

ASTERISK И LINUX: МИССИЯ IP-ТЕЛЕФОНИЯ ДЕЙСТВИЕ 3

Соединив нашу мини-АТС с «городом» [1], мы уже получили привлекательное решение, способное обеспечить связью сотрудников небольшого офиса. Но не будем останавливаться на достигнутом – рассмотрим несколько веб-интерфейсов, значительно облегчающих работу как пользователей, так и администраторов Asterisk.

МИХАИЛ ПЛАТОВ

Послушаем почту?

Как вы помните, для проверки голосовой почты в нашем номерном плане есть специальный номер – 8500. Позволив на него и введя личный пароль, наши пользователи могли прослушать оставленные им сообщения. Правда, иногда такой способ доступа бывает несколько неудобен (например, вы находитесь в другом городе или просто под рукой нет телефона). Для решения этой проблемы можно отсылать записанные сообщения на e-mail (см. [2]) или организовать централизованный доступ через Интернет с использованием безопасного SSL-соединения. Отрадно то, что в состав стандартного дистрибутива Asterisk веб-интерфейс для работы с голосовой почтой уже входит, просто в целях безопасности автоматически он не устанавливается. Что, впрочем, не мешает нам сделать это самостоятельно. Для работы данного интерфейса необходим веб-сервер apache с пакетом perl-suid. Краткости ради будем считать, что эти пакеты у нас уже установлены и работают, а веб-сервер сконфигурирован так, что cgi-bin расположен в /var/www/cgi-bin, и cgi-скрипты, входящие в стандартную поставку, выполняются без проблем. Итак, перейдем в каталог /usr/src/asterisk и выполним команду make webvmail. Программа make скопирует все необходимые файлы и выдаст следующее предупреждение:

```
+----- Asterisk Web Voicemail -----+
+
+ Asterisk Web Voicemail is installed in
+ your cgi-bin directory.  IT USES A SETUID
+ ROOT PERL SCRIPT, SO IF YOU DON'T LIKE
+ THAT, UNINSTALL IT!
+
+-----
```

Если использование `setuid` вас устраивает, проигнорируйте это сообщение и перейдите к следующему шагу. В противном случае вы можете воспользоваться специальным патчем для Asterisk, описанным в [3].

Отредактируем файл скрипта `vmail.cgi`, определив следующее значение для контекста по умолчанию:

```
context="office"
```

Откроем ваш любимый браузер и введем URL только что установленного интерфейса (в моем случае это будет `http://ast-test/cgi-bin/vmail.cgi`). Если все было сделано правильно, то мы увидим следующее (см. **рис. 1**).

Введя имя пользователя (номер телефона) и пароль, мы попадем в папку «Входящие» нашего ящика (см. **рис. 2**).

Пользователей linux-систем придется немного расстроить – воспроизведение в браузере через плагин (кнопка

play) скорее всего не заработает, и для прослушивания сообщений придется загружать файлы на компьютер (кнопка download), а вот с Windows все нормально (см. рис. 3).

При организации работы из Интернета рекомендуется ограничить доступ к странице средствами веб-сервера, а также настроить использование SSL (https://) для безопасной передачи паролей.

Виртуальный «круглый стол»

Возможности современных систем телефонии не ограничиваются ведением разговоров между двумя абонентами. С помощью специального режима – конференц-связь – возможно организовать одновременное общение нескольких человек. Системы разного «калибра» отличаются максимально допустимым количеством конференций, а также числом участников в них. И если в недорогих офисных мини-АТС количество собеседников, как правило, невелико (не больше 8-10 человек), то в Asterisk их максимальное число ограничивается лишь аппаратными возможностями используемого сервера. Существует как минимум 3 известных модуля, предоставляющих возможности конференций для Asterisk:

- MeetMe
- Conference
- MeetMe2

MeetMe является исторически первым приложением, предоставившим возможности конференций. Модуль является достаточно функциональным и позволяет решать широкий класс задач. Несмотря на то что MeetMe входит в стандартную поставку Asterisk, для его работы необходимы высокоточные таймеры, присутствующие в аппаратуре Digium. Модуль conference же, напротив, не требует наличия zaptel-устройств, однако он предоставляет несколько меньшую функциональность. Отличительной особенностью conference является более эффективное использование ресурсов сервера, что позволяет обеспечивать одновременную работу большого числа пользователей. MeetMe2 является переработанной версией MeetMe. Наиболее значимые отличия – хранение конфигурации в СУБД и несколько другие возможности управления.

Все модули имеют схожую схему использования. После звонка по определенному номеру абонент попадает в «виртуальную комнату». Как только в «комнате» окажутся хотя бы два участника, начнется разговор. В конференциях существует 3 типа пользователей: «слушатели», «ораторы» и «администраторы». «Слушатели» лишены права голоса, «ораторы» могут не только слушать, но и разговаривать. На «администраторов» возлагаются обязанности управления – изменение статуса, удаление и добавление пользователей, открытие и закрытие «комнаты» и т. д. Основы работы с конференциями рассмотрим на примере модуля MeetMe.

