Об одном подходе к мониторингу и последующей оптимизации элемента памяти грид-структуры, реализованном на основе системы dCache

В. Трофимов, П. Дмитриенко

e-mail: tvv@jinr.ru, Лаборатория информационных технологий, ОИЯИ, Дубна

Для массового хранения экспериментальных данных и результатов моделирования на ЦИВК ОИЯИ используется системы dCache [1] и XROOTd [2], получившие широкое распространение в WLCG. Обе системы не содержат в своём составе встроенных средств анализа эффективности работы, которых бы было достаточно для принятия решений по изменению аппаратной и программной конфигураций систем хранения данных. Использование разнородных систем усложняет задачу учёта распределения и анализа эффективности ресурсов дисковой памяти. Для её решения была создана система мониторинга ресурсов дисковой памяти, использующая NAGIOS [3] в качестве системы хранения и отображения накопленной информации.

На Рис.1 приведена иерархия уровней системы управления памятью в Грид на основе dCache.

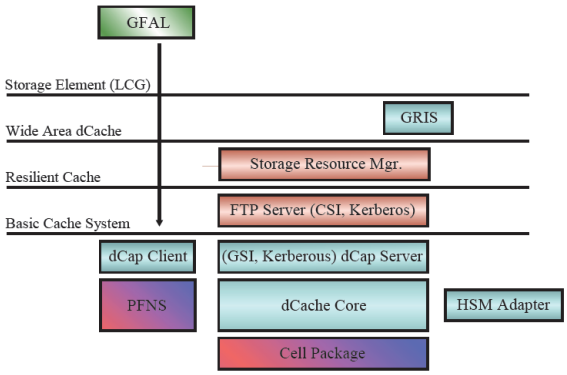

Рис. 1: Схема уровней системы управления памятью: GFAL – библиотека доступа к грид-файлам, Storage Element LCG – элемент памяти(SE), GRIS – информационная грид-служба, описывающая в данном случае SE, Wide Area dCache –слой dCache к которому обращаются внешние (относительно сайта) пользователи, Storage Resource Manager –компонент управляющий процессом резервирования памяти, Resilient Cache –динамический слой распределения файлов по пулам, GSI , Kerberos – средства аутентификации и авторизации, dCap Server – сервер внутреннего протокола чтения грид-файлов, PNFS – один из вариантов файловой метасистемы, используемый dCache, HSM Adapter – средства управления памятью третьего уровня, dCache Core, Cell Package – программное обеспечение системы dCache

Построение системы мониторинга дисковых ресурсов на базе NAGIOS

Выбор метрик

Параметры сайта отображаются во внешний мир информационной системой и описываются в GLUE схеме. Однако состав параметров выбран исходя из потребностей систем анализа заданий на сайте. Это набор недостаточен для целей оптимизации системы хранения информации. Чтобы не конфликтовать с задачами, решаемыми информационной системой, было принято решение сформировать собственное подмножество метрик для мониторинга.

Список параметров, подлежащие мониторированию и отображению должен отвечать обычным требованиям к такого рода наборам параметров: достаточностью, отсутствием избыточности по времени и составу. Кроме этого он должен отражать сложившиеся административные процедуры управления системами хранения в организации и служить информационной базой для принятия административных решений.

В качестве параметров были выбраны следующие:

1. **Объём и эффективность использования памяти**. Объём памяти утверждается совещанием экспериментов, причина включения в состав параметров этой административной цифры заключается во временном и объёмном разрыве потребностей экспериментов с действительно выделенным пространством. Правила хранения файлов, предусмотренные для уровня Tier2, постепенно меняются в сторону увеличения нагрузки. Тем не менее, до сих пор Tier2 не рассматривается, как ёмкость для постоянного хранения больших объёмов информации. Поэтому представляется целесообразным определять, как интенсивно используется память. Полезным будет знать количество и объём файлов, которые были однажды записаны и с тех пор ни разу не читались. Такие файлы можно рассматривать как информационный мусор и ставить перед экспериментом вопрос об их удалении. При этом надо учитывать, что процедура определения таких файлов – достаточно трудоемкое занятие. Необходимо также выбрать

эмпирически временной интервал после записи в течение которого ожидается хотя бы одно обращение к файлу на чтение.

