

Проект Asterisk Management System



Николай Шестаков

Asterisk Management System (AMS) – программное обеспечение с открытым исходным кодом для конфигурации, мониторинга, биллинга Asterisk PBX, включает дружелюбный веб-интерфейс, написанный на PHP и JavaScript, и несколько модулей на C.

Проект родился из практической задачи, когда для организации с объемом звонков около 150 000 минут в месяц и примерно 200 абонентов потребовалось обеспечить биллинг, запись всех разговоров и прием факсов с сохранением их в базе данных и последующей отправкой на электронную почту (функция Fax-to-Email).

Необходимо было оформить все это в дружелюбном, удобном интер-

фейсе, а также обеспечить разграничение доступа к различным функциям. Разграничение доступа должно было быть гибким. Например, руководитель должен иметь доступ к различным биллинговым отчетам, архивам записей и факсам, секретарю и телефонисту не нужно иметь возможность прослушивать записанные разговоры. Администратор системы, кроме того, должен иметь доступ к функции редактирования конфигурацион-

ных файлов, к интерфейсу командной строки Asterisk, просмотру лог-файлов или модулю, позволяющему просматривать и редактировать содержимое внутренней базы данных Asterisk. И, конечно, программное обеспечение должно быть локализовано на русском языке.

Тестирование существующих Open Source-биллинговых программ и веб-интерфейсов (A2Billing, MOR, AstBill и других) показало, что по крайней ме-

ре в части биллинга нет подходящих проектов, а для решения других задач можно кое-что использовать, но это потребовало бы соединения из кусочков различных проектов, что зачастую сложнее, чем написать все с нуля.

С программной точки зрения AMS построена по модульному принципу, что позволяет легко адаптировать ее под конкретную задачу, а также с перспективой на то, что, будучи Open Source-системой, разработчики могут добавлять свои модули.

Интересной особенностью AMS является то, что она целиком построена с использованием AJAX-технологии. Это позволило не только ускорить работу интерфейса, но и дало разработчику возможность построить полноценное веб-приложение, а также реализовать такие функции, как запоминание истории команд (и результатов их вывода) AsteriskCLI, с возможностью их просмотра. Но в основном это сделано с прицелом на будущее развитие системы – предполагается реализовать операторскую панель с функциональностью call-центра (об этом в конце статьи).

В настоящий момент в качестве базы данных используется MySQL (в дальнейшем предполагается адаптировать под PostgreSQL и, возможно, Oracle).

Перейдем к изучению основных функций AMS.

Биллинг

Биллинг в AMS – это post-paid-биллинг (т.е. подсчитывающий стоимость звонка по факту его прохождения), ориентированный на корпоративного пользователя и обеспечивающий:

- Работу с несколькими валютами, т.к., например, VoIP-провайдеры предпочитают давать цены в долларах или евро, а местные телефонные операторы в рублях, т.е. необходимо при подсчете суммарных итогов приводить к одной (базовой) валюте.
- Работу с различными системами тарификаций. Для примера, VoIP-провайдер может дать посекундную тарификацию, телефонный оператор – поминутную, а мобильный оператор при звонках через GSM-шлюз тарификацию 60+1 (посекундная после первой минуты).
- Вывод разнообразных отчетов, включая CDR (call detail records), и различные суммирующие отчеты за выбранный период. Можно, например, получать суммарные отчеты по направлениям звонков, по тарифным планам, по подразделениям, отделам, отдельным абонентам, графические отчеты по месяцам, по дням. Дополнительно отчеты можно отфильтровать, например, по длительности звонка, стоимости, CallerID.

Для корректной работы биллинга необходимо скомпилировать и установить модуль `ams_cdr_mysql.so` из пакета `ams-modules`. Этот модуль выполняет всю необходимую работу по записи CDR, включая тарификационную информацию, в базу данных.

После установки нужно в файле `ams.conf` в секции `[global]` установить

параметры соединения с базой данных MySQL (`hostname`, `dbname`, `user`, `password`, `port`, `sock`).

Параметры, указанные в `ams.conf`, и параметры соединения в файле `config.php` должны соответствовать одной базе данных.

Установите параметры в секции:

```
[billing]

billing_enable=1
tables_in_memory=1
; 0-используются обычные таблицы, 1- необходимые для работы
; биллинга таблицы копируются в память, что позволяет
; ускорить время выборки примерно на 30%. Необходимо помнить,
; что в этом случае требуется перезагрузить конфигурацию
; Asterisk после внесения изменений в тарифы или тарифные
; планы для того, чтобы эти изменения вошли в силу
```

Биллинг работает следующим образом.

