

АВТОМАТИЗАЦИЯ НАСТРОЙКИ BIOS СЕРВЕРОВ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

А. С. Гусихин, С. О. Черных

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Введение

Сервера различных производителей предоставляют различный функционал для конфигурирования BIOS. Так, например, сервера Dell позволяют конфигурировать BIOS с помощью сценариев применяемых после перезагрузки сервера. Однако, такой функционал встречается редко у других производителей, либо приобретается, как проприетарный продукт, за дополнительную плату.

К сожалению, сервера Supermicro и Intel не поддерживают настройки BIOS с помощью сценариев исполняемых в ОС, что вынуждает выполнять конфигурирование каждого сервера в ручном режиме.

Таким образом, настройка BIOS на серверах вычислительного поля мультипроцессорной вычислительной системы является одной из самых долгих процедур предварительной подготовки вычислительной системы.

В стандартном виде, настройка BIOS всего вычислительного поля вычислительной системы выглядит так:

- подключится к серверу посредством монитора и клавиатуры;
- перезагрузить сервер. В зависимости от типа сервера, перезагрузка может длиться от минуты до нескольких десятков минут;
- дождаться и не пропустить предложения войти в меню конфигурации BIOS. Некоторые сервера чувствительны к количеству нажатий кнопки Del указывающей серверу на необходимость входа в режим конфигурирования BIOS;
- настроить несколько пунктов BIOS;
- сохранить выбранные настройки и выйти.

Такую операцию необходимо выполнить для каждого сервера. С увеличением количества серверов, длительность операции линейно увеличивается.

Чуть позже, настройка BIOS осуществлялась по описанной выше схеме с некоторыми элементами автоматизации:

- *подключение клавиатуры и монитора к серверу.* Доступ к клавиатуре и видео удаленного сервера выполнялся посредством поддерживаемых BMC функций по ретрансляции видео сигнала, клавиатуры и мыши.
- *ввод сервера в режим конфигурирования BIOS.* Автоматизирован за счет поддержки сервера-

ми Supermicro и Intel стандарта IPMI на интегрированные средства мониторинга оборудования, а именно, за счет указания перехода при последующей перезагрузке в меню конфигурирования BIOS;

- *настройка BIOS.* Продолжала осуществляться вручную;
- *проверка правильности установленных настроек BIOS.* Продолжала осуществляться вручную;
- *перевод сервера в режим загрузки ОС.*

При описанном выше уровне автоматизации настройки BIOS, скорость конфигурирования, в идеальном случае, могла доходить до одного сервера в минуту. Данная рутинная процедура требует концентрации внимания операторов и быстро утомляет.

Последующее изменение даже одного из параметров BIOS на всем вычислительном поле становится длительной операцией, требующей группы операторов и длительной профилактики. Проверить единообразность конфигурирования всех серверов можно только выборочно или полностью довериться добросовестности оператора.

Так, например, настройка BIOS на гипотетической вычислительной системе, с количеством узлов равным тысяче, потребовала бы пяти полных рабочих дней и усилий коллектива из шести сотрудников.

Таким образом, при большом количестве узлов актуальность автоматизации конфигурирования возрастает.

Данная работа предлагает автоматизацию этапа ручной настройки BIOS на серверах.

Предлагаемый подход должен значительно сократить длительность данного этапа.

В своей основе работа опирается на возможность удаленного доступа к последовательному порту сервера через сеть (serial-over-lan, sol).

В процессе работы были рассмотрены существующие методы решения данной проблемы, графический и текстовый режим работы с BMC.

В обоих режимах рассматривалась возможность передачи выбранному процессу последовательности нажатий клавиш.

1. Обзор возможных подходов к конфигурированию

Длительность и рутинность операции конфигурирования BIOS вынуждала искать пути автоматиза-

ции для снижения длительности и трудоемкости данной процедуры.

Предприималось множество попыток автоматизации за счет различных подходов.

1.1. Конфигурирование на основе копирования и прошивки конфигурации

Одним из наиболее удобных средств настройки BIOS, до некоторого времени, была утилита CMOS.

Настройка BIOS узлов поля выполнялась за счет конфигурирования одного из серверов, с дальнейшим копированием из CMOS материнской платы настроек BIOS и записью в CMOS всех серверов.

Считывание и запись осуществлялись свободно распространяемой утилитой. Однако, с выходом архитектуры SandyBridge, данная утилита перестала функционировать. Она считывает и записывает некую область памяти, однако это никак не влияет на настройку BIOS.

Существуют проприетарные утилиты считывания и записи информации BIOS, однако найти их в свободном доступе не удалось.

1.2. Конфигураторы на основе возможностей BMC и стандарта IPMI

Следующей ветвью развития конфигураторов стало выполнение настройки в посредством возможностей контроллера BMC и стандарта IPMI.

Как известно, контроллер BMC имеет возможность предоставлять доступ к терминалу (клавиатура, мышь, экран) различными способами

- графический режим, через web браузер, посредством java апплета. Преимущество данного подхода в том, что возможна ретрансляция любой растровой графики с контролируемого по BMC сервера;

- в текстовом режиме, в терминале – Serial-Over-Lan или SOL. Выполняется за счет ретрансляции доступа к последовательному порту контролируемого сервера. BMC, в данном случае, выполняет роль транспорта. Поддержка графики только на уровне псевдографики. Поддерживается только при соединении по интерфейсу IPMI lanplus.

