



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Linux Bonding: отказоустойчивость сети и агрегация каналов

Как мы закладываем active-backup и LACP 802.3ad на серверах 1С, гипервизорах и NAS клиентов

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Сервер 1С или гипервизор Proxmox клиента подключён к коммутатору одним кабелем в один порт сетевой карты. Отказ SFP-модуля, патч-корда, порта коммутатора или карты — и 30–50 сотрудников теряют 1С, RDP и общие папки до приезда инженера. Мы закрываем эту точку отказа bond-интерфейсом ядра Linux: два порта в active-backup или LACP, переключение за 100–300 мс, замена карты на ходу.

Почему это важно бизнесу

- Один патч-корд или порт коммутатора — единственная точка отказа сервера 1С: час простоя 40 рабочих мест дороже второй сетевой карты.
- Замена сетевой карты без bond — плановый простой ночью с выездом; с active-backup меняем порт в рабочее время без остановки сервисов.
- LACP из двух портов 10G снимает узкое место бэкапов Veeam/PBS и обмена 1С↔SQL, не требуя покупки 25G-карт и коммутаторов.
- Bond с MII-мониторингом переключается за 100–300 мс: TCP-сессии RDP и 1С переживают обрыв, пользователи не замечают инцидента.



Ключевые параметры реализации

100 мс

miimon: период опроса состояния
линка каждого порта; потеря carrier
видна на ближайшем цикле

bonding.rst, параметр miimon

200 мс

updelay/downdelay, кратны miimon
(иначе ядро округляет вниз) —
защита от флаппинга линка

наш стандарт; bonding.rst updelay

mode=1

active-backup — база для серверов
до 50 PM: без настройки свитча,
порты в два разных свитча

bonding.rst, параметр mode

802.3ad

LACP (mode=4) с lacp_rate=fast:
LACPDU раз в 1 с вместо 30 с — где
свитч держит port-channel

bonding.rst lacp_rate; IOS XE 17.9

layer3+4

xmit_hash_policy для LACP: хеш по
IP+порт раскладывает потоки
одного хоста по обоим линкам

bonding.rst, xmit_hash_policy

x2 10G

Dual-port 10G Intel X710 на Dell
R640/R740 — штатная
комплектация серверов, которые
мы сдаём

наш стандарт



active-backup через systemd-networkd на сервере 1C (Ubuntu 24.04)

Что настраиваем

Сервер 1C + PostgreSQL клиента на Dell R640, два порта 10G в два независимых коммутатора доступа

Как мы это делаем

- 1 10-bond0.netdev: [NetDev] Name=bond0 Kind=bond; [Bond] Mode=active-backup MIIMonitorSec=100ms UpDelaySec=200ms DownDelaySec=200ms FailOverMACPolicy=none
- 2 20-ens1f0.network и 20-ens1f1.network: [Match] Name=...; [Network] Bond=bond0; на первом порту PrimarySlave=true — трафик возвращается на него после восстановления
- 3 30-bond0.network: [Link] RequiredForOnline=yes; [Network] Address=/Gateway=/DNS=, VLAN=bond0.20 для сети 1C; коммутаторам ничего настраивать не нужно
- 4 networkctl reload && networkctl reconfigure bond0; проверка cat /proc/net/bonding/bond0: MII Status up у обоих slave, Link Failure Count 0
- 5 Приёмочный тест: ping -i 0.2 с соседнего хоста, ip link set ens1f0 down — допускаем ≤ 2 потерянных пакета; active_slave в sysfs должен смениться на ens1f1

РЕЗУЛЬТАТ

Отказ любого патч-корда, SFP или одного из двух коммутаторов переживается за 200–300 мс без разрыва RDP и сеансов 1C. Сетевую карту меняем в рабочее время: переводим active_slave на второй порт через sysfs, выводим первый из bond командой ip link set ... nomaster.

КЛЮЧЕВОЙ НЮАНС

Bond получает MAC первого slave; при fail_over_mac=none оба порта светятся одним MAC — на коммутаторах с port-security это ловится как MAC-flap. Для таких свитчей ставим FailOverMACPolicy=active.