Как было сказано выше, для работы MeetMe необходимы высокоточные таймеры, присутствующие в оборудовании Digium. Если же никаких плат в машине с Asterisk нет, не расстраивайтесь, есть несколько способов программной эмуляции таймеров! Одним из них мы сейчас и воспользуемся. Этот способ применим для ядер 2.4.x (с модулем usb_



Рисунок 1. Окно входа веб-интерфейса системы голосовой почты

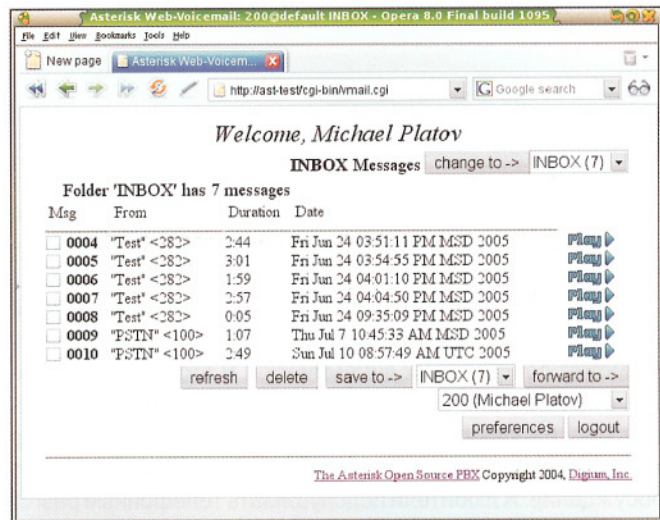


Рисунок 2. «Входящие» сообщения голосовой почты

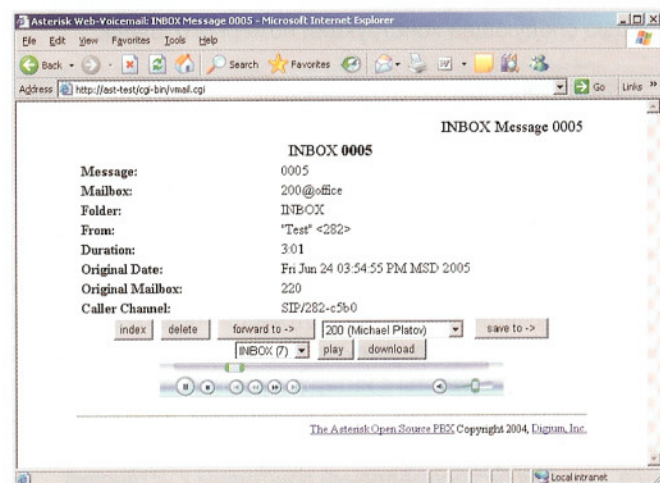


Рисунок 3. Прослушивание сообщения голосовой почты

uhci) и 2.6.x. Откроем Makefile библиотеки zaptel и уберем комментарий (символ «#») перед ztdummy. Для zaptel 1.0.9 это строка 61. После этого пересоберем библиотеку (make & make install) и попробуем загрузить модуль ztdummy:

```
# modprobe ztdummy
```

Убедимся, что у нас это получилось:

```
# lsmod |grep ztdummy
```

```
ztdummy      3620  0
zaptel       225732  1 ztdummy
```

Настроим автоматическую загрузку модуля, добавив соответствующую запись в стартовые скрипты системы

(в случае Gentoo это будет строка в одном из файлов внутри /etc/modules.autoload.d). После этого можно смело приступать к настройке конференций.

Определим «место встречи» (conference room) пользователей. Для этого в файл /etc/asterisk/meetme.conf добавим следующее:

```
[rooms]
conf=> 400
```

Этим мы создали «комнату» с номером 400. Теперь сделаем в ней «дверь». Для этого в файл extensions.conf добавим следующие строки:

```
[confs]
exten => 400 , 1, Meetme, 400
[office]
include=> confs
```

Перезапустите Asterisk и позвоните на номер 400 со всех телефонов. Пользователи попадут в общую «комнату» №400 и смогут одновременно общаться друг с другом.

Кроме статического описания «комнат» конференций (пример, который мы только что рассмотрели), MeetMe также может создавать их динамически. В этом случае человек, открывающий «комнату», определяет для нее соответствующий пароль доступа. После этого он сообщает номер комнаты и пароль остальным участникам, и они начинают обсуждение. А любители подслушивать телефонные разговоры в это время злобно грызут ногти и кусают локти.

Реализовать эту возможность на практике можно с помощью следующей конструкции номерного плана:

```
[confs-dynamic]
exten => 500, 1, MeetMe([MD])
[office]
include => confs-dynamic
```

Теперь сделаем еще один шаг в сторону «идеального сервера» и установим веб-интерфейс для управления конференциями. С его помощью можно легко и просто подключать и отключать участников, давать и отбирать «право» голоса, да и вообще – просто наблюдать за тем, что происходит в наших «комнатах». Для работы с интерфейсом нам так же, как и с голосовой почтой, понадобится веб-сервер, только теперь с интерпретатором PHP. Будем считать, что все это у нас имеется, поэтому без промедлений приступим к установке.

Первым делом скачаем с [12] архив с Web-MeetMe и распакуем его в Apache_DocumentRoot/meetme. Зайдем в каталог phpagi и скопируем файл phpagi.example.conf в /etc/asterisk/phpagi.conf. Внесем в этот файл некоторые изменения:

```
server=127.0.0.1
port=5038
username=webmm
secret=webmmpw
```

Данное приложение для своей работы использует командный интерфейс управления Asterisk (Asterisk Manager Interface – AMI). В целях безопасности по умолчанию он отключен, нам же потребуется его включить. Для этого отредактируем файл /etc/asterisk/manager.conf:

```
[general]
enabled = yes
port = 5038
bindaddr = 127.0.0.1

[webmm]
secret = webmmpw
permit=127.0.0.1
read = system,call,log,verbose,command,agent,user
write = system,call,log,verbose,command,agent,user
```

С конфигурационными файлами теперь все нормально, а вот с самими PHP-скриптами не очень. Дело в том, что web-meetme написан с учетом того, что переменная PHP RegisterGlobals определена как «On». Из соображений безопасности, начиная с PHP-4.2.0, значение по умолчанию для этой переменной – Off. Поэтому, для того чтобы скрипты заработали, необходимо либо поменять значение RegisterGlobals в файле php.ini, либо модифицировать файл conf_control.php, добавив в его начале следующие строки:

```
getpost _ ifset(array('confno'));
getpost _ ifset(array('action'));
getpost _ ifset(array('user _ id'));
```

Кроме того, для нормальной работы с версией Asterisk 1.0.9 мне пришлось закомментировать строку 113 (вызов break) в файле conf_control.php.

