vmstat – выявление узких мест

Утилита **vmstat** служит для определения производительности системы. Наиболее эффективным средством для оценки необходимого объема ресурсов является команда vmstat, которая предоставляет информацию о загруженности процессора, интенсивности операций дискового ввода-вывода и использовании оперативной памяти. Если вы укажете в команде vmstat довольно большой период сбора информации, например, 10 секунд, то выполнение этой команды не потребует много процессорного времени.

Ключи запуска vmstat

- **-S** unit size k:1000 K:1024 m:1000000 M:1048576 (default is K)

Расшифровка вывода vmstat

Выдавать информацию о состоянии системы каждые 10 секунд, количество запросов 3 (если не указано – выводить бесконечно).

```
# vmstat 10 3
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in cs us sy id wa st
 1  0  18616 8160120 329144 5773240    0    0     8    58    0 2  5 10 81  4  0
 1  0  18616 8159608 329152 5773372    0    0     0   312 6485 2252  3  8 87  3  0
 1  0  18616 8159932 329164 5773464    0    0     0   323 6703 1760  3  8 87  2  0
```

Первый отчет команды vmstat содержит информацию, накопленную с момента загрузки системы до вызова команды vmstat. В каждом следующем наборе выдается информация, собранная за предшествующий интервал времени (в данном случае – за 10 секунд).

- Если сумма значений в полях us и sy (соответственно пользователь и система) в разделе ресурсов cpu в каком-либо интервале превышает 90%, то нагрузка на ЦП в этом интервале была близка к предельной.
- Сокращения si и so означают количество операций загрузки и выгрузки страниц из области подкачки (swap). Если эти операции выполнялись, то объем оперативной памяти недостаточно, стоит задуматься об увеличении.
- Если значение в колонке wa (ожидающие операции ввода-вывода) отлично от нуля (в то время как значения в колонках si и so нулевые), то это означает, что некоторая часть времени уходит на ожидание операций ввода-вывода, поэтому производительность некоторых процессов в системе ограничена ресурсами устройств ввода-вывода.

Если в отчете команды vmstat указано значительное время ожидания ввода-вывода (колонка wa), то рекомендуется вызвать команду iostat для получения более подробной информации.
Для получения информации о работе процессора удобнее использовать команду vmstat, а не iostat, так как ее построчный вывод проще анализировать, а в случае, когда к системе подключено много дисков, ее вывод намного менее объемный.

Для систем SMP в столбцах us, sy, id и wa указываются средние значения по всем процессорам (статистическую информацию об отдельных процессорах можно получить с помощью команды sar). Оптимальной является ситуация, когда процессор работает 100 процентов времени. Это условие всегда выполняется в однопользовательской системе, если в ней нет конкуренции за процессор. В общем случае, если суммарное время us + sy меньше 90 процентов, то считается, что производительность однопользовательской системы не ограничена ресурсами CPU. Однако если значение us + sy в многопользовательской системе превышает 80 процентов, то некоторые процессы часть времени находятся в состоянии ожидания. Это может отрицательно сказаться на времени ответа и скорости работы приложений.

В столбцах **procs** указывается следующее:

- **r**. Среднее количество процессов, которые каждую секунду находятся в *очереди выполнения*(т.е. стоят в очереди на выполнение процессором). В это количество не входят процессы, находящиеся в режимах ожидания различных событий (как то , ввод с терминала, дисковый и сетевой ввод-вывод и т.п.). Усредненное значение этой величины называется Load Average и его можно видеть в выводе команд top и uptime. Высокое значение r(Load Average) говорит о том, что системе потенциально не хватает производительности. Для всех систем, за исключением SMP, оно должно быть меньше пяти. В системах SMP это число должно быть не меньше следующего значения:

  5 x (Ntotal - Nbind)

  Здесь Ntotal – это общее число процессоров, а Nbind – это число процессоров, которые связаны с определенными процессами, например. Если это значение быстро растет, то проверьте правильность работы приложений. Некоторые системы могут нормально работать, даже если в очереди выполнения находится от 10 до 15 нитей, поскольку все зависит от назначения нитей и времени их выполнения.