2. **Количество процессов передачи, выполняемых одновременно.** В архитектуре системы dCache максимальное количество передач, выполняемых одновременно, определяется статически для каждого дискового пула. Если количество запросов на передачу превышает это число, то запрос становится в очередь ожидания. Опыт эксплуатации показал, что появление запросов в очереди скорее говорит о произошедшей ошибке, нежели о большом потоке запросов. Таким образом, следить за количеством запросов необходимо по двум причинам. Надо устанавливать предельное число запросов так, чтобы не допустить ситуацию ложной тревоги, с другой стороны как можно быстрее распознать ошибку, пока не накопились очереди. Выбрать это число можно только исходя из опыта эксплуатации, постоянно наблюдая за количеством процессов, поскольку ситуации с обращением к файлам изменяются в широком диапазоне.

3. **Уровень ошибок**. Статистика успешных и неуспешных завершений – стандартное свойство любой системы мониторинга. В качестве метрики выбрано количество операций, а не объём. Опыт показал, что такой выбор более информативен. Метрика берётся суммарно по всем виртуальным организациям. Детализация по каждой виртуальной организации нецелесообразна, поскольку цель наблюдения в данном случае - выявить проблемы системы, а не проблемы отдельных ВО, инициатива исправления которых лежит на самих ВО.

4. **Уровень нагрузки.** Трафик определяется для системы в целом и по каждой виртуальной организации. Для определения нагрузки на систему мониторинга количества передач недостаточно, поскольку, как было указано выше, по этому критерию определяются ошибки, а не нагрузка. Разделение на входящий и исходящий трафик для системы в целом никакой информации не даёт, поэтому не измеряется. С другой стороны сравнивать общий трафик системы нужно с суммарным трафиком по каждой ВО. Разделение трафика на чтение и запись имеет смысл для ВО, поскольку отвечает на вопрос баланса чтения и записи.

5. **Нагрузка на двери.** Архитектура dCache построена так, что весь трафик проходит через процессы, называемые двери (“door”). Как правило, они запускаются на отдельных серверах. Эта метрика интересна для определения необходимого количества дверей и вмешательства в том случае, если двери по какой-то причине перестают правильно отрабатывать запросы.

6. **Нагрузка на пулы.** Пулов в dCache намного больше дверей. Однако может возникнуть ситуация, когда к одному файлу обращается много процессов. Таким образом, на пуле может возникнуть перегрузка. Для пула целесообразно задать метрику – процент загрузки процессора и памяти. Определение количества процессов сложно и мало что даёт.

Выбор платформы для сбора информации, архитектура системы

В качестве платформы выбрана связка NAGIOS – MRTG [4]. Выбор NAGIOS для мониторинга обусловлен следующим:

А) Система уже используется для анализа аппаратных проблем сайта.

Б) Сведение аналогичных функций в рамки одной системы всегда удобнее, чем использование разнородных программ.

В) Система NAGIOS достаточно распространённая.

Г) Её функциональность удовлетворяет требованиям, а избыточность не отражается на удобстве использования.

Однако, протокол SNMP, который NAGIOS использует для сбора информации о состоянии компьютеров, недостаточен для того, чтобы получить информацию о состояния dCache. Поэтому в качестве агента выбран NRPE [5], стандартный для NAGIOS. В дополнении к связке, написаны скрипты Bash, которые служат интерфейсом к информационной системе dCache и базе данных лога dCache. Скрипты унифицированы и расширение их функции не представляет труда.

Выбор MRTG для отображения информации обусловлен следующим:

А) Система ориентирована на отображение информации об операциях записи и чтения, что в целом отвечает потребностям.

