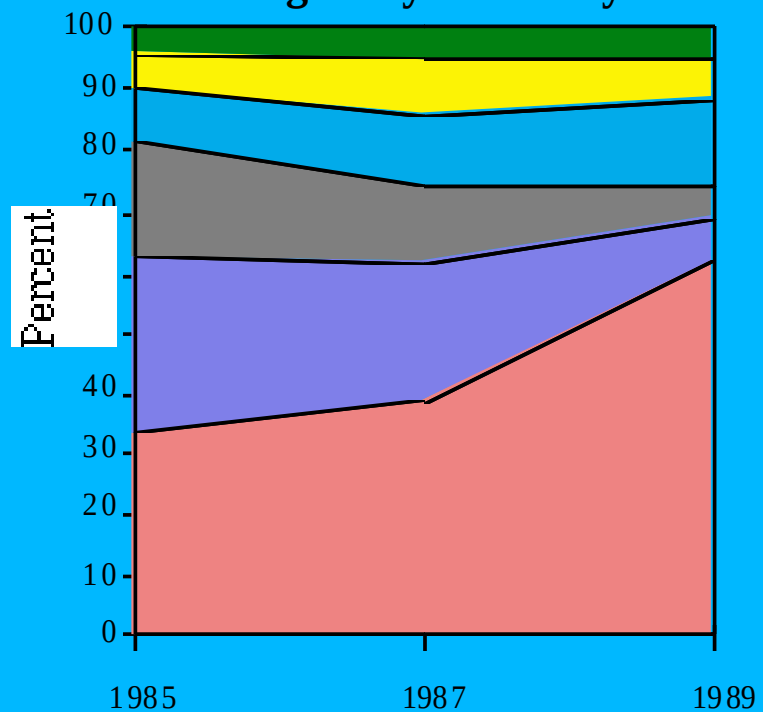
7,517 отказов, MTTF ~ 10 недель, средняя длительность ~ 90 минут

Тенденции

**Outages/1000 System Years
by Primary Cause**



% of Outages by Primary Cause



■ unknown ■ environment ■ operations ■ maintenance ■ hardware ■ software

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

Ключевые идеи обеспечения отказоустойчивости



- ❑ Программное обеспечение устраняет/автоматизирует выполнение операций
- ❑ Все проблемы граничат с ошибками программного обеспечения и дизайна
- ❑ Отказоустойчивость 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

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

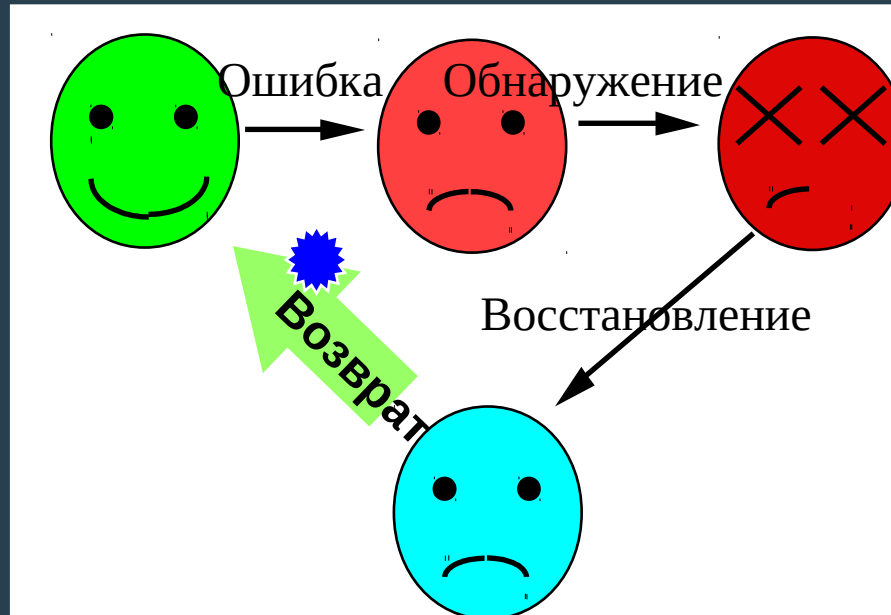
MESSAGE BASED OS: Изоляция ошибок, нет разделяемой памяти

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

PROCESS PAIRS : Маскирование проблем аппаратного и программного обеспечения

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

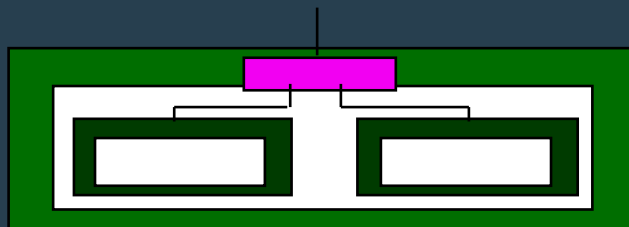
Fail-fast модули



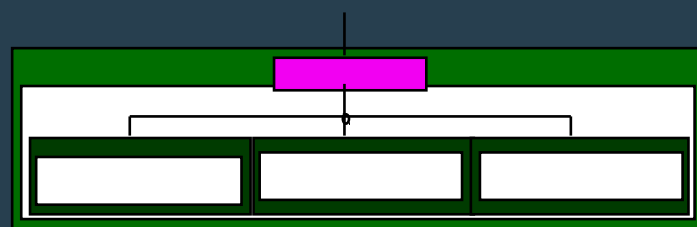
- ❑ Короткое время обнаружения ошибки
- ❑ Основная задача состоит в улучшении MTTF и MTTR

