Описание функциональных характеристик Arenadata Streaming Platform T4 (ADS T4)

Москва
2024

Содержание:

1 Аппаратные требования к серверам

Системные требования, приведенные ниже, являются минимальными. Целевой сайзинг необходимо рассчитывать, исходя из требований вашей организации.

Приоритетнее всего использовать физические серверы, но необходимо проанализировать энергопотребление предполагаемого оборудования.

Arenadata Streaming Platform T4 одинаково эффективно работает как на bare metal, так и в облаке.

При обеспечении сетевой доступности между серверами возможны также следующие сценарии:

- гетерогенная ИТ-инфраструктура;
- Multi-clouds;
- распределенная установка на разные инфраструктуры.

Примеры требований к разным сервисам в зависимости от профиля нагрузки показаны далее.

ZooKeeper, Kafka Manager, Schema Registry:

Для обеспечения HA (High Availability) необходимо, чтобы количество экземпляров сервиса ZooKeeper было нечетным.

Minimal:

Хранение	**CPU**	**RAM**	**Сеть**
400 ГБ SSD для ОС 500 ГБ SSD для данных	8 ядер	32 ГБ	1 x 10 GbE

Proof of concept:

Кол-во узлов	**Хранение**	**CPU**	**RAM**	**Сеть**
3 и более	1 x 500 ГБ HDD для ОС 1 x 2.4 ТБ SSD для журналов транзакций 1 x 1 ТБ HDD для данных	8 ядер	32 ГБ	1 x 10 GbE

Commodity:

Кол-во узлов	Хранение	CPU	RAM	Сеть
3 и более	2 x 500 ГБ HDD RAID1 для ОС 2 x 2.4 ТБ SSD RAID1 для журналов транзакций 2 x 1 ТБ HDD RAID1 для данных	8 ядер	32 ГБ	2 x 10 GbE

Balanced:

Кол-во узлов	Хранение	CPU	RAM	Сеть
3 и более	2 x 500 ГБ HDD RAID1 для ОС 2 x 2.4 ТБ SSD RAID1 для журналов транзакций 2 x 1 ТБ HDD RAID1 для данных	8 ядер	32 ГБ	2 x 10 GbE

Compute-intensive:

Кол-во узлов	Хранение	CPU	RAM	Сеть
3 и более	2 x 500 ГБ HDD RAID1 для ОС 2 x 2.4 ТБ SSD RAID1 для журналов транзакций 2 x 1 ТБ HDD RAID1 для данных	8 ядер	32 ГБ	2 x 10 GbE

Kafka Broker:

Proof of concept:

Кол-во узлов	Хранение	CPU	RAM	Сеть
3 и более	1 x 500 ГБ HDD для ОС 2 x 500 ГБ SSD RAID1 для данных	8 ядер	32 ГБ	2 x 10 GbE

Commodity:

Кол-во узлов	Хранение	CPU	RAM	Сеть
3 и более	2 x 500 ГБ HDD RAID1 для ОС 4 x 500 ГБ SAS12G SSD RAID10 (PCIe Gen3 x4+) для данных	8 ядер	32 ГБ	2 x 25 GbE

Balanced:

Кол-во узлов	Хранение	CPU	RAM	Сеть
3 и более	2 x 500 ГБ HDD RAID1 для ОС 8 x 500 ГБ SAS12G SSD RAID10 (PCIe Gen3 x8) для данных	16 ядер	64 ГБ	2 x 25 GbE

Compute-intensive:

Кол-во узлов	Хранение	CPU	RAM	Сеть
6 и более	2 x 500 ГБ HDD RAID1 для ОС 8 x 500 ГБ SAS12G SSD RAID10 (PCIe Gen3 x8) для данных	24 ядер	96 ГБ	2 x 25 GbE

NiFi:

Minimal:

Хранение	CPU	RAM	Сеть
400 ГБ SSD для ОС 500 ГБ SSD для данных	8 ядер	32 ГБ	1 x 10 GbE

Proof of concept:

Кол-во узлов	Хранение	CPU	RAM	Сеть
2 и более	1 x 500 ГБ HDD для ОС 1 x 500 ГБ SSD для данных	8 ядер	32 ГБ	1 x 10 GbE