При создании тарифных планов (модуль «Тарифные Планы») задается уникальный идентификатор – `AccountCode`.

Для того чтобы биллинговая система правильно посчитала стоимость звонка, соответствующего какому-либо тарифному плану, необходимо корректно установить переменную CDR (`accountcode`) в процессе обработки данного звонка Asterisk. Если в момент завершения звонка переменная CDR(`accountcode`) будет соответствовать `AccountCode` тарифного плана (`AccountCode` может либо полностью совпадать, либо являться подстрокой CDR (`accountcode`)), звонок будет посчитан по тарифам указанного тарифного плана.

Например, если создан тарифный план `Test Plan` с `AccountCode = test`.

```
extensions.conf

exten => 100,1,Set(CDR(accountcode)=test)
; Этот звонок будет соответствовать тарифному плану
; Test Plan

exten => 200,1,Set(CDR(accountcode)=test-123)
; Этот тоже
```

Тарифные планы могут быть заданы в различных валютах. Валюты можно добавлять, редактировать курсы в модуле «Валюты».

При создании тарифных планов можно задать правила трансляции префиксов. Эти правила нужны, когда пользователь набирает номер в одном формате, а в тарифном плане используется другой формат записи.

Например, в тарифном плане задан тариф «7495 – Russia Moscow», а пользователь набирает в привычном формате «8495...».

Что планируется в новых версиях

В разработке находятся модули – «Конференция», «Мониторинг каналов», «Операторская панель».

Для реализации данных модулей потребовалось обеспечить надежный механизм постоянного коннекта с Asterisk для мониторинга текущих событий. Механизм, подобный реализованному в AsteriskNow, через AJAX-запросы, я считаю не самым удачным. Для постоянного длительного соединения лучше всего использовать со-

кет. Поскольку сокет невозможно реализовать на JavaScript, был написан flash-скрипт с использованием XMLSocket. А далее скомпилирован маленький swf-файл. Данный механизм успешно протестирован совместно с AsteriskManagerProxy.

В ближайшее время также будет доступна для скачивания версия с pre-paid-биллингом, позволяющая создать полноценную платформу для работы с телефонными картами или SIP (IAX)-клиентами.

Тогда в тарифном плане нужно задать правило трансляции: исходный префикс – 8, заменяющий префикс – 7.

Если поле, заменяющее префикс, оставить пустым, исходный префикс будет «отрезан».

Правила трансляции никак не влияют на прохождение звонка и не меняют набранные цифры, они нужны только для правильного поиска направления в тарифном плане.

Параметр «Тарификация» в тарифном плане влияет на алгоритм подсчета стоимости звонка.

Первое поле – нетарифицируемые секунды, второе – минимальная биллинговая длительность, третье – биллинговый шаг после превышения минимальной биллинговой длительности. Например, посекундная тарификация с первой секунды – 0/1/1.

Тарификация 0/30/6 означает, что звонок длительностью меньше 30 секунд будет приравнен к 30 секундам, а далее длительность будет округляться с точностью 6 секунд в сторону большего значения.

В модуле «Тарифы» можно добавлять, редактировать, удалять тарифы для выбранного тарифного плана.

При добавлении и редактировании удобно использовать «Справочник кодов», откуда можно непосредственно вставлять направления в тарифный план.

При записи cdr-звонка определяется его принадлежность тарифному плану и находится соответствующее направление в данном тарифном плане. Вначале ищется направление с максимальной длиной совпадения префикса.

Например, если в тарифном плане есть записи «Russia proper – 7» и «Russia Moscow – 7495», то звонок 7495... будет определен как Russia Moscow.

Также каждый тариф имеет параметры – минимальная длина (количество цифр) и максимальная длина. То есть звонки с одинаковыми префиксами, но с разной длиной могут быть определены как разные направления.

Например, есть тарифы test1 с префиксом 123 и длиной 4-4 и тариф test2 с префиксом 123 и длиной 5-10. Тогда звонок с набранными цифрами 1235 будет определен как test1, звонок с набранными цифрами 12356 – как test2.

Если не удалось определить принадлежность звонка ни к одному тарифному плану или в тарифном плане не найдено соответствующего тарифа, в поле «Направление» будет записано Unknown и стоимость звонка 0.00.

Нужно добавить также, что в дальнейшем была реализована версия pre-paid-биллинга (по заказу интернет-провайдера для работы со своими абонентами) со всей необходимой атрибутикой – проверкой баланса, максимальной длительностью звонка по данному направлению и т. д.