- **b**. Количество процессов в *очереди ожидания*, заблокированных на операциях дискового ввода-вывода, которые требуют завершения. В этой очереди находятся нити, ожидающие освобождения ресурса или выполнения операции ввода-вывода. Кроме того, в эту очередь помещаются нити, ожидающие загрузки одной из своих страниц в оперативную память. Обычно это значение близко к нулю. Однако если число нитей в очереди выполнения увеличивается, то, как правило, длина очереди ожидания также увеличивается. Если в течение одной секунды одновременно было активировано несколько нитей, то длина очереди выполнения может быть довольно большой. При этом показатель использования процессора может быть довольно

низким, если эти нити были сразу же приостановлены. Если процессы в системе приостанавливаются из-за перегрузки памяти, то прирост нитей отражает именно значение в столбце b отчета vmstat, а не число нитей в очереди выполнения.

В столбцах **memory** указывается следующее:

- **free**. Значение отражает среднее число свободных страниц памяти. Страницей называется блок физической памяти размером 4 Кб.

В столбцах **swap** указывается следующее:

- **si**. Количество памяти выгруженной в области подкачки (swap).
- **so**. Количество памяти загруженной из области подкачки (swap).

В столбцах **io** указывается следующее:

- **bi**. Количество блоков информации прочитанных с жестких дисков в секунду (blocks/s).
- **bo**. Количество блоков информации записанных на жесткие диски в секунду (blocks/s).

В столбцах **system** указывается следующее:

- **in**. Число прерываний, поступавших от устройств каждую секунду в течение интервала сбора информации.
- **cs**. Число переключений контекста, выполнявшихся каждую секунду в течение интервала сбора информации. Все время работы процессора разделяется на логические кванты времени по 10 миллисекунд. Любая нить выполняется до тех пор, пока не истечет квант времени, пока она не будет замещена нитью с более высоким приоритетом, либо пока она добровольно не передаст управление другой нити. Когда управление передается другой нити, контекст, или рабочая среда, предыдущей нити сохраняется, а вместо него загружается контекст текущей нити. В операционной

системе предусмотрена очень эффективная процедура переключения контекста, поэтому на переключение контекста тратится лишь незначительная часть ресурсов. Однако при значительном росте числа переключений контекста, например, когда значение cs становится намного больше скорости дискового ввода-вывода и скорости передачи пакетов по сети, необходимо дополнительно проанализировать ситуацию.

В столбцах **cpu** указывается следующее:

- **us**. В столбце us указывается, какая доля времени ЦП (в процентах) была затрачена на работу в пользовательском режиме. Процессы UNIX могут выполняться в пользовательском или системном режиме (режиме ядра). При работе в пользовательском режиме процесс выполняется в рамках приложения. Ему не требуются ресурсы ядра для выполнения вычислений, управления памятью и настройки переменных.
- **sy**. В столбце sy указывается, какая доля времени процессора была затрачена на выполнение процессов в системном режиме. Это значение отражает ресурсы ЦП, которые были затрачены на выполнение процессов ядра и других процессов, использующих ресурсы ядра. Для получения доступа к ресурсам ядра процесс отправляет системный вызов, в результате чего он переключается в системный режим. Например, при выполнении операции чтения или записи данных в файл ресурсы ядра применяются для открытия файла, смещения указателя в нужную позицию и последующего чтения или записи данных (если файл еще не загружен в оперативную память).
- **id**. В столбце id указывается, какую долю времени (в процентах) процессор простаивал, не ожидая завершения локальной операции дискового ввода-вывода. Если ни одна нить не готова к работе (очередь выполнения пуста), то система запускает специальный процесс wait. В системе SMP нить wait может быть запущена на каждом

процессоре. В отчете команды ps(запущенной с опцией -k или -g 0) этот процесс указывается как kproc или wait. В однопроцессорной системе ИД этого процесса (PID) обычно равен 516. В системах SMP процесс kproc будет указан для каждого процессора в отдельности. Если в отчете ps указано, что на эту нить в целом затрачивается много времени, то это означает, что существуют большие промежутки времени, в течение которых ни одна нить не готова к работе и не ожидает запуска. Следовательно, система преимущественно простаивает, ожидая появления новых задач. Если нет ожидающих операций ввода-вывода на локальный диск, то все время, указанное как время ожидания, задает время простоя.