Commodity:

Кол-во узлов	Хранение	CPU	RAM	Сеть
2 и более	2 x 500 ГБ SSD RAID1 для ОС 4 x 500 ГБ SAS12G SSD RAID10 (PCIe Gen3 x4+) для данных	12 ядер	64 ГБ	2 x 25 GbE

Balanced:

Кол-во узлов	Хранение	CPU	RAM	Сеть
2 и более	2 x 500 ГБ SSD RAID1 для ОС 8 x 500 ГБ SAS12G SSD RAID10 (PCIe Gen3 x8) для данных	24 ядер	96 ГБ	2 x 25 GbE

Compute-intensive:

Кол-во узлов	Хранение	CPU	RAM	Сеть
4 и более	2 x 500 ГБ SSD RAID1 для ОС 16 x 500 ГБ SAS12G SSD RAID10 (2 контроллера PCIe Gen3 x8) для данных	48 ядер	192 ГБ	2 x 25 GbE

ksqlDB:

Профиль нагрузки	Кол-во узлов	Хранение	CPU	RAM
Все	2 и более	SSD (объем зависит от количества запросов и агрегаций)	4 ядра	32 ГБ

REST Proxy:

Профиль нагрузки	Кол-во узлов	Хранение	CPU	RAM
Все	2 и более	HDD (без требования к скорости)	16 ядер и более (работа с producer/ consumer)	1 ГБ 64 МБ на producer 16 МБ на consumer

2 Аппаратные требования к сети

Требуются одна или несколько выделенных одноранговых Interconnect-сетей для внутренней коммуникации кластера.

К Interconnect-сетям должны быть подключены все серверы кластера.

Все серверы должны иметь полное доменное имя (FQDN) и возможность обмениваться данными друг с другом на заданных скоростях.

Должны быть известны IP-адреса каждого сервера.

Должна быть обеспечена доступность Ping любого из серверов (ICMP).

В случае инсталляции с доступом в Интернет требуется доступ к развернутому на выделенном сервере Arenadata Cluster Manager.

В случае инсталляции без доступа в Интернет (offline) требуется доступ к развернутому на выделенном сервере Arenadata Enterprise Tools.

На серверах, используемых для установки кластера, должны быть открыты порты, приведенные ниже.

Каждый порт относится к серверу, на который устанавливается конкретный сервис.

Сервис	**Порт**	**Назначение**
Общие	22	Порт для подключения к SSH-провайдеру
	81	TCP-порт Arenadata Enterprise Tools (доступ к репозиториям в случае offline-инсталляции)
	8000	TCP-порт для отправки статусов компонентов кластера в ADCM
	2015	TCP-порт для отправки метрик на сервер мониторинга
	2016	UDP-порт для отправки метрик на сервер мониторинга
ZooKeeper	2181	Порт доступа к сервису
	2888 3888	Порты межсерверного взаимодействия для компонентов кворума
Kafka	9092	HTTP-порт доступа к сервису
	9093	HTTPS-порт доступа к сервису
	9999	JMX-порт для доступа к метрикам сервиса
Kafka-Manager	9898	Порт доступа к сервису
Kafka REST Proxy	8082	Порт доступа к сервису
	9998	JMX-порт для доступа к метрикам сервиса
chema-Registry	8081	Порт доступа к сервису
	9997	JMX-порт для доступа к метрикам сервиса
ksqlDB	8088	Порт доступа к компоненту ksqlDB Server
NiFi	9090	HTTP-порт доступа к UI сервиса
	9091	HTTPS-порт доступа к UI сервиса
	11443	Порт взаимодействия между серверами NiFi, объединенными в кластер
	18080	Порт доступа к UI компонента NiFi-Registry

3 Программные требования

3.1 Операционная система

1. ADS поддерживает только архитектуру x86_64 и следующие операционные системы:

- Centos 7.6.1810;
- RHEL 7.6;
- AltLinux sp 8.4 — доступно только в Enterprise-версии;
- Astra Linux 1.7 "Орел" SE с Axiom JDK — доступно только в Enterprise-версии;
- RedOS 7.3.2 — доступно только в Enterprise-версии.

