



ЦИТАДЕЛЬ

ООО «Цитадель»
127015, г. Москва, ул. Новодмитровская, д. 2Б
+7 (495) 666 2 333, info@ctdl.ru

Программное обеспечение «ОЛИМП-374» в составе АПК «ОЛИМП-374 CS»

Руководство по установке и эксплуатации



Оглавление

1	Введение.....	3
1.1	Назначение документа.....	3
2	Состав «ОЛИМП-374».....	3
3	Конфигурация сервера.....	3
4	Установка ПО.....	5
4.1	Установка операционной системы и предварительная настройка.....	5
4.2	Установка и настройка пакетов съемника.....	5
4.2.1	Установка ПО для съема IP (медиа и/или сигнализации).....	5



1 Введение

1.1 Назначение документа

В документе описывается состав и процесс установки программного обеспечения «ОЛИМП-374» (далее – ПО «ОЛИМП-374») в составе аппаратно-программного комплекса «ОЛИМП-374» (далее - АПК «ОЛИМП-374»). Комплекс осуществляет функции пассивного съема трафика и передачи информации в систему накопления данных и на ПУ ОРМ-86 в сетях фиксированной и мобильной телефонной связи Оператора связи в рамках реализации требований ФЗ – 374 (закон «Яровой»).

2 Состав «ОЛИМП-374»

На данный момент предусмотрено три варианта съема: 1 - с IP, 2 - с STM (тоже IP, но его рассматриваем отдельно), 3 - с E1. Можно использовать один и тот же сервер для съема разных типов трафика IP/STM/E1.

В состав ПО «ОЛИМП-374» входят следующие компоненты:

disk	Открытый (open-source) дистрибутив ОС Linux
dppk	Открытый (open-source) пакет библиотек и драйверов
balancer	ПО модуля «Устройство съема IP»
SnifferSignCalls и SnifferSignSMS	
VariatorCalls и VariatorSMS	
SormGwCalls и SormGwSMS	
SnifferMedia	
Order573_server	
E1-SPAN	
SnifferMedia	
Balancer	ПО модуля «Устройство съема STM-1»
Odrv balancer	
SnifferMedia	
libmfstorage	ПО модуля «Шлюз-86»
gateway86 и liborder86_proto	
Converter	

3 Конфигурация сервера

Рекомендуемые аппаратные требования *				
Элемент комплекса	«Шлюз-86» 1U	УС 1U	УС 2U	УС 3U
Серверная платформа (доступный тип питания AC - 220 В, DC - 48 В)	AC	AC/DC	AC/DC	AC
Процессор	2xCPU не менее 8 физических ядер в каждом	2xCPU 14 физических ядер в каждом	2xCPU 14 физических ядер в каждом	2xCPU 14 физических ядер в каждом
Оперативная память	64 Гб	64 Гб	64 Гб	64 Гб