Аппаратное обеспечение. Мультиплексирование

Пара



Триплекс

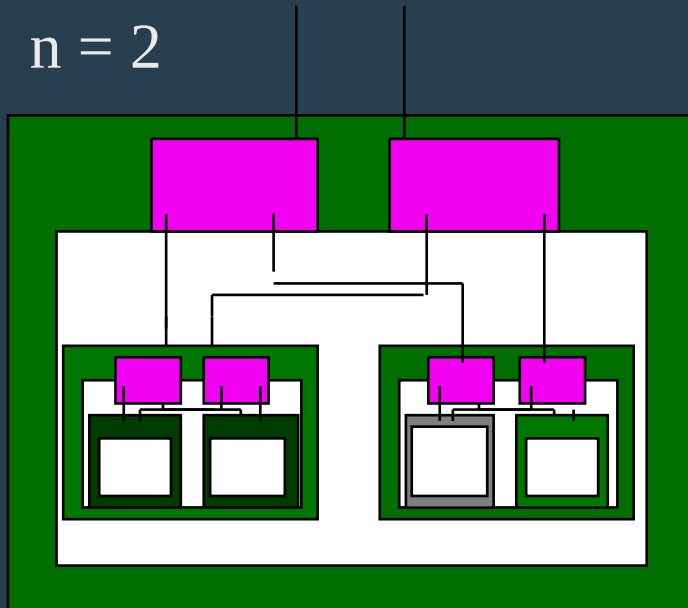


❶ Стратегии голосования

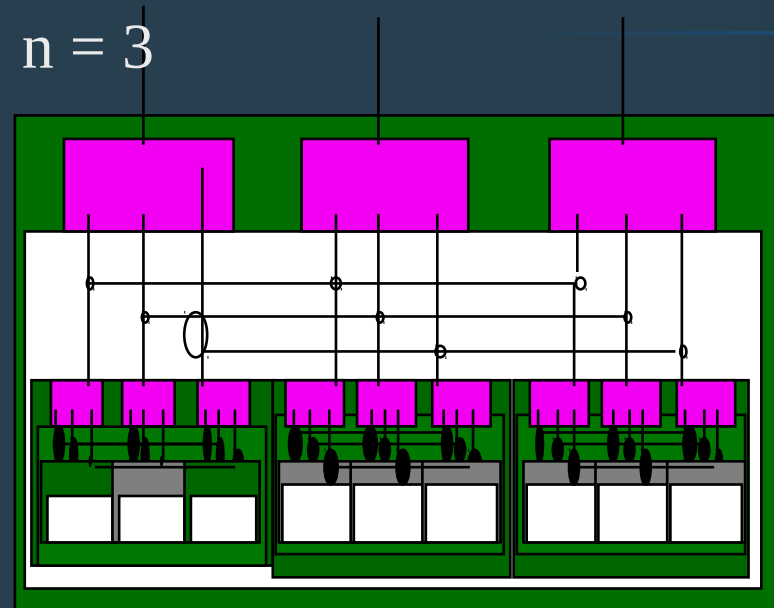
- ❶ Fail-fast – ошибка, если не работает половина устройств
- ❶ Fail-soft – ошибка, если не работают все

Аппаратное обеспечение. Рекурсивное построение

$n = 2$



$n = 3$

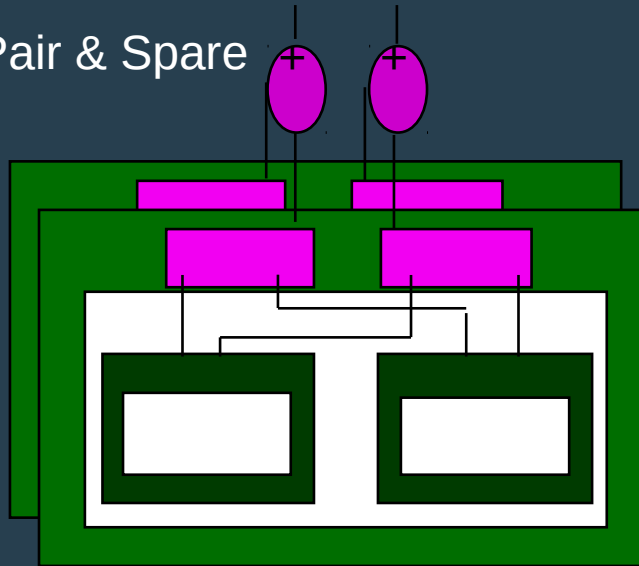


Аппаратное обеспечение.