2. Необходимо подготовить все зависимые репозитории дополнительно к выбранной операционной системе.
3. Со всех машин в кластере должен быть доступ к официальным репозиториям (или создано локальное зеркало):

- CentOS Extras
- CentOS Updates
- CentOS Base

4. В случае online-установки необходимо обеспечить доступ к репозиториям, перечисленным на вкладке Configuration в меню кластера ADS в UI ADCM.
5. В случае offline-установки необходимо обеспечить доступ к хосту, на котором развернут кластер Enterprise Tools.

3.2 Предустановленное ПО

Должен быть предварительно настроен NTP на серверах.

Должен быть обеспечен доступ к пользователю *root* или любому пользователю с правами *sudo* на серверах кластера.

3.3 Особенности установки ADS

Процесс установки модифицирует файлы, которые не должны контролироваться configuration management system (при ее наличии):

- системный файл /etc/hosts;
- системный файл /etc/selinux/config;

- системный файл /etc/sysctl.conf.

Процесс установки создает файлы в /usr/lib/systemd/system/, которые не должны контролироваться configuration management system (при ее наличии).

Сервисы, устанавливаемые в кластере, не должны контролироваться configuration management system (при ее наличии).

3.4 SELinux

SELinux должен быть выключен.

3.5 Firewall

Firewall должен быть остановлен.

3.6 Браузеры, поддерживающие сервис NiFi

Работу сервиса NiFi поддерживают браузеры:

- Chrome
- FireFox
- Edge
- Safari

Пользовательский интерфейс поддерживается в текущей стабильной версии этих браузеров и в предыдущей. Например, если текущий стабильный выпуск — 45.X, то официально поддерживаемыми версиями будут 45.X и 44.X.

Для Safari, который выпускает major-обновления гораздо реже, поддерживаются только две последние версии.

Поддерживаемые версии браузеров зависят от возможностей пользовательского интерфейса и используемых им зависимостей. Функции пользовательского интерфейса разрабатываются и тестируются в поддерживаемых браузерах.

4 Рекомендации по настройке

4.1. Kafka

4.1.1. Рекомендации по настройке брокеров Kafka

После установки ADS в конфигурационном файле */usr/lib/kafka/config/server.properties* автоматически устанавливаются необходимые параметры для брокеров Kafka.

Если необходимая настройка отсутствует, для ее добавления следует воспользоваться строкой *Add key,value* в списке группы *server.properties*, где требуется написать ключ и значение в соответствующих полях. Другой вариант — в файле */usr/lib/kafka/config/server.properties* добавить соответствующую строку.

Некоторые параметры могут быть установлены непосредственно в командной строке запуска скриптов при помощи опций. Эта установка переопределяет значения по умолчанию, приведенные в */usr/lib/kafka/config/server.properties*.

Ниже приведены рекомендации для настройки некоторых параметров брокеров Kafka.

4.1.1.1 Настройка репликации топиков

Репликацию топиков определяют следующие параметры:

- *default.replication.factor* — коэффициент репликации по умолчанию для автоматически создаваемых топиков.

Высокий коэффициент репликации необходим для проектирования надежной системы. При коэффициенте репликации *2* и более Kafka реплицирует журнал для партиций каждого топика на нескольких серверах. Когда сервер выходит из строя, происходит автоматическое переключение на эти реплики, сообщения остаются доступными. Для эффективной работы системы рекомендуется установить коэффициент репликации не менее *3*.

Этот параметр может быть изменен в интерфейсе ADCM.

Существует возможность установить коэффициент репликации для каждого топика отдельно.

При необходимости возможно изменить коэффициент репликации у существующей партиции.

- *min.insync.replicas* — минимальное количество синхронизированных реплик, необходимых для разрешения запросов производителя с установленным параметром *request.required.acks.*

Когда производитель устанавливает *request.required.acks* на all (или -1), *min.insync.replicas* указывает минимальное количество реплик, которые должны подтвердить запись, чтобы запись считалась успешной. Если этот минимум не может быть соблюден, то производитель вызовет ошибку (NotEnoughReplicas или NotEnoughReplicasAfterAppend).

Этот параметр может быть задан в интерфейсе ADCM.

При совместном использовании *min.insync.replicas* и *request.required.acks* позволяют обеспечить более высокие гарантии надежности.