Если интерфейс установился нормально, то в браузере мы увидим следующее (см. рис. 4).

Системный телефон

Неотъемлемым атрибутом любой мини-АТС является системный телефон. С его помощью оператор может полностью контролировать состояние мини-АТС (какие линии в данный момент свободны, какие заняты и т. д.). Более функциональным аналогом системного телефона для Asterisk является Flash Operator Panel. С помощью FOP можно переключать и завершать звонки, просматривать запаркованные вызовы, следить и управлять конференц-комнатами, да и просто наблюдать за тем, что происходит с Asterisk.

С точки зрения администратора, FOP является клиент-серверным приложением. В качестве клиента выступает flash-приложение, выполняемое в любом из популярных браузеров (IE, Opera, Mozilla/Firefox). Серверная часть представлена perl-скриптом, в реальном времени получающим от Asterisk информацию о совершаемых в нем действиях. На момент написания этих строк последняя доступ-

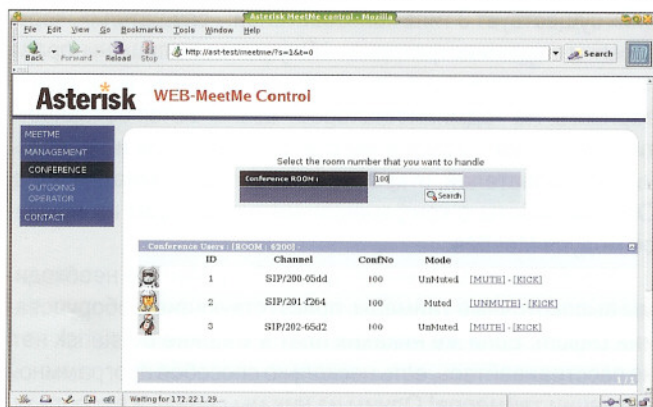


Рисунок 4. Интерфейс управления конференциями

ная версия FOP – .22 (см. рис. 5). Домашняя страница проекта – [4].

При работе с Asterisk FOP также использует Asterisk Manager Interface. Таким образом, первое, что нам нужно будет сделать, – добавить в файл `manager.conf` соответствующую разрешающую запись:

```
[fop_user]
secret = superfopsecret
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.0
read = system,call,log,verbose,command,agent,user
write = system,call,log,verbose,command,agent,user
```

Теперь перейдем к FOP. Настройки серверной части задаются в файле `op_server.cfg`. Наиболее значимыми для нас параметрами являются:

```
manager_host=127.0.0.1
manager_user=fop_user
manager_secret=superfopsecret
```

Этим мы указали perl-серверу, где и как нужно искать Asterisk. К слову, с помощью FOP можно отслеживать состояние несколько серверов Asterisk. В этом случае надо продублировать эти строки необходимое количество раз, задав в них соответствующие параметры других серверов.

```
flash_dir=/var/www/html/fop
web_hostname=ast-test
security_code=fop_pwd
poll_voice_mail=1
```

Первой строкой мы задали местоположение файлов панели. Второй – имя, по которому к серверу будут обращаться клиенты (это необходимо для того чтобы FOP правильно формировал URL в процессе своей работы). Параметр `security_code` назначает пароль, который необходимо будет вводить для выполнения действий (переключение, завершение звонков). Четвертый параметр говорит серверу о том, что ему следует периодически самостоятельно проверять состояние голосовых ящиков (без использования AMI). Данный параметр необходим в случаях, если для доступа к почте используется веб-интерфейс. Следующие настройки касаются работы с конференциями:

```
conference_context=confs
barge_rooms=400
barge_muted=1
```

Первая строка тривиальна – контекст номерного плана, содержащий внутренний номер для доступа к конференции.

А вот на последних двух остановимся более подробно. Дело в том, что с помощью `fop` можно «собирать» людей в конференции. Третий параметр очень полезен в тех ситуациях, когда оператору необходимо прослушать разговор других абонентов. При значении «1» у пользователей, добавляемых FOP к конференции, будет «отключены микрофоны», таким образом можно незаметно подключать участников к конференции или организовывать прослушивание разговоров оператором. При необходимости «микрофон» можно включить, воспользовавшись все тем же веб-интерфейсом. Строкой `barge_rooms` мы определяем те номера



Рисунок 5. Структура Flash Operator Panel

«комнат», которые будут использоваться для реализации этой возможности. Кроме того, на панели FOP можно группировать кнопки по контекстам. Для этого перед ними необходимо написать имя соответствующего контекста.

Параметры визуализации (клиентская часть) задаются в файле `op_buttons`. Синтаксис его достаточно прост. Сначала описываются кнопки, отвечающие за абонентов Asterisk:

```
[SIP/200] ; имя абонента
Position=1 ; номер кнопки на консоли
Label="Vasya" ; надпись на кнопке
Extension=200 ; номер абонента
Context=office ; контекст, в котором
; определен абонент
Mailbox=200@office ; ящик голосовой почты пользователя
Icon=1 ; номер иконки для данного
; пользователя (то 1 до 6)
```

Чтобы не определять настройки для каждого пользователя вручную, можно задать общее описание с использованием регулярного выражения:

```
[_SIP/*]
Position=n ; использовать следующий
; незапятнанный номер
Label="SIP Users"
Extension=-1 ; на эту кнопку перенаправлять
; звонки нельзя
Context=generic_inc
Icon=2
```

Для отображения состояния места парковки воспользуемся следующим:

```
[PARK701]
Position=n
Icon=3
Extension=700
Label="Park 701"
```

А вот так опишем «комнату» конференций:

```
[400]
Position=n
Label="Meetme 400"
Extension=400
Context=confs
Icon=6
```

Следующие строки покажут нам состояние внешней `zaptel`-линии:

```
[Zap/1]
Position=10
Label="External 1"
Extension=-1
Icon=2
```

Для большего удобства на панели можно нарисовать разделяющие прямоугольники и вспомогательные подписи. Сделать это можно, отредактировав файл, отвечающий за визуальное представление – `op_style.cfg`.