Рекомендуемые аппаратные требования *				
Элемент комплекса	«Шлюз-86» 1U	УС 1U	УС 2U	УС 3U
Операционная система	Debian	Debian	Debian	Debian
Сетевая плата	1) Адаптер 4x1000Base-T. 2) Адаптер 4x10GBase-LR.	2 слота PCI-E под сетевые карты (полнопрофильные). Возможна установка следующих карт: 1) Адаптер 4x1000Base-T. 2) Адаптер 4x10GBase-LR. 3) Parabel Quasar-16PCX (съём 8 операторских дуплексных потоков E1). 4) FPGA (ПЛИС) 4xQSFP - 1 шт. в платформу.	6 слотов PCI-E под сетевые карты: - 4 полнопрофильных. - 2 низкопрофильных. Возможна установка следующих карт: 1) Адаптер 4x1000Base-T - до 6 шт. в платформу. 2) Адаптер 4x10GBase-LR - до 6 шт. в платформу. 3) Parabel Quasar-16PCX (съём 8 операторских дуплексных потоков E1) - до 4 шт. в платформу. 4) FPGA (ПЛИС) 4xQSFP - 1 шт. в платформу.	11 слотов PCI-E под сетевые карты (полнопрофильные): - 10 слотов PCI-E 3.0 x8 - 1 слот PCI-E 2.0 x4 (in x8) Возможна установка следующих карт: 1) Адаптер 4x1000Base-T - до 10 шт. в платформу. 2) Адаптер 4x10GBase-LR - до 10 шт. в платформу. 3) Parabel Quasar-16PCX (съём 8 операторских дуплексных потоков E1) - до 10 шт. в платформу. 4) FPGA (ПЛИС) 4xQSFP - 1 шт. в платформу.
Кол-во и тип интерфейсов	Интегрированные: 1) 1xIPMI. 2) 2x1000Base-T. Сетевые карты: 1) Адаптер 4x1000Base-T. 2) Адаптер 4x10GBase-LR / 4x1000Base-LX.	Интегрированные: 1) 1xIPMI 2) 2x1000Base-T Сетевые карты (до 2 штук в платформу): 1) Адаптер 4x1000Base-T. 2) Адаптер 4x10GBase-LR. 3) Parabel Quasar-16PCX (съём 8 операторских дуплексных потоков E1). 4) FPGA (ПЛИС) 4xQSFP - 1 шт. в платформу.	Интегрированные: 1) 1xIPMI 2) 2x1000Base-T Сетевые карты: 1) Адаптер 4x1000Base-T - до 6 шт. в платформу. 2) Адаптер 4x10GBase-LR - до 6 шт. в платформу. 3) Parabel Quasar-16PCX (съём 8 операторских дуплексных потоков E1) - до 4 шт. в платформу. 4) FPGA (ПЛИС) 4xQSFP - 1 шт. в платформу.	Интегрированные: 1) 1xIPMI 2) 2x1000Base-T Сетевые карты: 1) Адаптер 4x1000Base-T - до 10 шт. в платформу. 2) Адаптер 4x10GBase-LR - до 10 шт. в платформу. 3) Parabel Quasar-16PCX (съём 8 операторских дуплексных потоков E1) - до 10 шт. в платформу. 4) FPGA (ПЛИС) 4xQSFP - 1 шт. в платформу.
Объём жёсткого диска	2xHDD (по 2 ТБ)	2xHDD (по 2 ТБ) 1xSSD (по 480 Гб) **	2xHDD (по 2 ТБ) 1xSSD (по 480 Гб) **	2xHDD (по 2 ТБ) 1xSSD (по 480 Гб) **

* Указаны рекомендуемые к внедрению аппаратные требования. Для отдельных случаев (малых Операторов связи при малых объемах трафика) ТТХ могут быть пересмотрены (требуется дополнительное согласование).

** Опциональная позиция. Допускается использование одного SSD-диска объемом не менее 480 Гб с интерфейсом M.2, если серверная платформа поддерживает данный тип физических накопителей.



4 Установка ПО

4.1 Установка операционной системы и предварительная настройка

1. Установка ОС и подготовка серверного оборудования в соответствии с техническими условиями, изложенными в документации.
2. Размещение архива с исполняемыми файлами ПО в корне файловой системе установленной ОС.
3. Настройка BIOS.

4.2 Установка и настройка пакетов съемника

Скопировать на сервер последний релиз ПО «ОЛИМП-374» (его расположение см. в Release Notes), и распаковать.

Каждый релиз ПО «ОЛИМП-374» включает следующие компоненты:

- dpkg_fz374_01.12.2023_debian_12.2.tar.gz
- sorm_variator_calls.tar.gz
- sorm_variator_sms.tar.gz
- sormgw.tar.gz
- sniffer.tar.gz
- install_elspan.tar.gz

4.2.1 Установка ПО для съема IP (медиа и/или сигнализации)

Для корректной установки необходимо строго соблюдать описанную ниже последовательность действий:

1. установить DPDK;
2. установить все требуемые пакеты путем выполнения скрипта install.sh

4.2.1.1 Установка и настройка DPDK

Если съем трафика на сервере будет производиться только с E1-каналов, установка DPDK не требуется.