Успешный сценарий:

- топик с коэффициентом репликации 3;
- min.insync.replicas со значением 2;
- request.required.acks со значением all (или -1).

Такое сочетание параметров гарантирует, что производитель вызовет ошибку, если большинство реплик не получит запись.

Существует возможность установить *min.insync.replicas* для каждого топика отдельно.

4.1.1.2 Настройка партиций топиков

Параметр *num.partitions* определяет количество партиций по умолчанию для каждого создаваемого топика.

Разбиение топиков на партиции приводит к лучшей балансировке данных и способствует параллелизму потребителей. Для данных с ключами следует избегать изменения количества партиций в топике.

Этот параметр может быть изменен в интерфейсе ADCM.

Существует возможность установить количество партиций для каждого топика отдельно.

4.1.1.3 Настройка политик хранения

Для оптимизации дискового пространства необходимо установить корректную политику хранения сообщений в топике.

Политику хранения определяют следующие параметры:

- *log.cleanup.policy* — определяет политику хранения для использования в сегментах журнала. Значение по умолчанию действует для всех пользовательских топиков. Может быть изменен в интерфейсе ADCM. Возможные значения:
  - delete — отбрасывает старые сегменты топика, когда их время хранения или предельный размер были достигнуты;
  - compact — включает сжатие журнала, при котором сохраняется последнее значение для каждого ключа. Также можно указать обе политики в списке, разделенном запятыми (например, delete, compact). В этом случае старые сегменты будут отброшены в соответствии с конфигурацией времени хранения и размера, а оставшиеся сегменты будут сжаты.

Политика compact имеет смысл для топиков, где важно последнее значение для каждого ключа в топике.

- *log.retention.ms* — определяет максимальное время, в течение которого сообщения в топике будут храниться, прежде чем будут удалены старые сегменты топика, чтобы освободить место, если установлена политика хранения delete. Если установлено значение -1, ограничение по времени не применяется. Параметр может быть задан в интерфейсе ADCM.
- *log.retention.bytes* — определяет максимальный размер партиции, после достижения которого будут удалены старые сегменты журнала, чтобы освободить место, если установлена политика хранения delete. По умолчанию нет ограничений по размеру, только ограничение по времени. Поскольку это ограничение применяется на уровне партиции, умножьте его на количество партиций, чтобы вычислить размер неудаляемой части топика в байтах. Параметр может быть задан в интерфейсе ADCM.

Существует возможность установить данные параметры для каждого топика отдельно.

4.1.1.4 Настройка корректного завершения работы

Настройка корректного завершения работы важна для надежности системы.

Параметр *controlled.shutdown.enable* со значением true включает управляемое отключение сервера. Параметр может быть задан в интерфейсе ADCM.

При выходе сервера из строя или намеренного отключения для обслуживания или изменения конфигурации кластер Kafka автоматически обнаружит любое отключение или сбой брокера и выберет новых лидеров для партиций на этой машине.

Когда сервер корректно останавливается, он использует две оптимизации:

1. Синхронизация всех журналов с диском, чтобы избежать необходимости восстанавливать журналы при перезапуске (т.е. проверять контрольную сумму для всех сообщений в конце журнала). Восстановление журнала требует времени, поэтому это ускоряет преднамеренные перезапуски.
2. Перенос всех партиций, для которых сервер является ведущим, на другие реплики перед выключением. Это ускорит передачу лидерства и сведет к минимуму время недоступности каждой партиции до нескольких миллисекунд.

Обратите внимание, что контролируемое завершение работы будет успешным только в том случае, если все партиции, размещенные на брокере, имеют реплики (т.е. параметр *default.replication.factor* имеет значение 2 и более).

4.1.1.5 Настройка баланса лидерства

Всякий раз, когда брокер останавливается или аварийно завершает работу, управление партициями этого брокера передается другим репликам. После перезапуска брокера он будет ведомым только для всех своих партиций, то есть не будет использоваться для операций чтения и записи клиента. Чтобы избежать этого дисбаланса, в Kafka есть понятие предпочтительных реплик. Если список реплик для партиции равен 1, 5, 9, то узел 1 предпочтительнее в качестве лидера по сравнению с узлом 5 или 9, поскольку он находится раньше в списке реплик.