В качестве последнего штриха настроим автоматический запуск perl-сервера при старте системы. Для этого мож-

но воспользоваться одним из готовых скриптов, входящих в состав дистрибутива (есть скрипты для Debian и Redhat) или просто прописать вызов `op_server.pl` (или `safe_opserver`) в стартовых скриптах вашего дистрибутива.

Единый центр управления

Как вы могли заметить, настраивая Asterisk, мы редактировали те или иные конфигурационные файлы. Некоторым такой способ администрирования может показаться несколько неудобным. В таком случае рекомендую посмотреть в сторону Asterisk Management Portal. AMP представляет собой веб-приложение, позволяющее управлять настройками Asterisk с помощью популярного веб-браузера. Для лучшего понимания его работы обратимся к следующей схеме (см. рис. 6).

Управление производится следующим образом: посредством веб-интерфейса администратор редактирует базу данных, задавая в ней необходимые параметры (добавление/удаление пользователя, смена пароля, изменение правил маршрутизации и т. д.). Затем, по информации, хранимой в этой базе, веб-сервер обновляет конфигурационные файлы Asterisk и с помощью AMI сообщает серверу о том, что настройки необходимо применить заново. Кроме того, используя дополнительный модуль (веб-интерфейс, для которого также входит в состав AMP), Asterisk помещает информацию о всех совершенных звонках непосредственно в MySQL, что открывает дополнительные возможности для анализа исходящего голосового трафика организации.

После некоторого практического изучения данного продукта сложилось впечатление, что больше всего AMP подходит для 2 групп пользователей:

- Начинающие пользователи, имеющие лишь общее представление об Asterisk.
- Администраторы, уже имеющие опыт работы с Asterisk, без труда разбирающиеся в его конфигурационных файлах и желающие упростить выполнение некоторых наиболее типичных задач.

Для первой группы пользователей AMP позволит построить наиболее популярные вещи, не вдаваясь при этом в дебри конфигурационных файлов с их многочисленными и зачастую непонятными параметрами. Для данной категории самым сложным этапом является правильная установка портала. Впрочем, эта проблема легко решается применением специализированных дистрибутивов, уже имеющих в своем составе все необходимое, в том числе и AMP. Наиболее известный из них — Asterisk@Home [5].

Для второй категории пользователей AMP является мощным средством, позволяющим в короткие сроки получить законченное и достаточно функциональное решение на базе Asterisk. Причем здесь AMP выступает не как полный отказ от ручного редактирования конфигурационных файлов (как это было в первом случае), а скорее как удобное дополнение к ним (например, стандартные вещи делаются в портале, а нестандартные — путем custom-команд AMP или непосредственным редактированием конфигурационных файлов).

На этом позвольте закончить описание теоретическое и перейти к практическому рассмотрению AMP. Учитывая

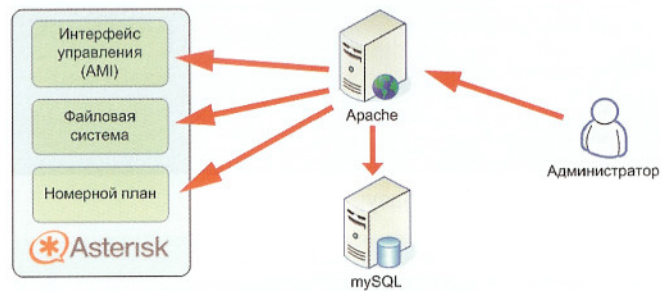


Рисунок 6. Связь AMP и Asterisk

характер работы данного приложения, не имеет смысла придерживаться уже созданной нами конфигурации. Поэтому при настройке Asterisk с использованием AMP будем считать, что перед нами вновь лежит «чистый лист». Итак, возьмем клавиатуру в руки и отправимся навстречу к нашей цели!

Устанавливаем AMP

Для установки портала нам понадобятся все компоненты популярной платформы LAMP — Linux, Apache, MySQL и PHP (последний должен быть скомпилирован с поддержкой nls). Кроме того, нам также потребуются PHP-PEAR-DB, интерпретатор perl, а также perl-модули: Net::Telnet, IPC::Signal и Proc::WaitStat.

Первые установим с помощью средств, имеющихся в распоряжении вашего дистрибутива, а для вторых воспользуемся репозитарием CPAN:

```
# perl -MCPAN -e "install Net::Telnet"
# perl -MCPAN -e "install IPC::Signal"
# perl -MCPAN -e "install Proc::WaitStat"
```

Помимо этого нам потребуется добавить поддержку perl к самому Asterisk. Для этого установим модуль asterisk-perl, предварительно загрузив его с [6].