Установка:

1. Переходим в папку **dpkg_fz374_debian_12.2/for_all_servers/dpdk**
2. Ставим DPDK: **dpkg_install install**

Настройка:

1. Выполните команду инициализации: **dpdk_utils init**;
2. Выберите сетевые устройства для привязки к драйверу DPDK;
3. Перезагрузите сервер;
4. В файле конфигурации **dpdk_cluster (/home/dpdk/dpdk_cluster/cfg/cl_daemon.ini)** настроить следующие параметры: **numa_node_num, preferred_cpu, num_buffs**.
5. Перезагрузить сервер.



4.2.1.2 Установка и настройка ПО

4.2.1.2.1 Установка directnetio/balancer, пакетов для работы снифера/вариатора/сорм-шлюза

1. Переходим в папку **dpkg_fz374_debian_12.2/for_all_servers**
2. Ставим пакеты: **./install.sh**

Данные пакеты ставим на все сервера, независимо от того, какие именно модули на них будут запускаться.

4.2.1.2.2 Настройка Directnetio

1. В файле конфигурации **/opt/directnetio/etc/directnetio.conf** необходимо отредактировать строки описания очередей. Строка должна быть следующего формата:
pipe: name size node block
2. Проверить состояние драйвера и сконфигурированных очередей: **dnettool -a**

4.2.1.2.3 Настройка balancer

Для настройки автоматического запуска балансера необходимо отредактировать следующие параметры в конфигурационном файле **/opt/balancer/etc/balancer.conf**:

- balancer: <balancer_name> (Имена процессов балансеров)
- <balancer_name>.cpu: <cpu_num> (Номер ядра)
- <balancer_name>.in: <dpdk:portX> (Входные устройства для каждого процесса балансера)
- <balancer_name>.out: dev="<device>"[,<parameter>=<value>][...] (Выходные устройства для каждого процесса балансера)
- l2_balancing_l3_type: proto="<protocol_name>", src="<src_MAC>", dst="<dst_MAC>", group="<group_name>" (Группы для балансировки L2-трафика)

Для запуска, остановки и перезапуска сервиса балансера используется команда:

systemctl start/stop/restart balancer

Проверить состояние балансера можно с помощью команды: **systemctl status balancer**

В автозагрузку сервис добавляется командой: **systemctl enable balancer**

4.2.1.2.4 Настройка съема сигнализации

Перед выполнением настройки должны быть выполнены все пункты из раздела 4.2.1.2.1 выше.

4.2.1.2.4.1 Настройка DPDK

1. До инициализации DPDK (выполнения команды **dpdk_utils_init**) необходимо проверить все сетевые интерфейсы, используемые для съема, на наличие сигнального трафика с помощью скрипта, в
dpkg_fz374_debian_12.2/for_all_servers/other/scripts_for_configuring/analyze_traffic/analyze_traffic_on_every_interface.sh
2. Выполнить **dpdk_utils_init**, привязать на нужные устройства драйвер DPDK
3. Удалить все дефолтные конфигурационные файлы из директорий **/home/dpdk/dpdk_apps/cfg** и **/home/dpdk/dpdk_cluster/cfg**, затем обновить их на типовые, в которых уже задана нужным образом большая часть параметров



4. Отредактировать параметр **mac_addr** в каждом из файлов в **/home/dpdk/dpdk_apps/cfg**, перечислив в **mac_addr** через запятую список интерфейсов
5. В **/home/dpdk/scripts/dpdk_utils_on_boot.sh** изменить для ноды с трафиком объем памяти на 4Гб. Для ноды без трафика оставить 1Гб. Далее изменить конфигурационные файлы **grub** с помощью скрипта **dpkg_fz374_debian_12.2/for_all_servers/other/scripts_for_configuring/configure_grub_cfg.sh**
6. Перезагрузить сервер
7. Запустить **dpdk_cluster** командой **systemctl dpdk_cluster start** (сервис устанавливается в систему на этапе выполнения **dpkg_fz374_debian_12.2/for_all_servers/install.sh**)
8. Добавить **dpdk_cluster** в автозагрузку: **systemctl enable dpdk_cluster**