Параметр *auto.leader.rebalance.enable* со значением true включает автоматическую балансировку лидера (возврат лидерства предпочтительной реплике) в фоновом режиме через регулярные промежутки времени. Параметр может быть изменен в интерфейсе ADCM.

Если для этого параметра установить значение false, необходимо вручную восстановить лидерство в восстановленных репликах после перезагрузки сервера. Для этого необходимо выполнить команду:

```
$ /usr/lib/kafka/bin/kafka-leader-election.sh --bootstrap-server localhost:9092 --election-type preferred --all-topic-partitions
```

4.1.2. Рекомендации по работе с производителями и потребителями

4.1.2.1 Настройка получения производителем подтверждения записи сообщения

Параметр *request.required.acks* определяет, когда запрос на производство считается выполненным. В частности, сколько других брокеров должны зафиксировать данные в своем журнале и подтвердить это лидеру. Возможные значения:

- 0 — производитель никогда не ждет подтверждения от брокера. Этот вариант обеспечивает наименьшую задержку, но гарантирует наименьшую надежность (некоторые данные будут потеряны при сбое сервера).
- 1 — производитель получает подтверждение после того, как ведущая реплика получит данные. Это значение по умолчанию.
- -1 — производитель получает подтверждение после того, как все синхронизированные реплики получили данные. Этот вариант обеспечивает наилучшую устойчивость.

Совместное использование параметров *min.insync.replicas* и *request.required.acks* позволяют обеспечить более высокие гарантии надежности.

4.1.2.2 Проверка позиции потребителя

При необходимости при помощи скрипта *kafka-consumer-groups.sh* возможно увидеть позицию потребителей топика.

4.1.2.3 Назначение топикам групп потребителей

Каждое сообщение отправляется каждой группе потребителей, подписавшейся на топик или партицию, но внутри группы оно отправляется только одному потребителю. Таким образом, все группы потребителей, подписавшиеся на топик, получают сообщения, но только один потребитель в группе потребителей получает сообщение из партиции.

Если необходимо передать сообщение нескольким потребителям, следует назначить им разные группы потребителей.

4.1.3. Настройка безопасной среды

Для получения информации о настройке безопасности в среде Kafka можно перейти к разделу *Базовые операции → Kafka → Управление доступом*.

4.1.3.1 Аппаратное обеспечение

Общие рекомендации при выборе оборудования:

- При возможности использовать для брокеров Kafka отдельные сервера или стойки.
- Для применения оптимального коэффициента репликации иметь в системе не менее 3 брокеров Kafka.
- Обеспечить HA (High Availability) в соответствии с требованиями, согласно планируемой нагрузке.

Рекомендации при выборе памяти:

- Kafka при работе может нуждаться в большом обьеме файловой системы для хранения и кеширования сообщений и мало использует heap-пространство.
- Kafka при работе может нуждаться в достаточной памяти для буферизации активных операций чтения и записи.

Рекомендации при выборе процессоров:

- Включение защиты канала по протоколу SSL может увеличить требования к процессору (точные данные зависят от типа ЦП и реализации JVM).
- При выборе процессора между более быстрыми процессорами или большим количеством ядер, следует выбрать большее количество

ядер. Дополнительный параллелизм, который предлагает несколько ядер, предпочтительнее более высокой тактовой частоты.

Хранение данных:

- Не рекомендуется использовать массив дисков JBOD, так как для многоуровневого хранилища требуется одна точка монтирования.
- Рекомендуется использовать несколько дисков, чтобы максимизировать пропускную способность.
- Не рекомендуется использовать одни и те же диски, используемые для данных Kafka, совместно с журналами приложений или другой активностью файловой системы ОС.
- Если вы настраиваете несколько каталогов данных, брокер помещает новую партицию по пути с наименьшим количеством сохраненных партиций. Каждая партиция будет полностью находиться в одном из каталогов данных. Если данные не сбалансированы между партициями, это может привести к дисбалансу нагрузки между дисками.
- Массив дисков RAID потенциально может лучше справляться с балансировкой нагрузки между дисками, потому что балансирует нагрузку на более низком уровне.
- Массив дисков RAID 10 рекомендуется как лучший вариант для большинства случаев использования. Он обеспечивает улучшенную производительность чтения и записи, защиту данных (способность выдерживать сбои диска) и быстрое восстановление.
- Основным недостатком RAID является то, что он уменьшает доступное дисковое пространство. Другим недостатком является стоимость операций ввода-вывода при восстановлении массива при сбое диска. Стоимость восстановления относится к RAID в целом, с учетом нюансов между различными версиями.
- Не рекомендуется использовать сетевые хранилища (Network Attached Storage, NAS). NAS часто медленнее, показывает большие задержки с более широким отклонением средней задержки и является единой точкой отказа.