Запасная пара

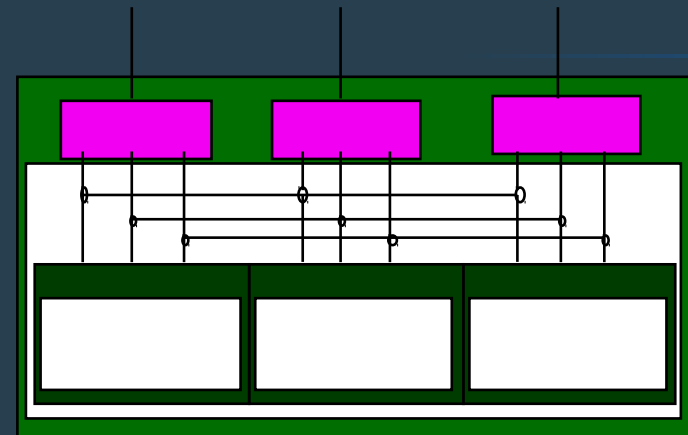
$n = 2$

Pair & Spare



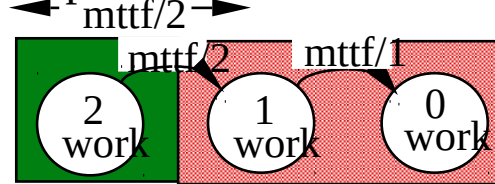
$n = 3$

Triple Modular Redundancy

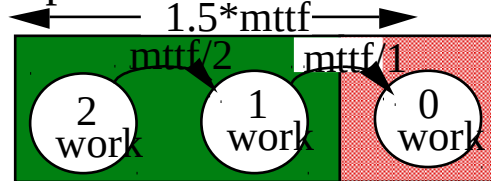


Доступность модуля без восстановления

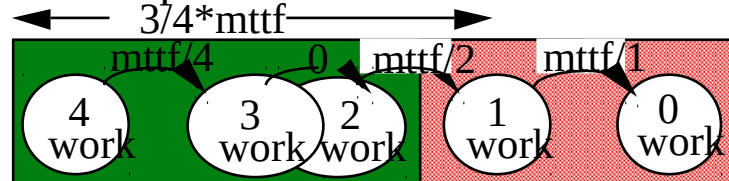
Duplex fail fast



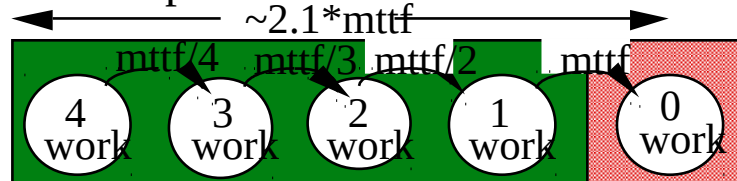
Duplex: fail soft



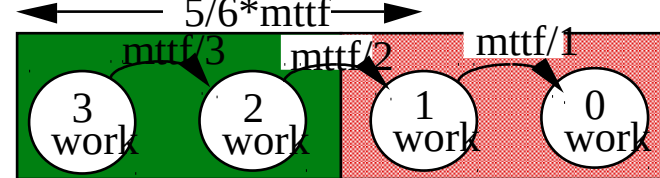
Pair & Spare: fail fast



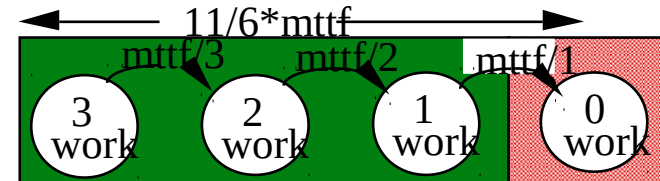
Pair & Spare: fail soft



TMR: fail fast

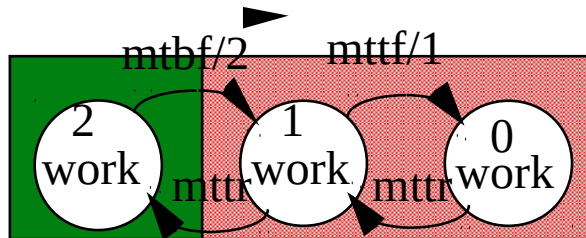


TMR: fail soft

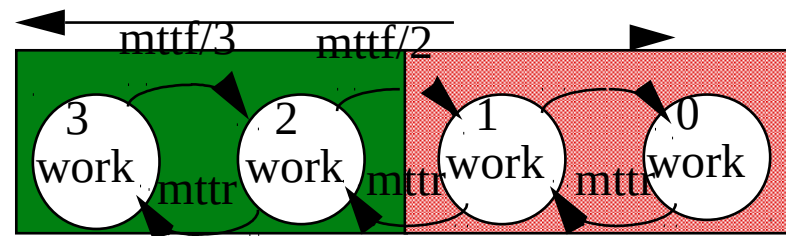


Доступность модуля с восстановлением

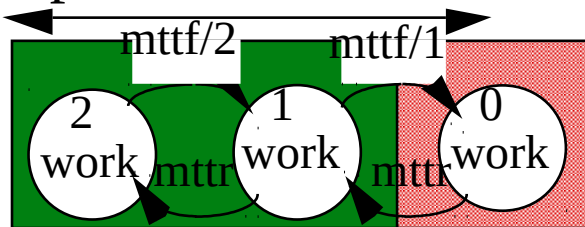
Duplex: fail fast: $\text{mttf}/2$



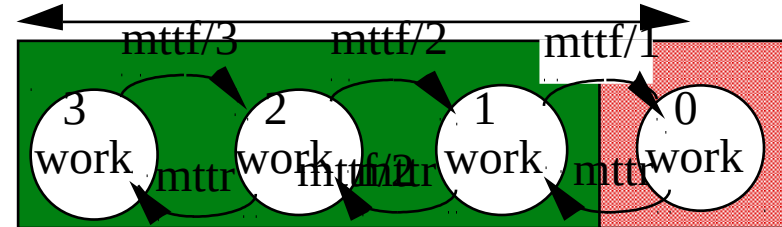
TMR: fail fast $\overset{4}{10} \text{ mttf}$



Duplex: fail soft $\overset{4}{10} \text{ mttf}$



TMR: fail soft $\overset{5}{10} \text{ mttf}$



Доступность модуля с восстановлением

Availability estimates 1 year MTTF modules 12-hour MTTR			MEQUA T TION T F
SIMPLEX	1 year	MTTF	1
DUPLEX: FAIL FAST	~0.5 years	$\approx \text{MTTF}/2$	$2+\varepsilon$
DUPLEX: FAIL SOFT	~1.5 years	$\approx \text{MTTF}(3/2)$	$2+\varepsilon$
TRIPLEX: FAIL FAST	.8 year	$\approx \text{MTTF}(5/6)$	$3+\varepsilon$
TRIPLEX: FAIL SOFT	1.8 year	$\approx 1.8\text{MTTF}$	$3+\varepsilon$
Pair and spare: FAIL-FAST	~.7 year	$\approx \text{MTTF}(3/4)$	$4+\varepsilon$
TRIPLEX WITH REPAIR	>10 ⁵	$\text{MTTF}^3/3\text{MTTR}^2$	$3+\varepsilon$

Серверные решения

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

Отказоустойчивость для систем хранения 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

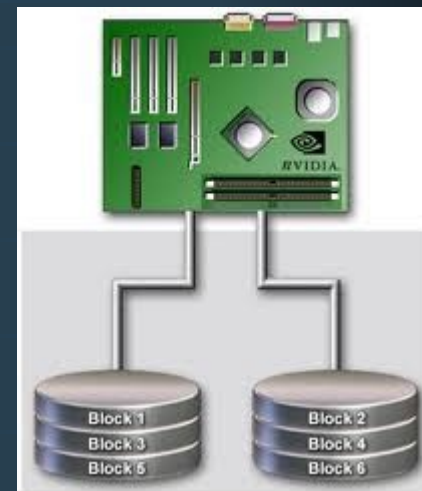
- Контрольная сумма
- Адрес вместе с данными
- Копия данных

■ В логике

- Скрытое контрольное чтение после записи
- Скрытое восстановление во время чтения

RAID

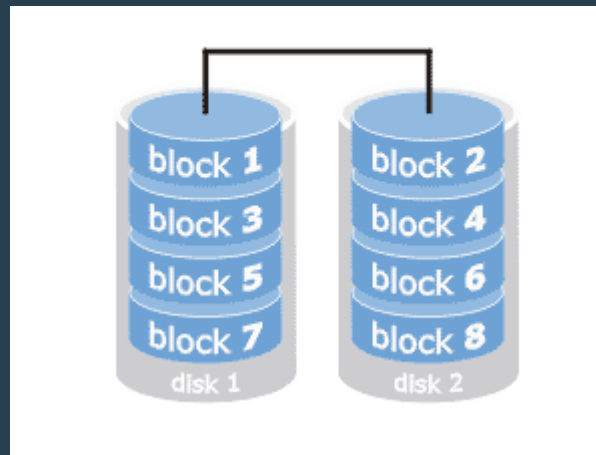
Redundancy
Array of
Inexpensive/Independent
Disks