Доступ к консоли может быть осуществлен любым из перечисленных способов.

1.2.1. Графические конфигураторы

Xdotool позволяет программно симулировать ввод клавиш и активность мыши, передвигать и масштабировать окна. Эта программа использует возможности графической оконной системы X Window System (X11) и функции xlib для выполнения своих действий. Xdotool имеет свои преимущества и недостатки. Преимуществом является то, что пользователь имеет возможность посылать события нажатия клавиш в любое окно.

Главный недостаток – окно, в которое необходимо передать событие, должно быть в фокусе (т. е. выделено). Это значит, что мы не имеем возможности параллельно работать с несколькими окнами.

В случае с настройкой BIOS, пришлось бы последовательно подключаться к компьютеру и настраивать нужные параметры.

Второй недостаток – контроль действий. Программа лишь посылает последовательность нажатий клавиш и не может контролировать текущее положение курсора.

Т. е. если в меню BIOS курсор будет стоять не на предполагаемом месте, то скрипт не будет работать.

Для корректной работы скрипта необходимо:

- сфокусированное окно;
- передача уникального идентификатора сфокусированного окна скрипту;
- в меню BIOS курсор должен стоять в определенном положении.

Для Xdotool были написаны и протестированы сценарии настройки BIOS. Выводом из проведенного тестирования явилось, что утилиту Xdotool трудно адаптировать для решения нашей задачи, так как конфигурирование на основе xdotool имеет ряд ограничений:

- нельзя проверить загружен ли узел в режиме конфигурирования BIOS;
- нельзя распараллелить задачу;
- нельзя осуществить контроль ввода;
- курсор в меню BIOS должен находиться в определенном положении;
- нельзя посмотреть результат работы;
- сложно организовать работу сценария со множеством узлов;
- работа скрипта ограничивает работу пользователя (т.к. окно должно быть постоянно в фокусе, пользователь не может использовать текущую графическую сессию во время исполнения сценария).

1.2.2. Консольный конфигурирование основе expect + IPMI SOL

Из двух возможных вариантов доступа к консоли (посредством web и sol), доступ через SoL выглядит наиболее удобным. SoL работает в текстовом режиме, что позволяет распознавать текст на консоли и обрабатывать его регулярными выражениями. Тогда как Web консоль представляет из себя снимок видео-буфера, процесс обработки которого, является сам по себе проблемой.

Expect – инструмент для автоматизации и тестирования в ОС Unix, написанный Don Libes как расширение к скрипт-языку Tcl, для интерактивных приложений таких, как telnet, ftp, passwd, fsck, rlogin, tip, ssh, и других.

С помощью Tk интерактивные приложения могут быть использованы с графическим интерфейсом (GUI) X11.

Expect выступает в качестве клея для склеивания существующих утилит. Главная идея состоит в том, чтобы использовать уже имеющиеся инструменты, а не решать проблему внутри Expect.

Одним из главных применений Expreст являются коммерческие приложения. Многие из этих приложений предоставляют интерфейс командной строки, но, как правило, его недостаточно для написания скриптов. Они созданы для обеспечения пользователя средствами администрирования, но производители зачастую не тратят денег на создание полноценного скриптового языка.

Позволяет по заданному пользователем сценарию двигаться по терминалу, генерируя нажатия клавиш.

Сетевое взаимодействие BMC с пользователем строится на базе протокола UDP, который допускает потерю пакетов. Одним из самых существенных минусов SoL являются изредка возникающие ‘притормаживания’ и потеря генерируемых нажатий.

Это выражается в том, что сценарий может сгенерировать нажатие ‘вниз’, а затем нажатие Enter для выбора необходимой опции. На практике нажатие ‘вниз’ может быть утеряно, а нажатие Enter исполнено.

Таким образом, сценарий будет конфигурировать все что угодно, кроме того что надо и проконтролировать его действия представляется сложным.

Данный минус интерфейса SoL перечеркивает все возможности утилит Xdotool и языка написания сценариев Expreст.

1.2.3. Конфигурирование на основе Tmux + IPMI SOL

Tmux – свободно распространяемая консольная утилита-мультиплексор с открытым исходным кодом, предоставляющая пользователю доступ к нескольким терминалам в рамках одного экрана. Tmux является штатным мультиплексором терминалов операционной системы OpenBSD. Аналогом Tmux является мультиплексор терминалов Screen. Однако, на взгляд авторов, Tmux обладает большей гибкостью и конфигурируемостью. Tmux может отправлять события нажатия клавиш в связанные с ним окна терминалов- клиентов, а также захватывать экран и выводить его в ASCII формате.