Система записи

Этот модуль позволяет осуществлять поиск записей по дате, номеру, CallerID, прослушивать записи, удалять, загружать на локальный компьютер, отправлять по e-mail.

Записи должны храниться в директории /var/spool/asterisk/monitor (задается переменной \$monitor_dir в файле config.php) в формате WAV.

Ниже приводятся два варианта конфигурации:

Вариант 1. С использованием стандартного приложения MixMonitor:

extensions.conf

```
; Для входящих звонков
exten => _X.,1,Gosub(call-monitor,${EXTEN},1)
exten => _X.,2,.....

; Для исходящих звонков
exten => _X.,1,Gosub(call-monitor,${EXTEN},outgoing)
exten => _X.,2,.....

[call-monitor]

exten => _X.,1,Set(CALLDIRECTION=i)
exten => _X.,n,Goto(m1)
exten => _X.,n(outgoing),Set(CALLDIRECTION=o)
exten => _X.,n(m1),Set(TZONE=YECT)
exten => _X.,n,Set(RECCALLERID=${CALLERID(number)})
exten => _X.,n,GotoIf(${RECCALLERID}?m2)
exten => _X.,n,Set(RECCALLERID=000)
exten => _X.,n(m2),_J
Set(RTIME=${STRFTIME(|${TZONE}|%F-%H:%M:%S)})
exten => _X.,n,Set(RECDIR=${RTIME:0:10}/${RTIME:11:2})
exten => _X.,n,Set(RECNAME=${RTIME}-${RECCALLERID}-_J
${EXTEN}-${CALLDIRECTION}-${UNIQUEID}.WAV)
exten => _X.,n,Set(TMPFILE=/tmp/monitor/${RECNAME})
exten => _X.,n,System(/bin/mkdir _J
/var/spool/asterisk/monitor/${RECDIR} -p -m 0757)
exten => _X.,n,MixMonitor(${TMPFILE}|b|/bin/mv _J
${TMPFILE} /var/spool/asterisk/monitor/${RECDIR})
exten => _X.,n,Return()
```

Вариант 2. С использованием приложения CMonitor из пакета ams-modules:

```
; Для входящих звонков
exten => _X.,1,CMonitor(i)
exten => _X.,2,.....

; Для исходящих звонков
exten => _X.,1,CMonitor(o)
exten => _X.,2,.....
```

Преимуществом CMonitor является простота, а также возможность использовать правила записи, позволяющие фильтровать записываемые звонки по префиксу и CallerID.

Приложение CMonitor пишет только в формате wav49 и начинает писать только после установки соединения.

В конфигурационном файле ams.conf нужно настроить параметры в секции:

```
[cmonitor]

use_rules=1
; 0 - пишутся все звонки, 1- пишутся звонки,
; соответствующие заданным правилам
tmp_dir=/tmp/monitor
; Директория, в которую пишется файл до момента
; окончания звонка
tables_in_memory=1
; Параметр имеет смысл, когда use_rules=1,
; 0 - используются обычные таблицы,
; 1- таблица с правилами записи копируется в память,
; что позволяет ускорить время выборки
```

Электронный факс

Модуль позволяет просматривать в PDF-формате, удалять, загружать факсы, принятые Asterisk.

Также можно осуществлять поиск по дате и времени, номеру и CallerID.

Факсы должны храниться в формате tif в директории /var/spool/asterisk/faxes (можно изменить директорию, отредактировав параметр \$faxes_dir в файле config.php).

Пример конфигурации приведен ниже:

extensions.conf

```
[fax-in]

exten => X,1,Answer
exten => X,n,Wait(1)
exten => X,n,Gosub(set-fax-vars,${EXTEN},1)
exten => X,n,Set(EMAILADDR=${DB(fax/${EXTEN})})
exten => X,n,GotoIf(${EMAILADDR}?getfax)
exten => X,n,Set(EMAILADDR=default@address.com)
exten => X,n(getfax),rxfax(${FAXFILE})

exten => h,1,System(/usr/local/bin/mail ${FAXFILE} \
    ${EMAILADDR} ${DATETIME} ${CALLERID(number)} \
    ${REMOTESTATIONID} ${FAXPAGES} ${FAXBITRATE} \
    ${FAXRESOLUTION})

[set-fax-vars]

exten => X,1,Set(TZONE=YECT)
exten => X,n,Set(FAXCALLERID=${CALLERID(number)})
exten => X,n,GotoIf(${FAXCALLERID}?ml)
exten => X,n,Set(FAXCALLERID=000)
exten => X,n(ml), \
    Set(FTIME=${STRFTIME(, ${TZONE}, %F-%H:%M:%S)})
exten => X,n,Set(FAXDIR=${FTIME:0:10}/${FTIME:11:2})
exten => X,n,Set(FAXFILE=/var/spool/asterisk/faxes/ \
    ${FAXDIR}/${FTIME}-${FAXCALLERID}-${EXTEN}- \
    ${UNIQUEID}.tif)
exten => X,n,System(/bin/mkdir \
    /var/spool/asterisk/faxes/${FAXDIR} -p -m 0757)
exten => X,n,Return
```