- 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

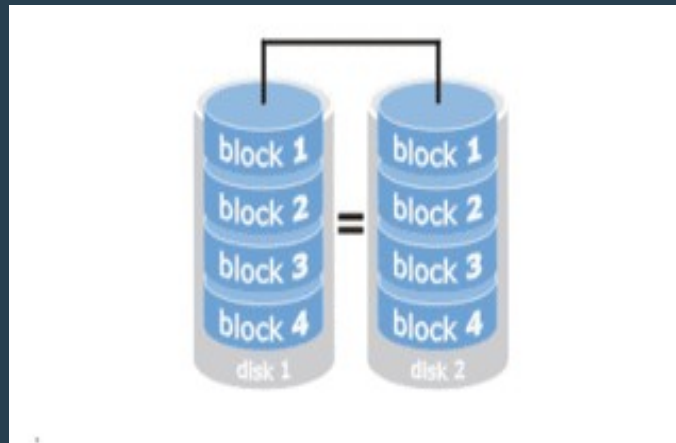
RAID 0

- Дисковый массив с чередованием без отказоустойчивости/чётности



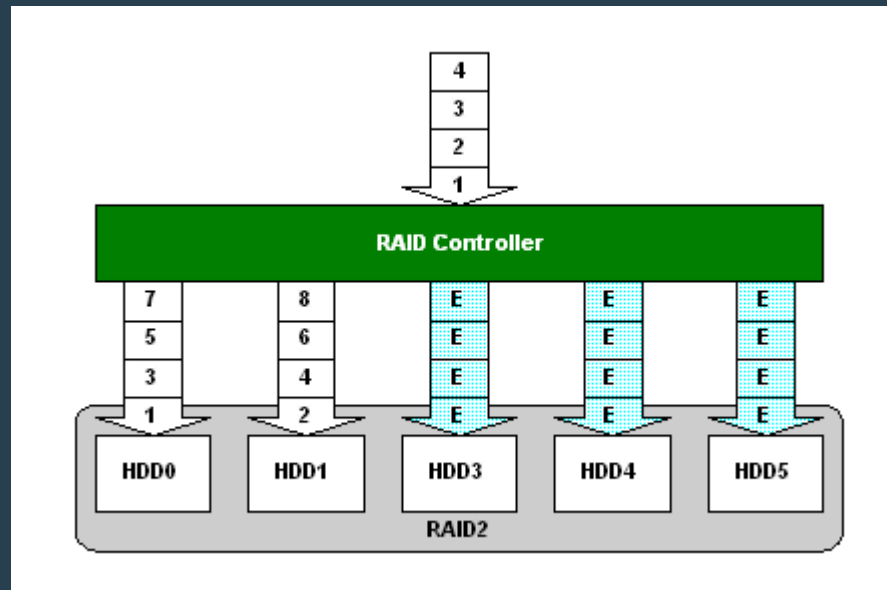
RAID 1

Зеркалирование



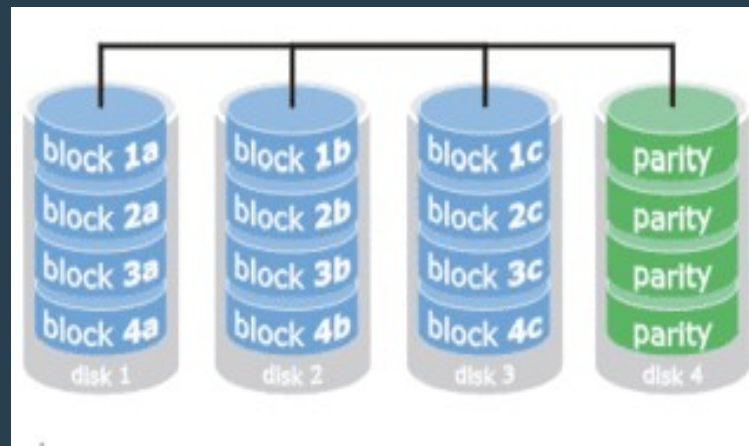
RAID 2

- Массив с использованием помехоустойчивого кода Хемминга (ЕСС память)