LACP 802.3ad: гипервизор Proxmox VE 8/9 + Cisco Catalyst 9300

Что настраиваем

Proxmox VE 8.x/9.x (ifupdown2), bond0 из двух 10G в один коммутатор Catalyst 9300, поверх — vbr0 с VLAN-aware bridge

Как мы это делаем

- 1 /etc/network/interfaces: iface bond0 — bond-slaves eno1 eno2, bond-mode 802.3ad, bond-miimon 100, bond-xmit-hash-policy layer3+4, bond-lacp-rate fast
- 2 vbr0: bridge-ports bond0, bridge-stp off, bridge-fd 0, bridge-vlan-aware yes, bridge-vids 2-4094; IP управления — на vbr0.10, а не на самом bond
- 3 Catalyst 9300 (IOS XE 17.9): Port-channel1 switchport mode trunk; на Te1/1/1-2 channel-group 1 mode active + lacp rate fast; load-balance src-dst-mixed-ip-port
- 4 ifreload -a без перезагрузки; в /proc/net/bonding/bond0 Aggregator ID одинаков у обоих портов, Partner Mac Address = MAC свитча, Actor/Partner Churn State: none
- 5 Corosync на bond не сажаем: по документации PVE он сам переключается между несколькими link, а часть bond-режимов для него проблемна — даём ему отдельные интерфейсы

РЕЗУЛЬТАТ

Обе 10G-линии работают одновременно: ночной бэкап на PBS и обмен VM 1C↔SQL идут параллельно, а отказ одного порта не роняет ни одну VM. Смена модуля SFP+ — штатная операция без миграции VM.

КЛЮЧЕВОЙ НЮАНС

Один поток TCP всегда идёт через один линк: LACP даёт 2×10G только суммарно по множеству сессий. Если у клиента один сервер бэкапа — выигрыш даёт layer3+4, а не «удвоение скорости» одной копии.



Мониторинг bond в Zabbix 7.0 и приёмочный тест отказа

Что настраиваем

Все Linux-хосты клиентов с bond: серверы 1C, Proxmox, PBS, Debian-шлюзы — под шаблоном «Linux by Zabbix agent»

Как мы это делаем

- 1 Шаблон «Linux by Zabbix agent»: LLD «Network interface discovery» видит bond0 и slave-порты; {\$NET.IF.IFNAME.NOT_MATCHES} не должен их отсекалть
- 2 Свои items:
vfs.file.contents[/sys/class/net/bond0/bonding/active_slave] и .../<slave>/bonding_slave/mii_status; триггер на смену active_slave (change()<>0), Warning
- 3 Штатный триггер «Interface {#IFNAME}: Link down» (Average) для slave-портов bond поднимаем до High: bond работает, но резерва нет — заявка на выезд, не ночной алерт
- 4 Link Failure Count из /proc/net/bonding/bond0 как item: рост при живом линке = дребезг SFP/патч-корда, меняем до полного отказа
- 5 Раз в квартал плановый failover: ip link set <port> down на 60 с под ping -i 0.2, фиксируем потери и возврат на primary после updelay

РЕЗУЛЬТАТ

Отказ резервного линка перестаёт быть «тихим»: без мониторинга bond маскирует деградацию неделями, пока не падает второй порт. Мы узнаём о потере резерва за 1–2 минуты и меняем кабель или SFP планово, а не аварийно.

КЛЮЧЕВОЙ НЮАНС

Штатные item'ы на bond0 (net.if.status, трафик) потерю резерва не видят: bond0 остаётся up. Деградацию ловят только mii_status и link_failure_count slave в bonding_slave sysfs и Link down физического порта.



Подводные камни

✗ **updelay не кратен miimon**

Ядро округляет updelay вниз до кратного miimon: 150 мс при miimon=100 станет 100. Держим 100/200/200 и сверяем «Up Delay» в /proc/net/bonding.

✗ **LACP без port-channel на коммутаторе**

mode=4 без LAG-группы на свитче: агрегатор не собирается, трафик идёт по одному порту или рвётся. Сначала channel-group mode active, потом bond.

✗ **miimon и arp_interval одновременно**

Ядро не совмещает ARP-мониторинг с MII, а для 802.3ad/tlb/alb отвергает arp_interval. MII — для LACP, ARP-таргеты (по одному на свитч) — для mode=1.

✗ **Порты bond на двух свитчах в LACP**

802.3ad требует, чтобы все линки шли к одному партнёру (или MLAG/vPC-паре). Два независимых коммутатора — только active-backup.

✗ **IP на физическом порту внутри bond**

IP или DHCP на slave-интерфейсе ломает маршрут по умолчанию при failover. Slave — только manual/Bond=, весь L3 на bond0 или bond0.VLAN.