4.2.1.2.4.2 Настройка сигнального кластера

Шаблон сигнального кластера располагается в директории **dpkg_fz374_debian_12.2/for_all_servers/other/signaling_scheme_template**

1. Необходимо скопировать его в **/ARCHIVE/SormovichVoipCluster**
2. Отредактировать файл **cluster.cfg** и внести в него актуальные значения:
 - LOCAL_IP;
 - MSHD_IP ;
 - MSHD_PORT;
 - SIGN_NODE.
3. Запустить скрипт **/ARCHIVE/SormovichVoIpCluster/modify_signaling_cluster_cfg.sh**. После выполнения, **modify_signaling_cluster_cfg.sh** и **cluster.cfg** можно удалить.
4. Переходим в **/ARCHIVE/SormovichVoipCluster/services** и выполняем скрипт **install_services.sh**, добавляя таким образом сервисы сигнальных модулей в ОС и в автозагрузку.

4.2.1.2.5 Настройка съема медиа IP

1. Настроить **dpdk** (см. раздел 4.2.1.2.4.1). Перед выполнением настройки должны быть выполнены все пункты из раздела 4.2.1.2.1 выше.
2. Настроить **directnetio** с помощью шаблона конфигурационного файла **directnetio.conf** из **/dpkg_fz374_debian_12.2/for_all_servers/other/media_scheme_template/other/directnetio_cfg**. Заменить дефолтный конфигурационный файл **directnetio.conf** в **/opt/directnetio/etc**.
3. Настроить **balancer** с помощью типовой конфигурационного файла модуля **Balancer** (**balancer.conf**) для съемника медиа IP в **dpkg_fz374_debian_12.2/for_all_servers/other/media_scheme_template/other/balancer_cfg/ip**. Заменить данным файлом дефолтный конфиг в **/opt/balancer/etc**
4. Запустить утилиту **test_balancer** и оценить объём трафика, подаваемого в каждую pipe.
5. Перезапустить балансир: **systemctl restart balancer**.

4.2.1.2.6 Генерация медиакластера

После того как настроили **dpdk**, **directnetio** и **balancer** - приступаем к развертыванию медиасниферов. Шаблон схемы, на основе которой осуществляется генерация медиакластера состоит из:



- Директория **Sniffer**, содержащая шаблоны конфигурационных файлов для IP и STM-схем, сервиса снифера, структуры папок и т.д.;
- Конфигурационный файл **cluster.cfg**;
- Скрипт **generate_media_cluster.sh**, который и осуществляет генерацию схемы в **/ARCHIVE/SormovichVoipCluster** на основе параметров, указанных в **cluster.cfg**, формирует необходимое количество сниферов, их конфигурационные файлы, создает сервисы для каждого из них, прописывает в автозагрузку, добавляет имена сервисов в скрипт для работы с кластером **/ARCHIVE/SormovichVoipCluster /action_scheme.sh** и т.д.

Конфигурационный файл **cluster.cfg** содержит следующие параметры:

- SNIFFERS_NUMBER
- LOCAL_IP
- VARIATOR_IP
- VARIATOR_CONTROL_PORT
- MSHD_IP
- MSHD_PORT
- Задаёт порт для записи в МСХД. Обычно принимает значение 3245
- CORES_NUM
- CAPTURE_DEVICES
- MAIN_STM_MACS и RESERVE_STM_MACS

Запускать скрипт необходимо напрямую из **dpkg_fz374_debian_12.2/for_all_servers/other/media_scheme_template**.

В случае если в **cluster.cfg** допущена ошибка, например, не совпадает количество значений сниферов в одном из параметров - будет выведена ошибка и отказано в генерации.

Если медиакластер был уже создан, работа скрипта также будет прервана с ошибкой.