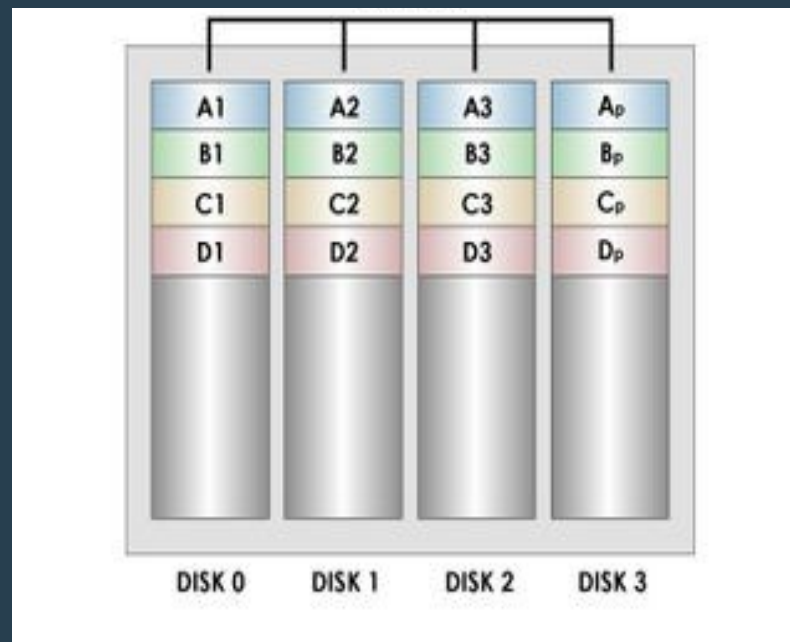
RAID 3

- Отказоустойчивый массив с битовым чередованием и чётностью



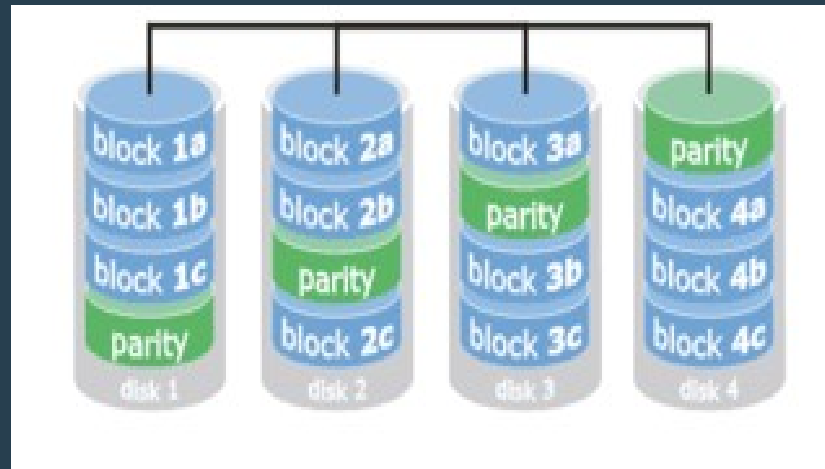
RAID 4

- Отказоустойчивый массив с блочным чередованием и чётностью



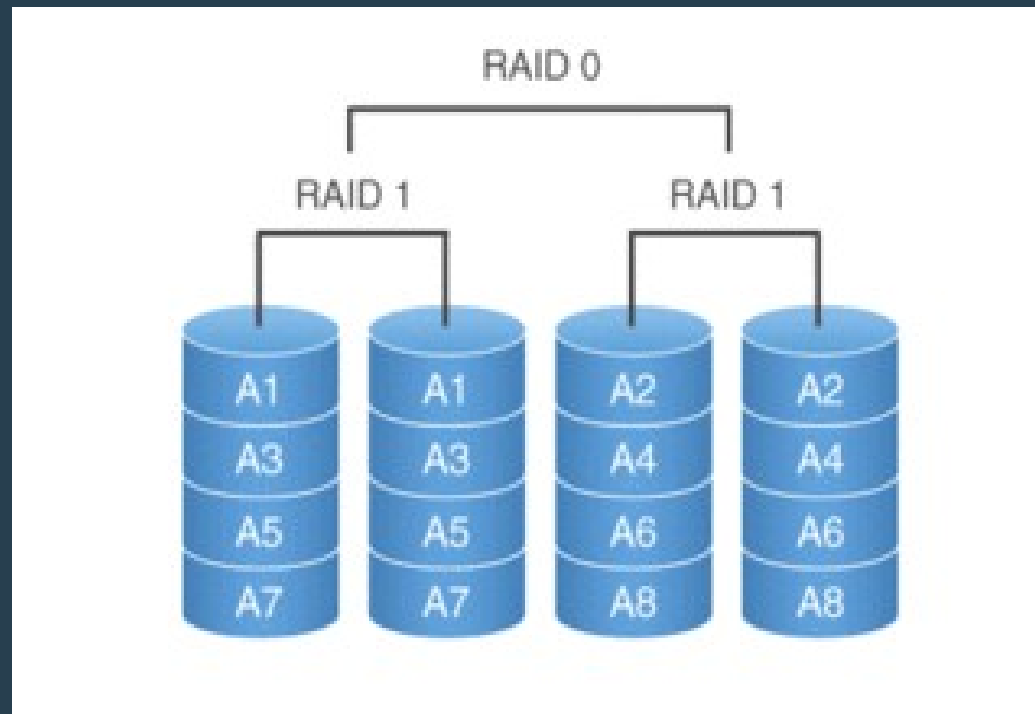
RAID 5

- 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

- ❶ Процессы с хранением состояния
- ❷ Пара процессов

Процесс с сохранением состояния

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

Процесс с сохранением состояния

❏ Печать лотерейных 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 лотерейных билетов

```
if (restart)
```

```
    read ( f, i);
```

```
else
```

```
    i = 0;
```

```
while ( i < N ) {
```

```
    i = i + 1;
```

```
    write ( f, i );
```

```
    PRINT_TICKET( i );
```

```
}
```