- **wa**. В столбце wa указывается доля времени, в течение которого процессор простаивал, ожидая завершения операции ввода-вывода на локальный диск. Если во время выполнения процесса wait в системе есть хотя бы один процесс, ожидающий выполнения ввода-вывода, то это время рассматривается как время ожидания завершения операции ввода-вывода. За исключением случаев, когда операции ввода-вывода выполняются асинхронно с процессом, процесс блокируется на время выполнения запросов на обмен данными с диском. После выполнения запроса на ввод-вывод процесс снова помещается в очередь выполнения. Чем быстрее выполняется обмен данными с диском, тем больше времени CPU будет тратиться на выполнение процессов. **Если значение wa больше 25 процентов, то можно сделать вывод, что дисковая подсистема плохо сбалансирована, либо в системе выполняется очень много дисковых операций ввода-вывода.** Если бы диски могли работать быстрее (wa маленький) то и выполнение процессов ускорилось бы также.

Экран разбит на шесть разделов: процесс (procs), память (memory), пейджинг (page), диски (disks), ошибки (faults) и процессор (cpu). Кратко рассмотрим каждый раздел, а затем углубимся в вопросы, имеющие наибольшее значение при

исследовании производительности.

```
root@web:vmstat
 procs memory page disks faults cpu
 r b w avm   fre flt re pi po fr  sr  ad0 pa0 in  sy   cs   us
sy id
 1 0 5 1223M 15M 595  0  1  1 648 391  0    0 159 1471 2492  2
3 95
```

Экран разбит на шесть разделов: процесс (procs), память (memory), пейджинг (page), диски (disks), ошибки (faults) и процессор (cpu). Кратко рассмотрим каждый раздел, а затем углубимся в вопросы, имеющие наибольшее значение при исследовании производительности.

Процессы

Под заголовком procs есть три колонки.

r
Количество процессов, ожидающих времени процессора. Они готовы к запуску, но не могут получить доступ к процессору. Если их много, то узким местом системы является процессор.

b
Количество процессов, блокированных в ожидании системного ввода или вывода. Обычно это ожидание доступа к диску. Такие процессы будут запущены сразу после получения нужных им данных. Если это число велико, узким местом является диск.

w
Показывает процессы, которые были запущены, но полностью вытеснены. Если процессы начинают вытесняться регулярно, размер памяти не соответствует заданиям, выполняемым в системе.

Память

FreeBSD разбивает память на фрагменты, которые называются *страницами*. Все страницы, выделяемые процессам, имеют одинаковый размер. Размер страницы зависит от аппаратной платформы и операционной системы. Система обрабатывает страницы целиком, например, если требуется вытеснить память в пространство свопинга (область подкачки), она вытесняется целыми страницами. Раздел memoryсостоит из двух колонок.

avm
Среднее количество страниц виртуальной памяти, используемых системой. Если это значение чрезмерно велико, значит, система активно расходует виртуальную память.

fre
Количество страниц памяти, доступных для использования. Если это значение чрезмерно мало, налицо проблемы с памятью.

Пейджинг

Раздел page показывает, как работает система виртуальной памяти. Внутренние механизмы виртуальной памяти — тайная наука, которую я не буду здесь описывать.

flt
Количество ошибок из-за отсутствия страницы (page faults), когда нужной информации нет в памяти и приходится загружать ее с диска, из пространства свопинга.

re
Количество страниц, восстановленных из кэша и повторно использованных.

pi
Сокращение от *pages in*. Показывает, сколько страниц было перемещено из физической памяти в пространство свопинга.

po
Сокращение от *pages out*. Показывает, сколько страниц было перемещено из пространства свопинга в реальную память.

fr
Количество освобождаемых страниц в секунду.

sr
Количество просмотренных страниц в секунду.