Сеть:

Требования к сетевым подключениям приведены в главе 2. Низкая задержка гарантирует, что узлы могут легко обмениваться данными, а высокая пропускная способность помогает перемещать и восстанавливать

сегменты. Современные сети центров обработки данных (1 GbE, 10 GbE) достаточны для подавляющего большинства кластеров.

4.1.3.2 Программное обеспечение

Общие рекомендации:

- Требования к программному обеспечению приведены в главе 3.
- Для запуска Kafka рекомендуется использовать файловую систему XFS или ext4.
- Используйте последнюю выпущенную версию Java, чтобы убедиться, что известные проблемы безопасности устранены.
- Рекомендуемая настройка GC (проверенная на большом развертывании с JDK 1.8 u5) выглядит следующим образом:

```
-Xms6g -Xmx6g -XX:MetaspaceSize=96m -XX:+UseG1GC -XX:MaxGCPauseMillis=20
       -XX:InitiatingHeapOccupancyPercent=35 -XX:G1HeapRegionSize=16M
       -XX:MinMetaspaceFreeRatio=50 -XX:MaxMetaspaceFreeRatio=80
```

- Приведенные ниже рекомендации могут быть полезны для улучшения совместимости между Kafka и вашей операционной системой. Для лучшего применения рекомендаций необходимо обратиться к документации по конкретному дистрибутиву.

Файловые дескрипторы:

Kafka в любой момент потенциально может требовать относительно большого количества доступных файловых дескрипторов.

Многие современные дистрибутивы Linux поставляются только с 1024 файловыми дескрипторами, разрешенными для каждого процесса. Это недостаточно для Kafka.

Необходимо увеличить количество файловых дескрипторов как минимум до 100,000, а возможно, и намного больше.

Для изменения лимита открытых файлов необходимо отредактировать */etc/security/limits.conf* — файл, устанавливающий лимиты ресурсов для пользователей, вошедших в систему через PAMPrivileged Access Management (PAM). Для этого требуется для всех групп пользователей увеличить значение параметра nofile — максимальное число открытых файлов для типов hard и soft, добавив в */etc/security/limits.conf* строки:

```
*  hard  nofile  100000
*  soft  nofile  100000
```

Виртуальная память:

Может потребоваться изменить предельный размер виртуальной памяти (vm.max_map_count), позволяющий обрабатывать необходимое количество функций mmap.

Чтобы рассчитать текущее количество mmap, необходимо узнать количество файлов *.index* в каталоге данных Kafka. Файлы *.index* представляют собой большинство файлов с отображением памяти.

Подсчитать файлы *.index* можно при помощи этой команды:

```
$ find . -name '*index' | wc -l
```

Установите vm.max_map_count в файле */etc/sysctl.conf* для сеанса. Это установит текущее количество отображаемых в память файлов. Минимальное значение лимита *mmap (vm.max_map_count)* — это количество открытых файлов ulimit.

Вы должны установить *vm.max_map_count* большим, чем количество файлов *.index*, чтобы учесть рост сегмента брокера.

```
$ sysctl -w vm.max_map_count=262144
$ echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
$ sysctl -p
```

4.2. NiFi

Типовые настройки для Linux могут не подходить для приложений с интенсивным вводом-выводом, таких как NiFi. Для используемых областей операционной системы требования дистрибутива могут различаться.

Приведенные ниже рекомендации могут быть полезны для улучшения совместимости между NiFi и вашей операционной системой.

Для лучшего применения рекомендаций необходимо обратиться к документации по конкретному дистрибутиву.