Tmux имеет ряд больших преимуществ перед xdotool:

- возможность работать в фоне без активного окна;
- визуальный контроль
- возможность генерации управляющих воздействий на терминал в сессии;
- возможность параллельной работы скриптов с несколькими сессиями;
- возможность сохранить текущее состояние терминала в текстовом ASCII формате с Esc последовательностями. Это позволяет применять для обработки полученного текстового снимка регулярные выражения и, по Esc последовательностям, отслеживать текущую позицию курсора.

Одна из основных особенностей Tmux – захват экрана. Эта функция в Tmux появилась с версии 1.9. Захват экрана позволяет преодолеть ограничения интерфейса SoL, где мы не могли определить текущее положение курсора и не могли осуществить проверку своих действий. С помощью захвата мы не просто посылаем события нажатия клавиш, а контролируем перемещение курсора, ищем интересующий нас элемент меню BIOS и выбираем этот элемент.

На первый взгляд, синхронизация ввода между панелями позволяет сделать конфигуратор простым и интуитивно понятным, однако в следствии пропадания команд в интерфейсе SoL нет гарантии, что все сессии будут выполнять одни и те же действия. Рассинхронизация между управляющей сессией и повторяющимися сессиями может привести к изменению абсолютно любых настроек без возможности это отследить.

Таким образом, Tmux со снимками консоли обладает всеми возможностями для создания на его базе конфигуратора.

2. Конфигуратор

Конфигуратор выполняет настройку BIOS по заданному конфигурационному файлу на узле посредством IPMI SoL. Для своей работы конфигуратор требует наличия ПО на сервере:

- libevent-2.0.20 и выше;
- tmux 1.9 и выше;
- bash, hostlist, sed, awk.

Конфигуратор состоит из:

- конфиг файла;
- сценария setBIOS.sh состоящего из:
 - блока разбора конфигурационного файла;
 - блока реакции на подвисание/пропадание команд в интерфейсе SoL;
 - блока поиска основного меню BIOS (на случай, если узел находится не в меню конфигурации BIOS);
 - блока поиска в основном меню (движение влево, вправо);
 - блока поиска в подпунктах меню (движение вниз, вверх);
 - блока выбора из перечисляемых параметров (выбор кнопкой +);
 - блока снимка экрана и его сохранения в указанный файл.

Одним из настраиваемых параметров генерации нажатий является интервал между командами.

Как показала практика, он не может быть равным нулю. Тогда генерируемые в терминал команды пропадают.

Внутри сценария таймаут выполняет с помощью команды sleep с указанием длительности простоя в сек. Допускаются дробные значения.

Наименьший таймаут между командами должен быть не меньше «Character Accumulate Level» (ms) или 0.1 сек.

Для запуска конфигуратора необходимо:

- перевести все узлы в режим конфигурирования BIOS;
- настроить конфиг-файл. Войти в меню BIOS и указать названия меню, его пунктов и принимаемые значения. Для своей работы конфигуратор требует точного соответствия синтаксиса. От версии BIOS синтаксис пунктов меню может меняться;
- выполнить сценарий конфигуратора в последовательном режиме или в параллельном режиме;
- отфильтровать файлы-снимки всех устанавливаемых опций и удостовериться в правильности установки.

Одной из самых больших проблем SoL является пропадание команд. Оно происходит спонтанно и не предсказуемо.

Для нивелирования данного эффекта, генерация команд передвижения курсора выполняется до тех пор, пока курсор не дойдет до искомой строки либо пока не остановится.

Кроме этого, не редки случаи подвисания IPMI SoL сессии (особенно часто это проявлялось на старых материнских платах для процессоров Westmere).

Индикатором данной проблемы был идентичный снимок экрана даже после генерации сдвига курсора.

Решалась такая проблема многочисленной генерацией клавиши Escape, что в большом количестве случаев, выводило сессию из подвешенного состояния.

3. Опыт применения

Для уверенности в работоспособности сценария конфигуратор BIOS был проверен на различных су-

ществующих и эмулируемых вычислительных системах.

Длительность успешного конфигурирования одного 7 узлов в последовательном режиме колеблется от 36 до 67 сек. Длительность колеблется из-за разного количества 'подвисших' операций, длительности распознавания 'подвисания' и их восстановления сценарием. Длительность не зависит от количества параллельно конфигурируемых узлов.

В процессе тестирования на разном оборудовании было выявлено, что в зависимости от версии BIOS Escape последовательности 'обрамляющие' позицию курсора могут незначительно меняться.

Исходя из этого, в сценарий внесены доработки, позволяющие учитывать разные Escape последовательности разных BIOS.

На более новых материнских платах таймаут между командами может быть уменьшен до 0.1, что так же уменьшает длительно конфигурирования.

Реализован и проверен метод автоматизации конфигурирования BIOS с возможностью последующей автоматизированной проверки конфигурации.

Метод основывается на возможностях интерфейса IPMI SoL и утилиты терминального мультиплексора Tmux.

Предлагаемый метод и сценарий конфигурирования позволяет снизить длительность настройки новых серверов вычислительного поля с нескольких дней и коллектива из 5–6 человек до нескольких минут и одного специалиста.

При поступлении нового оборудования конфиг-файл необходимо адаптировать к текущему меню конфигурации BIOS, однако, потери времени на адаптацию окупаются при автоматической настройке большого количества узлов.