На этом подготовительные действия завершены, и мы можем смело приступить к установке самого AMP. По уже сложившейся традиции, воспользуемся последней стабильной версией, доступной на момент написания этих строк — 1.10.008. Загрузим и распакуем tarball с файлами проекта в удобное для нас место. Для определенности пусть это будет /usr/src:

```
# cd /usr/src
# wget http://citkit.dl.sourceforge.net/sourceforge/ampportal/AMP-1.10.008.tar.gz
# tar xzf ./AMP-1.10.008.tar
```

Создадим две базы данных в MySQL. В первой будет храниться служебная информация AMP, во второй — информация о совершенных пользователями звонках.

```
# mysqladmin create asterisk -p
Enter password: <пароль root-пользователя MySQL>
# mysql -u root asterisk -p < usr/src/AMP/SQL/newinstall.sql
```

Первая база создана. Теперь дадим пользователю Asterisk необходимые права доступа:

```
# mysql -u root -p
```

```
Enter password: <пароль root-пользователя MySQL>
mysql> GRANT ALL PRIVILEGES
-> ON asterisk.*
-> TO auser@localhost
-> IDENTIFIED BY 'p0rtalpwd';
Query OK, 0 rows affected (0.00 sec)
mysql> \q
```

Создадим вторую базу:

```
# mysqladmin create asteriskcdrdb -p
Enter password: <пароль root-пользователя MySQL>

# mysql -u root asterisk -p < \
usr/src/AMP/SQL/cdr_mysql_table.sql

Enter password: <пароль root-пользователя MySQL>
mysql> GRANT ALL PRIVILEGES
-> ON asteriskcdrdb.*
-> TO auser@localhost
-> IDENTIFIED BY 'p0rtalpwd';
Query OK, 0 rows affected (0.00 sec)
mysql> \q
```

Теперь установим модуль регистрации звонков cdr_mysql, входящий в состав пакета asterisk-addons.:

```
# wget http://www.asterisk.org/html/downloads/ \
asterisk-addons-1.0.9.tar.gz
# tar xzf ./asterisk-addons-1.0.9.tar.gz
# cd ./asterisk-addons-1.0.9
```

В Makefile добавим следующую запись (строка 21):

```
CFLAGS+=DMYSQL_LOGUNIQUEID
```

И завершим инсталляцию модулей, выполнив команды:

```
# make && make install
```

Стандартная схема установки AMP предполагает, что Asterisk и веб-сервер должны выполняться одним пользователем:

```
# groupadd asterisk
# useradd -c "asterisk PBX" -d /var/lib/asterisk -u \
5060 -g asterisk asterisk
```

Настроим запуск веб-сервера от пользователя Asterisk. Для этого в конфигурационном файле apache напомним:

```
Userasterisk
Groupasterisk
```

Теперь установим файлы самого AMP. Для этого запустим соответствующий установочный скрипт (учтите, что после его выполнения существующие файлы конфигурации Asterisk будут замещены, поэтому при необходимости не забудьте сделать резервные копии):

```
# /usr/src/AMP/install _amp
```

В интерактивном режиме скрипт выводит у нас все необходимые имена, явки и пароли (попутно сохранив их для потомков в /etc/amportal.conf), копирует все необходимые файлы и проинициализирует базу данных MySQL. Настроим работу asterisk с использованием учетной записи обычного пользователя. Для этого в стартовом скрипте Asterisk используем конструкцию следующего вида:

```
/usr/sbin/asterisk -U asterisk
```

Убедимся, что автозапуск op_server.pl настроен и сам сервер запущен. Если все было сделано правильно, то, открыв http://имя_машины, мы увидим следующее (см. рис. 7).

Настройка AMP

С интерфейсом голосовой почты мы уже знакомы, поэтому без лишних проволочек перейдем в основной раздел управления, щелкнув по ссылке «Asterisk Management Portal».

Перед нами предстанут три раздела меню: настройка (Setup), отчеты (Reports) и панель FOP. Как следует из названия, для конфигурации системы используется Setup. Возможные действия в этом разделе перечислены в меню слева. Первым делом перейдем в подраздел Extensions и создадим несколько учетных записей пользователей (см. рис. 8).

При настройках по умолчанию (и с установленным zapтел-оборудованием) сразу же после добавления пользователей и применения конфигурации уже можно совершать звонки, в том числе и с использованием ZAP-каналов через префикс 7. Правда, скорее всего нам предварительно потребуются немного подправить настройки zaptel для учета специфики отечественных АТС, но, учитывая наш опыт из [1], это вряд ли вызовет у нас какие-либо серьезные трудности.

Для настройки междугородней IP-телефонии необходимо создать транки (способ описания точек терминирования внешнего по отношению к Asterisk трафика) и добавить соответствующие правила маршрутизации (раздел Outbound Routing). Формат этих правил во многом совпадает с тем, что мы определяли в текстовых файлах, поэтому практическое рассмотрение данной возможности мы оставим для самостоятельного изучения. Благо с использованием контекстных подсказок веб-интерфейса этот процесс не является очень сложным.

В разделе Ring Groups мы несколькими движениями мышки можем объединить абонентов в единую группу вызова. Кроме того, также имеется возможность задать альтернативное поведение системы для случаев, когда ни один из абонентов группы не доступен (см. рис. 9).

В Incoming Calls определяются правила обработки звонков, приходящих с ZAP-каналов. Стандартной функциональностью коробочного номерного плана AMP является возможность определения различных правил маршрутизации для рабочего и нерабочего времени. При этом в качестве адресатов могут выступать группы вызова, системы голосовых меню, очереди или обычные абоненты.