4.2.1. Увеличение количества файловых дескрипторов

NiFi в любой момент потенциально может иметь очень большое количество открытых файловых дескрипторов.

Дескрипторы файлов — это целочисленные идентификаторы, определяющие структуры данных. Ядро Linux обращается к этим структурам как к файловым структурам, поскольку они описывают

открытые файлы. Индекс файловой структуры является дескриптором файла.

Для изменения лимита открытых файлов необходимо отредактировать */etc/security/limits.conf* — файл, устанавливающий лимиты ресурсов для пользователей, вошедших в систему через PAMPrivileged Access Management (PAM). Для этого требуется для всех групп пользователей увеличить значение параметра nofile — максимальное число открытых файлов для типов hard и soft, добавив в */etc/security/limits.conf* строки:

```
*  hard  nofile  50000
*  soft  nofile  50000
```

Необходимо при увеличении лимита учитывать размер памяти ядра операционной системы.

4.2.2. Увеличение количества разветвленных процессов

NiFi может быть настроен на создание значительного количества потоков. Чтобы увеличить допустимое число, также необходимо отредактировать файл */etc/security/limits.conf*. Для этого требуется для всех групп пользователей увеличить значение параметра nproc — максимальное число процессов для типов hard и soft, добавив в */etc/security/limits.conf*строки:

```
*  hard  nproc  10000
*  soft  nproc  10000
```

И также при наличии в вашей операционной системе необходимо отредактировать файл */etc/security/limits.d/90-nproc.conf* или */etc/security/limits.d/20-nproc.conf*, переопределяющий значение nproc, заданное в файле */etc/security/limits.conf*. В этом файле необходимо для всех групп пользователей увеличить значение параметра nproc — максимальное число процессов для типа soft, добавив строку:

```
*  soft  nproc  10000
```

4.2.3. Увеличение количества доступных портов сокетов TCP

Увеличение количества доступных портов сокетов TCP особенно важно, если ваш поток будет устанавливать и отключать большое количество сокетов за небольшой период времени.

Для установки максимального количества сокетов TCP необходимо выполнить команду:

```
$ sudo sysctl -w net.ipv4.ip_local_port_range="10000 65000"
```

4.2.4. Установка длительности состояния TIMED_WAIT для сокетов

TIME_WAIT – это состояние серверного сокета, в котором он выдерживает таймаут, чтобы собирать случайно задержавшиеся в сети пакеты. Длительность таймаута можно изменять для получения возможности быстро настраивать и отключать новые сокеты.

Для изменения длительности таймаута используются следующие команды:

- для дистрибутивов с ядром 2.6:

```
$ sudo sysctl -w net.ipv4.netfilter.ip_conntrack_tcp_timeout_time_wait="1"
```

- для дистрибутивов с ядром 3.0:

```
$ sudo sysctl -w net.netfilter.nf_conntrack_tcp_timeout_time_wait="1"
```

4.2.5. Отключение параметра подкачки

Параметр подкачки (swap) влияет на предпочтение ядра перемещать страницы памяти из приложений на страницу подкачки, а не высвобождать память из кеша.

Для улучшения эффективности NiFi рекомендуется установить параметр swappiness равным нулю. Это гарантирует, что если память ограничена, кеш страниц будет уменьшен в попытке восстановить память до того, как страницы приложения будут перемещены в пространство подкачки.

Чтобы отключить параметр подкачки, необходимо отредактировать */etc/sysctl.conf* — файл, предназначенный для управления параметрами ядра.

Необходимо добавить следующую строку:

```
vm.swappiness = 0
```

Для разделов, обрабатывающих различные репозитории NiFi, отключите параметр atime — поле для записи метки времени. Это может вызвать неожиданный скачок пропускной способности. Для этого в файле */etc/fstab* (который используется для настройки параметров монтирования различных блочных устройств, разделов на диске и удаленных файловых систем) необходимо добавить для интересующих разделов параметр noatime, отключающий полностью запись времени доступа к файлу.

4.2.6. Увеличение ресурсов аппаратного обеспечения

Для эффективной работы NiFi рекомендуется:

- При возможности использовать для экземпляров NiFi отдельные сервера или стойки.
- Обеспечить HA (High Availability) в соответствии с требованиями, описанными в главе 1.