Менеджер файлов

Рассмотрим функции, предоставляемые менеджером файлов:

1. Копирование, перемещение, удаление, переименование файлов и папок. По умолчанию заданы рабочие директории /var/lib/asterisk как 'Asterisk Libs' и /var/lib/asterisk/sounds как 'Asterisk Sounds'. Вы можете отредактировать параметр \$filemanager_dirs в файле config.php. Веб-сервер должен иметь соответствующие права доступа к файлам в этих папках. Для этого можно выполнить следующую команду:

```
shell> chown -R nobody:nobody /var/lib/asterisk
```

Вместо nobody подставьте имя пользователя, от имени которого работает веб-сервер.

2. Просмотр и редактирование (в зависимости от типа файла) текстовых, pdf и других файлов.
3. Получение информации о файле. Для этого задержите курсор мыши на имени файла до появления всплывающего окна с детальной информацией о данном файле.
4. Прослушивание mp3- и wav-файлов.
5. Закачивание локальных и удаленных файлов на сервер.
6. Прослушивание голосовых файлов на телефонном аппарате. В сервисном окне вы можете задать протокол (SIP, IAX, Zap) и номер сервисного телефона, на который Asterisk дозванивается и проигрывает указанный голосовой файл.
7. Запись голосовых файлов в выбранную директорию с указанного сервисного телефона.

Примечание: при установке AMS файл install/ams_ext.conf копируется в /etc/asterisk и в файл extensions.conf добавляется строка #include «ams_ext.conf». Если вы устанавливали AMS вручную или по каким-либо причинам этого не было сделано, скопируйте файл install/ams_ext.conf в /etc/

asterisk и отредактируйте extensions.conf вручную. Это необходимо для корректной работы пунктов 6 и 7.

Asterisk DB

Позволяет редактировать и добавлять новые значения во внутреннюю базу данных Asterisk, а также осуществлять поиск в базе.

AsteriskCLI

Модуль дает возможность выполнять команды Asterisk, аналогично как из командной строки Asterisk.

Сохраняется история команд и результатов их выполнения (до 50). Кликая мышкой по стрелкам влево-вправо, можно просматривать историю команд.

Конфигурационные файлы

Файлы конфигурации Asterisk (обычно хранятся в папке /etc/asterisk) можно открывать для редактирования, сохранять и выводить на печать. В момент открытия файла сохраняется его копия (файл с расширением conf~). Это дает возможность отката на предыдущий вариант, если что-либо пошло не так после редактирования и сохранения. Для удобства работы в правом верхнем углу есть графическая кнопка для перезагрузки конфигурации Asterisk.

Немного о том, как в AMS реализовано разграничение доступа. Модули в AMS подразделяются на root-модули, административные и обычные. Как понятно из названия, root-модули – это модули, к которым имеет доступ только суперпользователь. В настоящий момент в списке root-модулей только один модуль Modules используется для конфигурации других модулей. Суперпользователь в системе может быть только один, он создается при установке системы и отличается от администратора только тем (кроме доступа к root-модулям), что его нельзя удалить.

Администраторы имеют доступ к административным и обычным модулям. Административные модули не видны обычным пользователям.

Модули также имеют три уровня доступа – полный, просмотр и запрещено. Уровень «просмотр» подразумевает запрет на редактирование. Например, пользователь может иметь возможность просматривать конфигурационные файлы без возможности редактирования.

Также была реализована возможность задать индивидуальные уровни доступа для конкретного модуля (на перспективу). Например, по желанию заказчика можно реализовать разграничение доступа на уровне отдела или подразделения.

Уровни доступа задаются либо индивидуально для каждого пользователя, либо можно создать роли с определенным набором уровней и присваивать эти роли пользователям.

Заключение

В настоящий момент система используется в нескольких коммерческих проектах и успешно эксплуатируется около года. На <http://www.amrex.ru> доступна для свободного скачивания последняя версия AMS (для версии Asterisk 1.4). Лицензия – GNU GPL. На сайте также можно ознакомиться с демо-версией. 