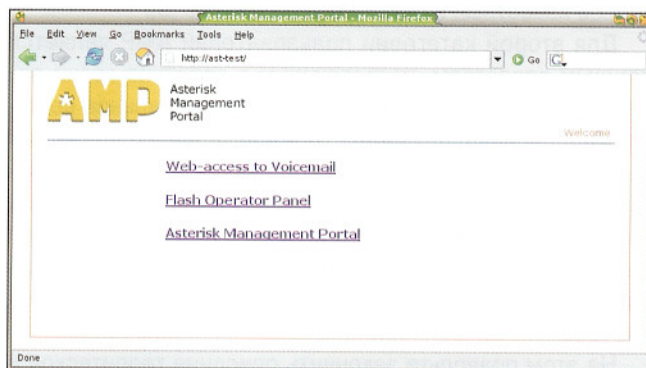


Рисунок 7. Стартовая страница AMP

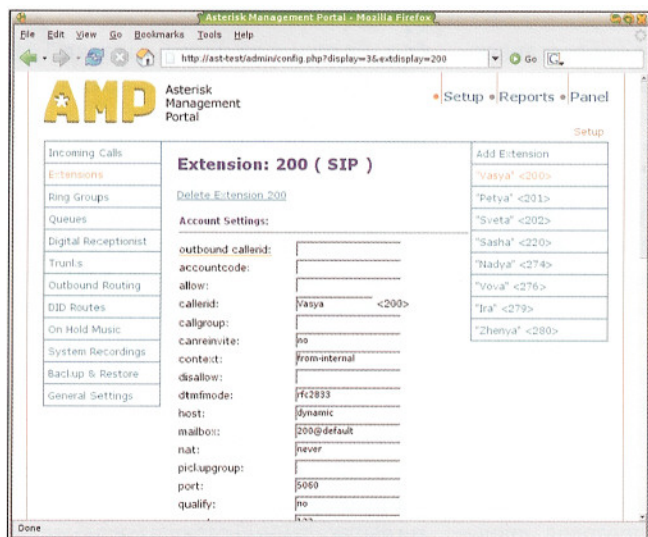


Рисунок 8. Создание абонентов в AMP

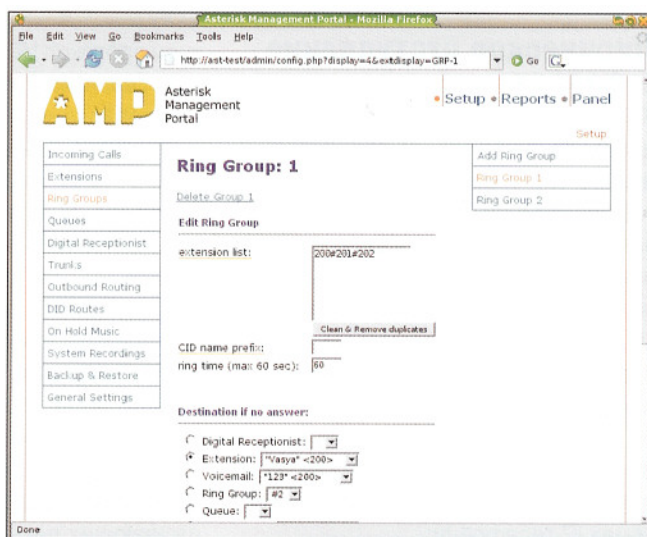


Рисунок 9. Группы вызовов абонентов

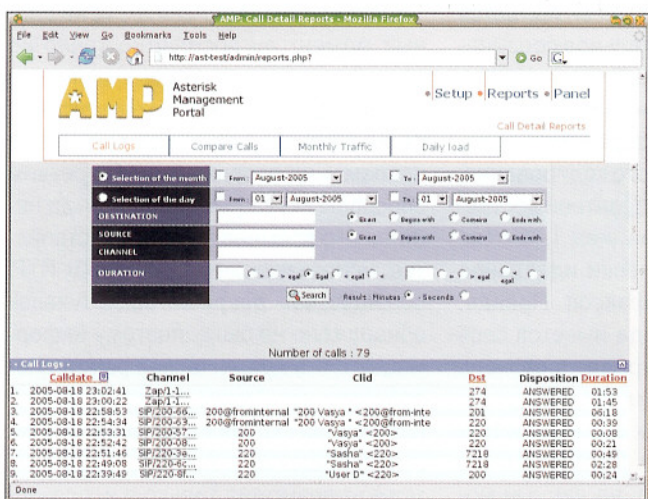


Рисунок 10. Модуль отчетов AMP

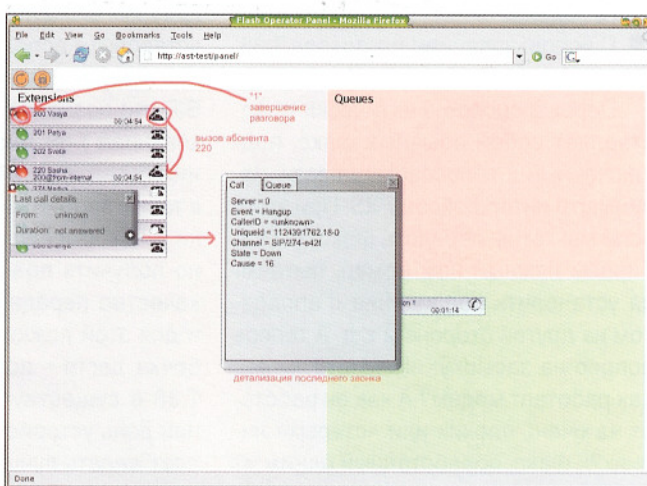


Рисунок 11. Интерфейс Flash Operator Panel

Для остальных разделов конфигурации ограничимся кратким описанием их назначения.

- **Queues** – здесь определяются параметры очередей и агентов. Более подробно эти параметры будут рассмотрены в одном из следующих номеров журнала.
- **Digital Receptionist** – автосекретарь. С помощью этого пункта можно быстро создать голосовое меню. Например, запишем такой текст: «Здравствуйте! Вы позвонили в СамуюЛучшуюКомпанию. Для соединения с коммерческим отделом нажмите 1. Технический отдел – 2. Бухгалтерия – 3 и т. д.» Затем с помощью того же мастера создадим правила маршрутизации для этого меню, перенаправляющие пользователей в соответствующие отделы (в общем случае правила могут не соответствовать текстовому сообщению). И, наконец, назначим созданное меню для входящих звонков в подраздел Incoming Calls. Этим самым мы значительно снизим нагрузку на нашего секретаря, освободив его время для более интересной работы.
- **DID Routes** – маршруты прямого вызова. С помощью данной функции можно напрямую позвонить внутреннему абоненту/агенту/группе вызова Asterisk с использованием механизма DID (при этом DID также должен поддерживаться городской телефонной станцией).