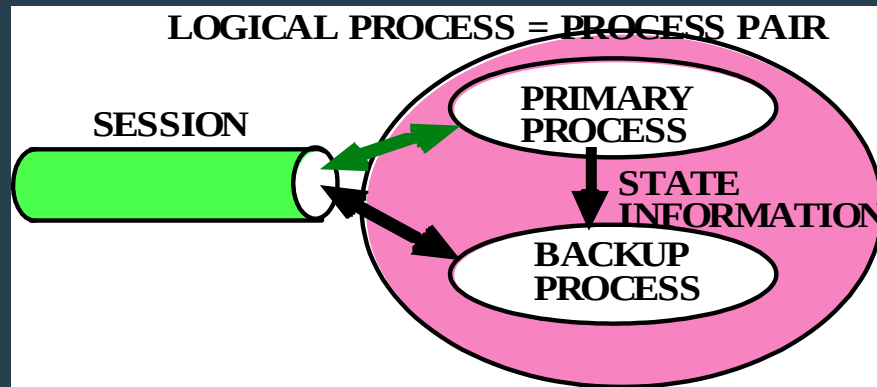
Процесс с сохранением состояния

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

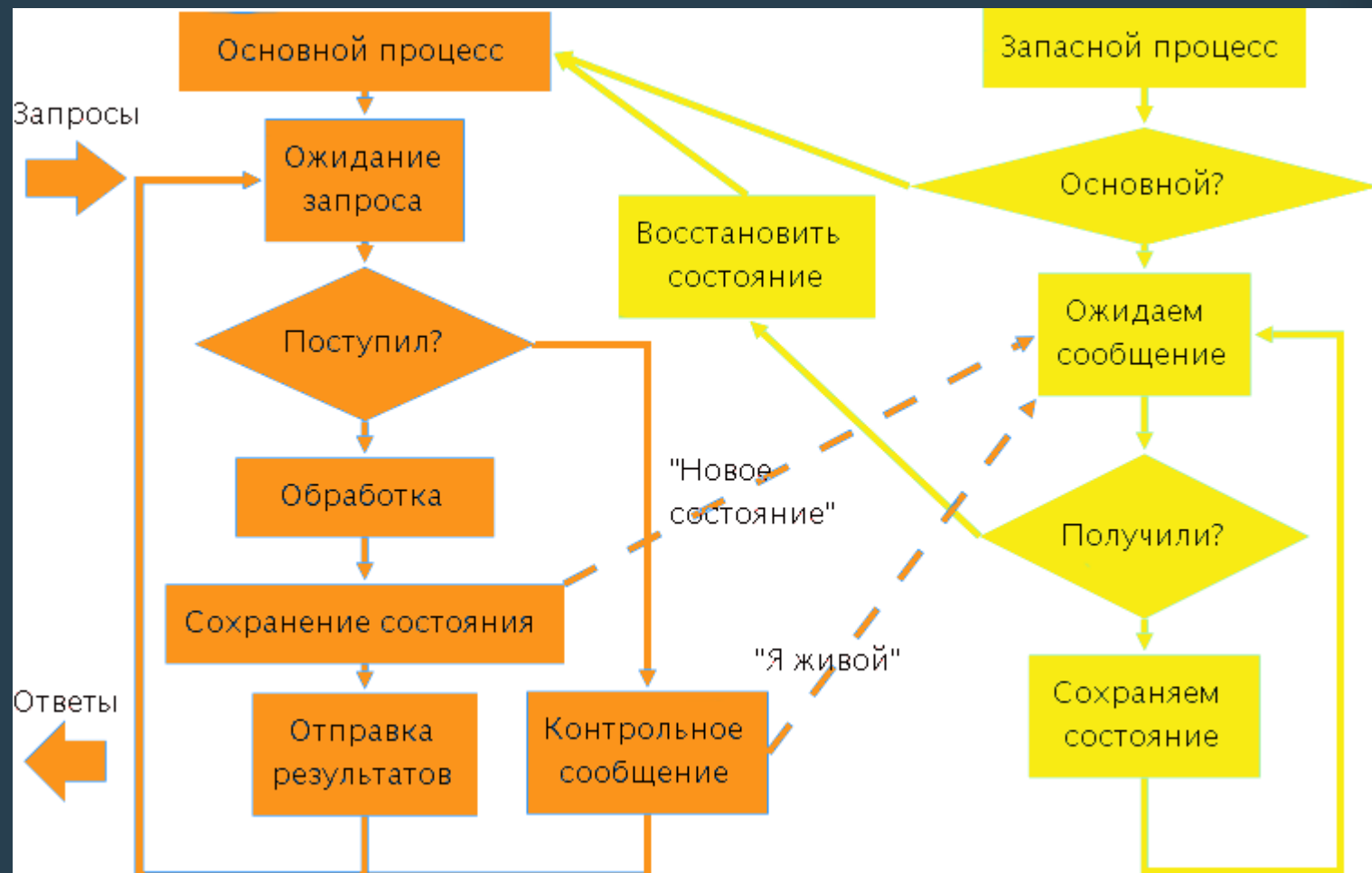
Пара процессов

ОСНОВНОЙ + ЗАПАСНОЙ =
ЛОГИЧЕСКИЙ ПРОЦЕСС

- ❑ Как обнаружить сбой?
- ❑ Как продолжить работу?



Блок схема работы процессов



Пара процессов

❶ Достоинства

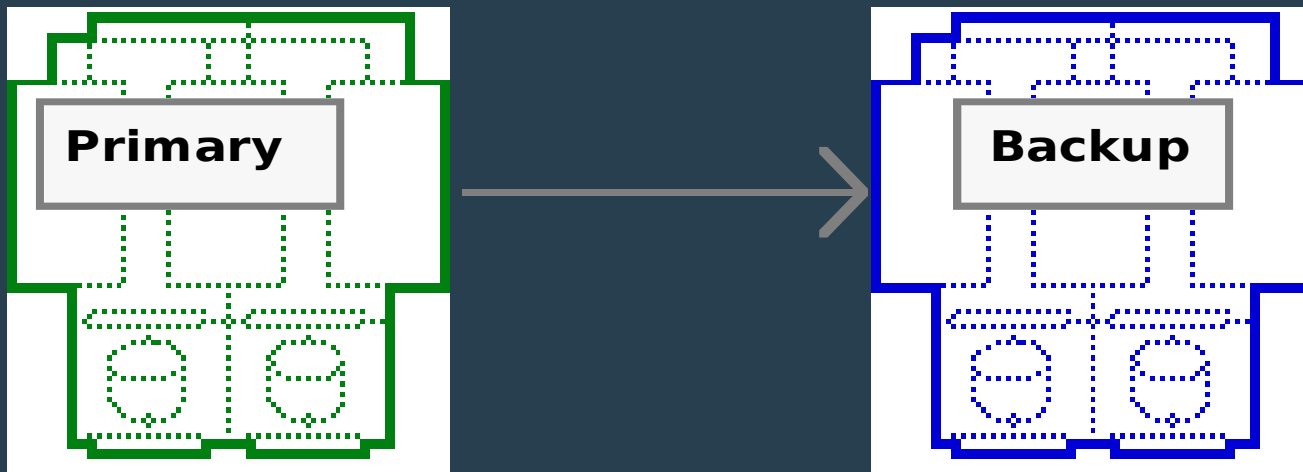
- ❶ Высокая скорость переключения
- ❶ Возможно масштабирование

❷ Недостатки

- ❶ Высокая нагрузка на сеть

Отказоустойчивость систем.