✗ **VLAN создан до первого slave**

До первого slave у bond MAC 00:00:00:00:00:00; VLAN, созданный раньше enslave, наследует пустой MAC. В networkd порядок задаёт VLAN= в bond0.network.

✗ **Один MAC на двух портах и port-security**

При fail_over_mac=none оба slave несут MAC bond — со sticky port-security порт свитча уходит в err-disable. Ставим fail_over_mac=active или снимаем s...

✗ **Резерв «умер», а bond молчит**

active-backup продолжает работать на одном линке без ошибок в приложениях. Без триггера на mii_status slave потерю резерва замечают при втором отказе.



Как правильно

МИНИМУМ

- Два порта NIC в active-backup, miimon=100, updelay/downdelay=200, primary задан
- Порты в разные коммутаторы или хотя бы в разные линейные карты/блоки питания
- Тест failover ping + ip link set down перед сдачей, результат в паспорт сервера

НОРМАЛЬНО

- Zabbix: триггеры на Link down физических портов и смену active_slave
- LACP 802.3ad, lacp_rate=fast, xmit_hash_policy=layer3+4 при управляемом свитче
- VLAN и bridge только поверх bond0; управление — на отдельном VLAN-интерфейсе
- Конфиг в git/laC (netplan, networkd, ifupdown2) с описанием портов свитча

ХОРОШО

- LACP на MLAG/vPC-пару коммутаторов: агрегация + переживание отказа целого свитча
- Алерт на «Number of ports» < 2 в /proc/net/bonding/bond0; min_links=2 не ставим
- Ежеквартальный плановый failover-тест с фиксацией потерь и Link Failure Count
- Отдельные линки под corosync/репликацию, не зависящие от bond данных



Чек-лист самопроверки

- | | |
|--|--|
| ☐ На каждом сервере, который нельзя гасить ночью, два физических порта собраны в bond? | ☐ Проведён и задокументирован тест отказа (потери ≤ 2 пакетов, возврат на primary)? |
| ☐ Режим bond выбран по коммутатору: active-backup без LAG, 802.3ad только при настроенном port-channel? | ☐ Zabbix шлёт алерт при падении любого физического порта bond, а не только при потере сети? |
| ☐ miimon=100, updelay/downdelay кратны miimon и это видно в /proc/net/bonding/bond0? | ☐ Для LACP: на обеих сторонах одинаковая скорость/дуплекс, Aggregator ID един, Partner MAC = свитч? |
| ☐ Порты bond уходят в разные коммутаторы или разные линейные карты одного шасси? | ☐ Конфигурация bond и портов коммутатора описана в паспорте сервера и хранится в git? |
| ☐ Ни на одном slave-интерфейсе нет собственного IP/DHCP — весь L3 живёт на bond0 или bond0.VLAN? | |

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Аудит серверов клиента: находим хосты с одним линком и считаем стоимость простоя против второго порта/карты
- Внедряем bond под ключ: networkd/netplan/nmcli/ifupdown2 + port-channel на Cisco/MikroTik/Zyxel, без простоя
- Приёмочный failover-тест с протоколом, паспорт сервера и схема портов коммутатора
- Подключаем к Zabbix 7.0: триггеры на потерю резерва, смену active_slave, дребезг линка
- Сопровождение: плановая замена SFP/патч-кордов по Link Failure Count, квартальные тесты отказа

15+

лет в ИТ-поддержке

50

рабочих мест — наш профиль

МТС

дата-центр, Москва



КОНТАКТЫ

Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru



Техническая база

- 01** Linux Ethernet Bonding Driver HOWTO (networking/bonding.rst) (docs.kernel.org — 6.x)
- 02** systemd.netdev(5) [Bond] и systemd.network(5) Bond=/PrimarySlave= (freedesktop.org — v255+)
- 03** Netplan YAML reference — bonds/parameters, vlans, bridges (netplan.readthedocs.io — 1.x)
- 04** RHEL 9: Configuring a network bond (nmcli, bond.options) (docs.redhat.com — RHEL 9)
- 05** Proxmox VE Admin Guide — Network Configuration: Linux Bond, VLAN bridge (pve.proxmox.com — PVE 8/9)
- 06** Cisco IOS XE 17.9 Catalyst 9300 — Configuring EtherChannels (LACP) (cisco.com — 17.9)
- 07** Zabbix template «Linux by Zabbix agent» — Network interface discovery (zabbix.com — 7.0)
- 08** Шаблон ITfresh: паспорт сервера + протокол failover-теста bond (itfresh.ru — 2026)