- **On Hold Music** – в этом подразделе мы можем загрузить на сервер дополнительные мелодии для музыки при ожидании.
- **System Recording** – с помощью этого пункта меню можно добавлять (или записывать) в Asterisk дополнительные звуковые сообщения, которые впоследствии могут быть использованы в голосовых меню.
- **Backup & Restore** – как следует из названия, с помощью данного раздела можно резервировать и восстанавливать конфигурацию Asterisk-сервера. Для работы этой функции необходимо добавить в crontab выполнение содержимого /etc/asterisk/backup.conf.
- **General Settings** – в этом пункте определяются некоторые системные параметры AMP – время дозвона до абонента перед перенаправлением к голосовой почте, параметры работы с каталогом пользователей и настройки обработки входящих факсов.

Более менее разобравшись с имеющимися возможностями настройки, остановимся на модуле отчетов. С его помощью администратор может не только получить детальную информацию за любой интересующий его период времени, но и просмотреть статистическую информацию о загрузке сервера в течение дня или на протяжении месяца.

Лирическое отступление о факсах

Как вы могли заметить, один из разделов меню AMP посвящен настройке приема факсов в Asterisk, однако в статье об этой возможности не было сказано ни единого слова. Причиной тому является текущий уровень реализации поддержки работы с факсами в системах IP-телефонии в целом и в Asterisk в частности. Давайте рассмотрим эту проблему более внимательно.

На самом верхнем уровне можно выделить два основных способа передачи факсимильных сообщений в системах IP-телефонии:

- Поверх существующего голосового кодека.
- С использованием протокола T.38.

Первый способ фактически представляет собой обычный факс, подключенный к голосовому шлюзу, например с интерфейсом FXS. При этом, если мы хотим передать факс, мы набираем нужный нам номер, пытаемся установить соединение с аппаратом на другой стороне и т. д. А теперь вопрос на засыпку: «Помните ли вы, как работает модем? А как он работает на очень плохих или «старых» линиях?» Факс, обработанный одним из голосовых кодеков с сильным сжатием (GSM, G723, G729), будет работать еще хуже, вернее, скорее всего он не будет работать вообще. С кодеками, не использующими сильное сжатие с потерями (в Asterisk это G711 в вариантах a-law и u-law), ситуация будет лучше. При этом качество передачи факсов во многом будет зависеть от реального состояния IP-канала (потери, джиттер, задержка) между абонентами (ведь на самом деле мы передаем «голос», следовательно, исполь-

зуем протокол UDP, не предоставляющий гарантий доставки пакетов). Если вдобавок еще и взглянуть на первую строчку таблицы в [8], то мы увидим, что при передаче факса через Интернет нам потребуется канал с реальной пропускной способностью около 80 кБит, что, согласитесь, уже немало. По этим и другим причинам для передачи факсов через голосовые сети все чаще стараются использовать T.38. Данный протокол был разработан организацией ITU-T специально для передачи факсов в VoIP-сетях. Механизм работы устройств, поддерживающих данный протокол, следующий – при обнаружении голосовым шлюзом сигналов факса он пытается установить дополнительное соединение с другим шлюзом (благо и H323 и SIP это позволяют), но уже с использованием T.38. Благодаря использованию помехоустойчивого кодирования, а также протокола с гарантией доставки – TCP с использованием T.38 можно получить практически идеальное качество передачи факсов. Правда, и для этой ложки меда имеется своя бочка дегтя – поддержка протокола T.38 в существующих на сегодняшний день устройствах все еще оставляет желать лучшего. Дело в том, что согласно стандартам, на данный момент существует 3 (!) различных способа передачи факсов с использованием T.38:

- Поверх протокола UDP.
- Поверх протокола TCP.
- Передача в RTP-пакетах.

Первые два способа фактически представляют собой специализированные протоколы для передачи факсов (протоколы IFP/udptl и IFP/TCP соответственно), в то время как третий

больше похож на использование T.38 вместо обычного кодека (факсовые данные упаковываются в обычные RTP-пакеты). Проще говоря, поддержка в прокси-серверах первых двух протоколов требует дополнительных усилий от разработчиков, в то время как для третьего варианта достаточно поддержки в конечных устройствах. К тому же ситуация усугубляется тем, что зачастую производители голосового оборудования если и реализуют поддержку T.38 в своих устройствах, то ограничиваются лишь одним из вариантов реализации (чаще первым), со всеми вытекающими отсюда несовместимостями и проблемами.

Что же касается Asterisk, то на данный момент он абсолютно точно не поддерживает два первых варианта использования T.38 (заметьте, это вовсе не означает, что такая поддержка невозможна в принципе). С третьим способом ситуация комичнее – до настоящего времени ни одного устройства, использующего T.38 поверх RTP, сообщество разработчиков Asterisk обнаружено не было, поэтому информации о его практической работоспособности нет. Таким образом, реальная поддержка передачи факсимильных сообщений в Asterisk на данный момент возможна только с использованием кодека G711, причем наиболее предпочтительна следующая схема: входящий факс-сервер, работающий поверх G711 с использованием zaptel-устройств Asterisk.

В завершение упомянем и другой взгляд на эту проблему. А нужна ли вообще передача факсов в VoIP-сетях? Если IP-инфраструктура уже существует, то не проще ли в данном случае использовать что-то другое, например e-mail?

Поддерживается выгрузка отчетных данных в формате pdf и csv (см. **рис. 10**).

Интерфейс панели управления

Третьим разделом интерфейса управления является уже знакомая нам Flash Operator Panel. В состав последней версии AMP входит несколько устаревшая версия FOP (20 против 22 на сайте разработчика). Правда, некоторой компенсации за это является интеграция AMP и FOP при установке «из коробки».