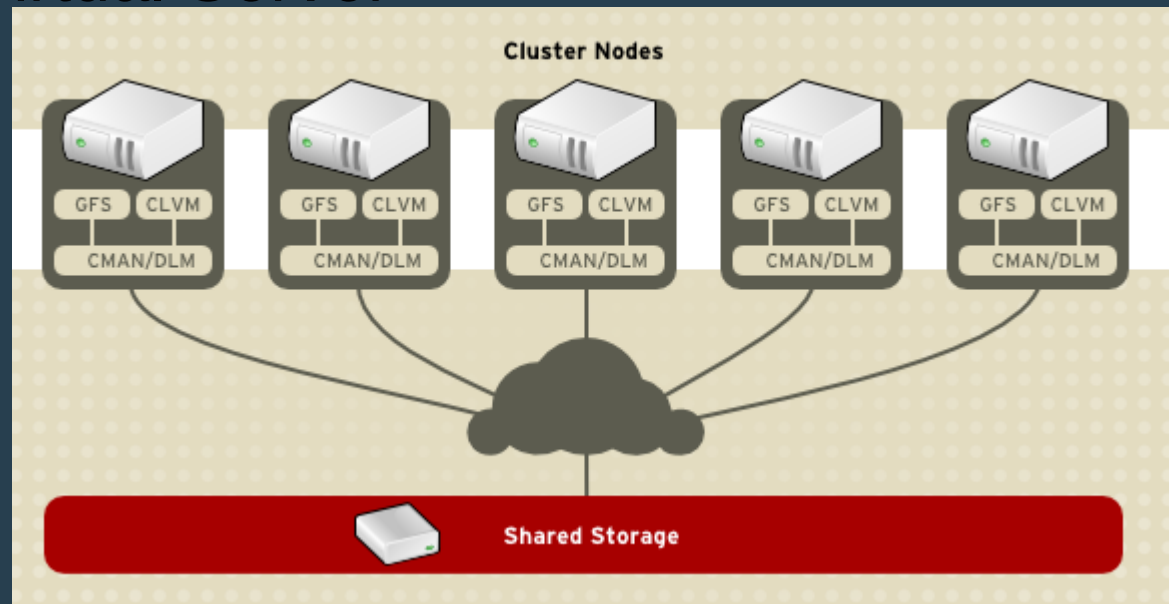
Общая концепция



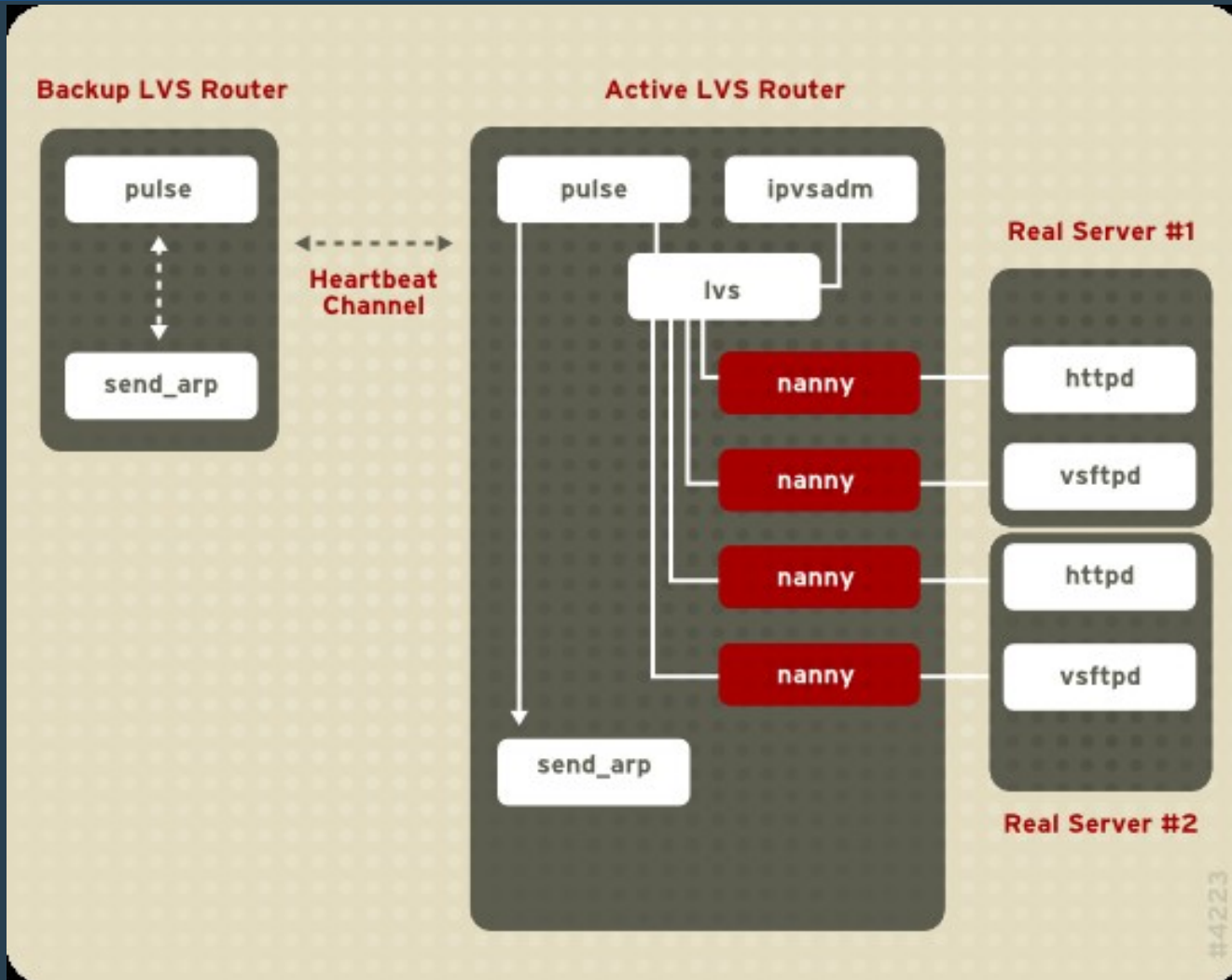
- ❑ Программы, данные, процессы реплицируются в 2-х узлах
- ❑ Пара ведет себя как единая система
- ❑ Система становится логической концепцией как процесс
- ❑ Сохранение журнала транзакций на Backup узле
- ❑ Переключение на Backup в случае сбоя

RedHat Cluster Suite

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



RedHat Cluster Suite



Надежная передача данных

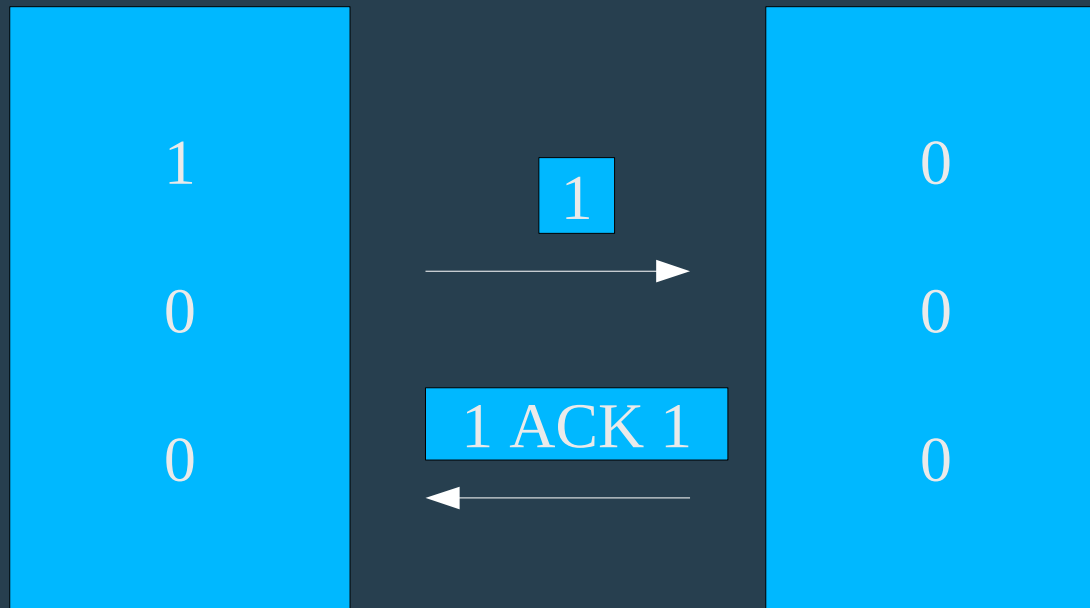
- ❶ Общая схема работы
- ❷ Протоколы ТСР, IPX