Вытеснение памяти в пространство свопинга — это совсем неплохо, но постоянное перемещение страниц из области подкачки в реальную память свидетельствует о нехватке памяти. Большие значения в колонках fr и flt могут свидетельствовать о чрезмерном количестве короткоживущих процессов, например CGI-сценарии запускают множество других процессов или слишком часто планируется выполнение некоторых заданий в cron.

Диски

Раздел disks показывает все диски по именам устройств. Числа, показанные здесь, — это количество дисковых операций в секунду, ценная подсказка для тех, кто стремится выяснить, как хорошо диски справляются с нагрузкой. При любой возможности следует распределять дисковые операции между различными дисками, а диски размещать на разных шинах. Если один диск загружен больше остальных, а система ожидает доступа к диску, то наиболее часто используемые файлы надо переместить с одного диска на другой. Одна из постоянно встречающихся причин высокой нагрузки на диски — это аварийное завершение программ с формированием дампа памяти на диске, которые могут перезапускать себя. Например, CGI-сценарий, содержащий ошибку и аварийно завершившийся с формированием дампа памяти на диске, всякий раз, когда кто-то щелкает на ссылке, может существенно увеличивать нагрузку на жесткий диск.

Если дисков много, можно заметить, что не все они появляются в выводе vmstat. Программа vmstat(8) разрабатывалась для работы

с экраном шириной 80 символов, поэтому она не может отобразить все диски, если их много. Однако если у вас широкий экран и вы не против того, чтобы вывод программы в ширину превысил ограничение в 80 символов, воспользуйтесь ключом -n, чтобы указать желаемое количество дисков для отображения.

Ошибки

В ошибках (faults) нет ничего плохого, они представляют собой всего лишь полученные системные прерывания. Плохо, когда их слишком много, но прежде чем заняться этой проблемой, необходимо знать, какое значение для вашей системы является нормальным.

in
Количество системных прерываний (запросов IRQ), полученных системой за последние пять секунд.

sy
Количество системных вызовов за последние пять секунд.

cs
Количество переключений контекста (context switches) за последнюю секунду или среднее число переключений в секунду с момента последнего обновления. (Например, если показания vmstat обновляются каждые пять секунд, в этой колонке отображается среднее число переключений контекста в секунду за последние пять секунд.)

Процессор

Наконец, в разделе cpu показано, сколько времени система потратила на выполнение пользовательских (us) и системных задач (sy) и сколько времени система бездействовала (id). Программа top(1) представляет эту же информацию в более дружественном формате, но только для текущего времени, тогда как vmstat позволяет просматривать деятельность системы за

длительный период.

Использование vmstat

Как же применить эту информацию? Прежде всего, проверьте первые три колонки и выясните, чего ждет система. Если она ожидает доступа к процессору (колонка r), значит, не хватает мощности процессора. Если система ждет завершения дисковых операций (колонка b), то узким местом являются диски. Если часто происходит вытеснение процессов (колонка w), то в системе не хватает памяти. Другие колонки помогут более детально исследовать причины нехватки этих трех видов ресурсов.

Непрерывное выполнение vmstat

Применяя vmstat, вы наверняка больше заинтересованы в получении информации за определенный период времени, нежели в изучении состояния системы в тот или иной момент. С помощью ключа -w и числа можно запустить программу vmstat так, что она будет непрерывно обновлять показания на экране через заданное число секунд. Параметры использования виртуальной памяти пересчитываются внутренними счетчиками системы каждые пять секунд, хотя другие счетчики обновляются непрерывно.

Тестирование (ключ -s)

Команда, вызванная с опцией -s, записывает в стандартный поток вывода итоговый отчет, который содержит суммарные значения показателей использования памяти, подсчитанные за все время, прошедшее с момента инициализации системы. Эту команду рекомендуется вызывать дважды: до запуска оцениваемой рабочей схемы и после выполнения этой рабочей схемы. Дальше нужно оценить разницу между значениями, полученными в двух отчетах команды.