Так, добавляя нового пользователя в AMP, запись о нем автоматически появляется и в FOP, что избавляет нас

от необходимости лишний раз редактировать файлы конфигурации.

Давайте более подробно остановимся на графическом интерфейсе. Как уже говорилось выше, клиентская часть FOP разработана с использованием Macromedia Flash. Это не могло не оставить отпечатка на характере работы с визуальным интерфейсом. Все действия с абонентами производятся либо с помощью двойного щелчка мыши, либо методом «drag and drop». Так, для того чтобы принудительно завершить сеанс пользователя, необходимо нажать на «1» (см. **рис. 11**), а для того чтобы соединить двух абонентов, – перетащить значок телефона на кнопку вызываемого абонента.

AMP мини-FAQ

Страница FOP «зависает» при открытии. В чем может быть причина?

Убедитесь, что серверный perl-скрипт запущен.

При работе с веб-интерфейсом выдаются сообщения об ошибках записи.

Возможно, установлены неверные права доступа к файлам конфигурации Asterisk. Для автоматического исправления запустите скрипт `apply_conf.sh` из дистрибутива AMP.

При работе с интерфейсом выдается сообщение об ошибке соединения с manager.

Неверно сконфигурирован (или отключен) интерфейс управления Asterisk. Проверьте содержимое файлов `/etc/asterisk.manager.conf` и `/etc/ampportal.conf`.

К сожалению, текущая реализация совместной работы FOP и Asterisk не лишена недостатков. Так, в варианте «из коробки» FOP не видит стандартных конференций AMP. Для решения этой проблемы воспользуемся нашими знаниями о структуре конфигурационных файлов. Первым делом в файле `op_server.cfg` в качестве контекста для конференций укажем `ext-meetme`, для параметра `barge_rooms` выберем одно из стандартных для AMP значений, например 8200, и опишем соответствующие кнопки в файле `op_buttons_custom.cfg` согласно инструкции, приведенной выше для FOP. После этого перезапустим perl-сервер, обновим страницу интерфейса клиента и порадуемся, увидев в нем кнопки, соответствующие MeetMe. Теперь с помощью веб-интерфейса мы можем «собирать» людей в виртуальных «комнатах» для проведения совместных обсуждений (drag-and-drop на картинку конференции), а также подключаться к уже идущим разговорам. (правда, для незаметной работы последнего, возможно, потребуются более тонко настроить работу модуля MeetMe). С отображением состояния парковочных мест можно поступить аналогично.

Заключение

В заключение давайте еще раз кратко рассмотрим сильные и слабые стороны AMP. Итак, преимущества:

- Возможность редактирования параметров Asterisk через веб-интерфейс.
- Возможность получения информации о совершенных звонках.
- Быстрое создание голосовых меню.
- Возможности автосекретаря.
- Использование «транков».
- Определение правил маршрутизации.
- Загрузка мелодий для «музыки при ожидании».
- Определение групп вызова.

Но, как известно, у каждой медали есть и вторая сторона. У AMP она выглядит так:

- Необходимость перезаписи конфигурационных файлов.
- Необходимость функционирования всех ключевых компонент на одном компьютере (как следствие из предыдущего).

Asterisk начинает запускаться, но, дойдя до загрузки zap-устройств, процесс внезапно завершается.

Проверьте, что модули ядра `zaptel` и `wcfxo` загружены и устройства проинициализировались без ошибок (`lsmod`, `dmesg`).

Или проверьте, что файл `/etc/asterisk/zaptel.conf` существует и в нем правильно заданы параметры устройств.

Не показываются (или показываются неверно) сообщения голосовой почты через веб-интерфейс.

Убедитесь, что пользователь веб-сервера имеет доступ к файлам голосовой почты (`/var/spool/asterisk/voicemail`).

Добавил пользователей, но в FOP они не появились. В чем может быть дело?

Необходимо перезапустить `op_server.pl`

- Возможные проблемы с масштабируемостью (как следствие из предыдущего).
- Отсутствие управления параметрами MeetMe через веб-интерфейс.
- Отсутствие встроенной системы безопасности.
- Достаточно сложная структура номерного плана (особенно для начинающих пользователей).
- Шероховатости в интеграции с FOP.
- Отсутствие русского интерфейса меню.
- Проблемы совместимости между номерным планом и аппаратным голосовым оборудованием (некоторые шлюзы не могут корректно набирать символы «#,*» которые очень активно используются в номерном плане).

При всем этом, несмотря на указанные недостатки, Asterisk Management Portal, безусловно, можно считать одним из лучших веб-интерфейсов для Asterisk. Проект очень активно развивается, и вполне возможно, что многие из перечисленных недочетов будут исправлены в следующей версии. При этом не стоит забывать, что AMP полностью открыт и вы сами можете принять непосредственное участие в его дальнейшем развитии. Ну а если разработка веб-приложений вам не близка, а AMP вам просто понравился как пользователю, не забудьте сделать `donation` на официальном сайте проекта [7].

Литература, ссылки:

1. Платов М. Asterisk и Linux – миссия IP-телефония. Действие 2. – Журнал «Системный администратор», №7, 2005 г. – 32-38 с.
2. Платов М. Asterisk и Linux – миссия IP-телефония. – Журнал «Системный администратор», №6, 2005 г. – 12-19 с.
3. <http://www.voip-info.org/tiki-index.php?page=Asterisk+gui+vmail.cgi>.
4. <http://www.asternic.org>.
5. <http://asteriskathome.sourceforge.net>.
6. <http://asterisk.gnuinter.net>.
7. <http://sourceforge.net/projects/ampportal>.
8. Платов М. Что важно знать об IP-телефонии. – Журнал «Системный администратор», №5, 2005 г. – 20-25 с.