Б) Система распространённая и используется на многих сайтах.

В) Несмотря на то, что отображение более 2 переменных на одном графике не допускается, этот недостаток преодолевается организацией отображения.

Г) Система автоматически генерирует графы с разным разрешением по времени. Это сильно облегчает анализ.

Поскольку средствами NAGIOS отслеживается состояние оборудования сайта, серверы dCache исключены из общего мониторинга и перенесены в мониторинг системы хранения. Состояние дверей и пулов определяется через протокол SNMP. Для всех метрик, кроме дверей и пулов отключен механизм оповещения, поскольку процессы, отображаемые системой мониторинга медленные. При отображении количество ошибок умножается на 10. В противном случае на масштабе графика ошибки не видны. Графики объединяются в пары:

1. Объём памяти запрошенный – занятый по всем BO.
2. Объём памяти выделенный – занятый по всем BO.
3. Количество процессов передачи файлов в система – количество процессов в очереди.
4. Количество успешно завершённых операции – количество операций с ошибками по всей системе.
5. Сумма прочитанных и записанных байтов в система – та же сумма по BO.
6. Сумма записанных байтов – сумма прочитанных байтов по всем BO.

Примеры визуализации системы мониторинга приведены на рис.2-4.

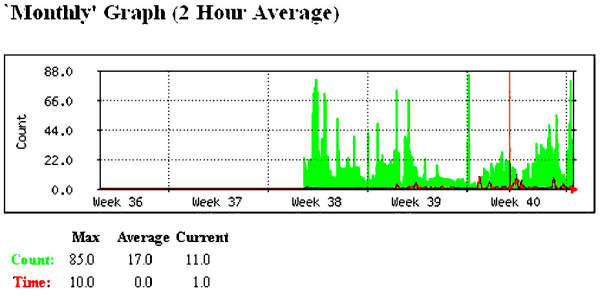

Рис. 2: Соотношение успешных и ошибочных операций для BO CMS

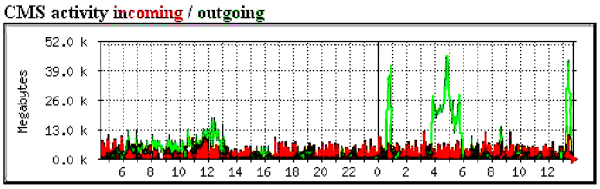

Рис. 3: Оперативный график объёмов чтения и записи для BO CMS на сайте ОИЯИ

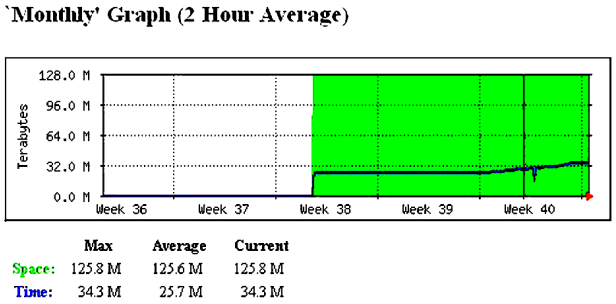

Рис. 4: Месячная динамика объёма хранения для BO CMS на сайте ОИЯИ

Заключение

На основании описанного подхода была построена система мониторинга, в которой сбор информации осуществляется через выделенный сервер NAGIOS, а отображение - через программу MRTG. Созданная система мониторинга используется исключительно для операторов (системных администраторов) с соответствующим ограничением доступа. Полученные результаты наблюдения дают возможность повысить эффективность использования выделенной памяти, а постоянный мониторинг ошибок - возможность своевременно реагировать на возникающие сбои.

Список литературы

[1] http://www.dcache.org

[2] http://xrootd.slac.stanford.edu/

[3] http://www.nagios.org/

[4] http://oss.oetiker.ch/mrtg/

[5]http://support.nagios.com/knowledgebase/officialdocs