Нумерация пакетов

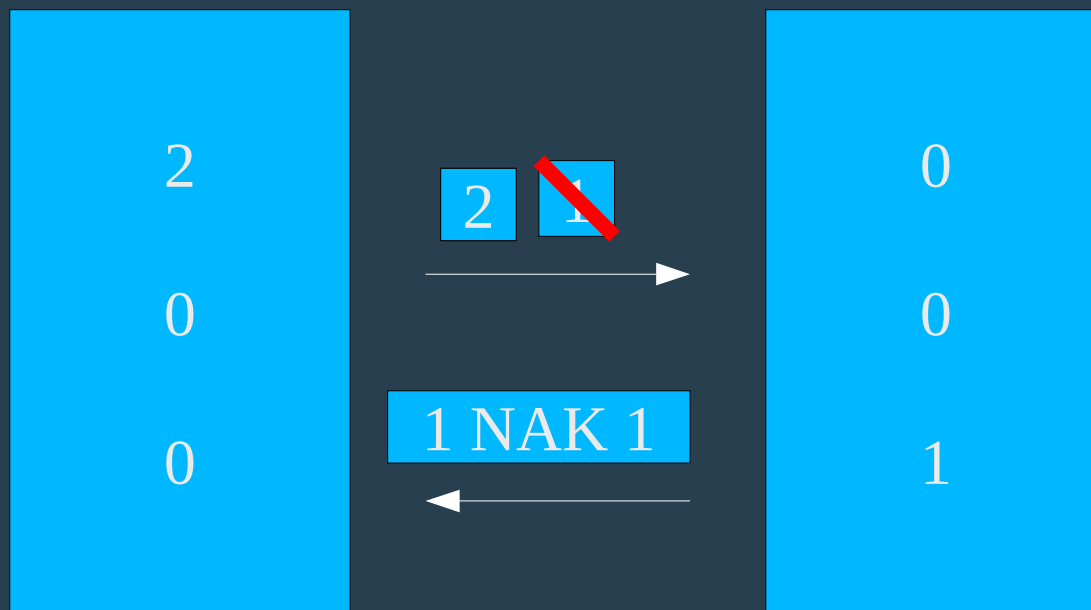
❶ Узел А, Узел В

- ❷ Номер последнего переданного пакета
- ❸ Номер последнего подтвержденного пакета
- ❹ Номер последнего полученного пакета

Отправление пакета



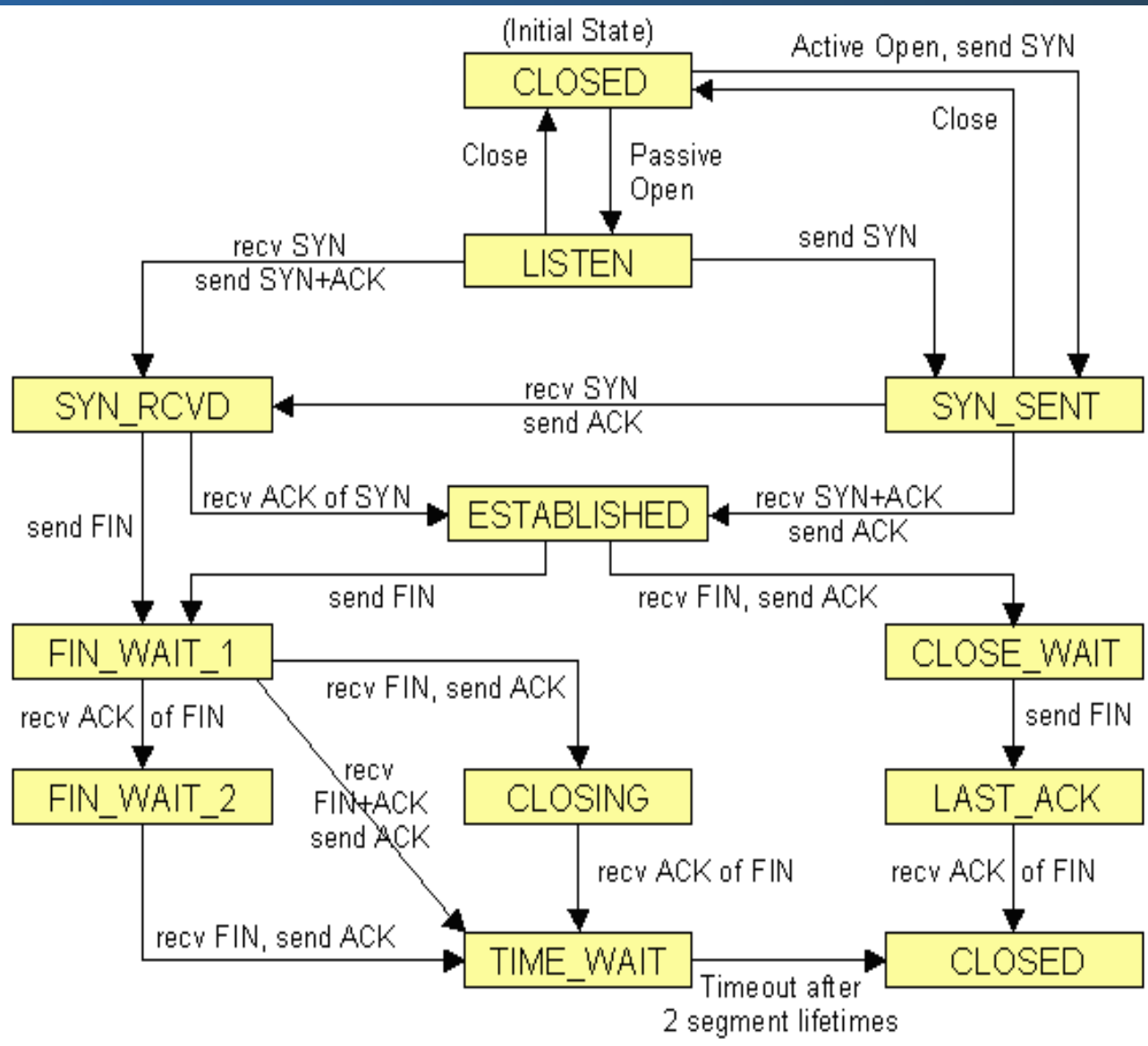
Неподтверждение пакета



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





www.pictofgo.com

Вопросы?