

ВЫСОКОПРОИЗВОДИТЕЛЬНАЯ АРХИТЕКТУРА ХРАНИЛИЩА ДАННЫХ ДЛЯ СИСТЕМ МОНИТОРИНГА КОРПОРАТИВНЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ

Широкий А.А.

*Институт проблем управления им. В. А. Трапезникова РАН,
Россия, г. Москва, ул. Профсоюзная д.65
shiroky@ipu.ru,*

Натаров А.М.

*Волгоградский государственный университет, г. Волгоград, пр-кт. Университетский, 100
natarov.a.m@volsu.ru*

Аннотация: В работе предлагается решение проблемы достижения пределов производительности систем мониторинга IT-инфраструктуры в виде модификации архитектуры подсистемы хранения данных. Решение заключается в использовании каскада СУБД двух различных классов: резидентной – для обработки оперативных данных и колоночной – для долгосрочного хранения и анализа.

Ключевые слова: системы мониторинга, хранение данных, модель нагрузки, СУБД для системы мониторинга, оптимизация хранения данных

Введение

Системы мониторинга (СМ) отслеживают изменения состояния узлов (инфраструктуры, программных сервисов, компонентов корпоративных бизнес-приложений и др.), опрашивая или получая данные от них с заданной частотой. Современные СМ должны выполнять две основные функции: контролирующую и аналитико-прогнозную [1, 2].

Пока отслеживанию подвергались показатели работоспособности IT-инфраструктуры предприятия, требования к производительности СМ оставались низкими. Однако в современном бизнесе информационные системы не просто сопровождают рабочие процессы, а являются их основой. Теперь недостаточно просто знать, что сервер запущен и работает: необходимо отслеживать взаимодействие модулей, активность пользователей, и другие сервисные метрики [3]. В связи с этим требования к производительности СМ резко возросли и стала актуальной задача оптимизации мониторинга.

В настоящей работе предлагается вариант решения этой задачи с помощью модификации архитектуры СМ в части подсистемы хранения данных. В такой СМ данные мониторинга поступают в высокотранзакционную СУБД, хранящую их в оперативной памяти, а по истечении заданного администратором времени выгружаются в OLAP-СУБД для долговременного хранения и построения отчётности. Такой способ организации хранения данных позволяет получить практически неограниченные возможности по масштабированию системы мониторинга, гибко настраивать политики сбора и хранения данных, а также, несмотря на некоторое усложнение архитектуры, повысить отказоустойчивость системы в целом.

1 Оценки предельного числа узлов под наблюдением системы мониторинга

1.1. Ограничения производительности системы мониторинга

В общем виде задача оптимизации мониторинга информационных систем формулируется так: максимизировать число одновременно обслуживаемых метрик при заданных ограничениях на ресурсы. Здесь под ресурсами понимается машинное время, необходимое для функционирования СМ, объём памяти, необходимой для работы СМ, занимаемая ею доля пропускной способности сетевых интерфейсов.

На практике стоимость ресурсов неравноценна (системный администратор легко добавит ядро CPU и/или увеличит объём доступной RAM виртуальному серверу, но вряд ли сможет быстро расширить сетевой интерфейс). Число обслуживаемых метрик (и затраты на их обслуживание) также может изменяться – обычно в большую сторону и довольно быстро.

Поэтому оценивать потенциальную производительность системы мониторинга мы будем с помощью синтетического показателя – максимального числа «типовых» узлов, которые система может держать под наблюдением. Наблюдаемым узлом будем считать виртуальный или физический сервер, на котором развёрнут агент СМ. Состав наблюдаемых метрик может быть различным для каждого узла и для каждого конкретного узла определяется индивидуально. Ниже мы предложим два типовых сценария, соответствующего двум доминирующим подходам к организации мониторинга.

Существование ограничения на число наблюдаемых узлов определяется двумя группами факторов: программными (архитектура СМ, её конфигурация, и т. д.) и аппаратными (мощность сервера мониторинга, пропускная способность сетевого интерфейса и др.). На практике главными причинами становятся 1) необходимость принимать (и записывать) большое количество данных, приходящих маленькими порциями с наблюдаемых узлов; 2) сравнительно низкая производительность накопителей в режиме случайной записи [4, 5].

Поэтому вначале необходимо оценить, какой совокупный объём данных мониторинга СМ сможет обработать при заданной конфигурации сервера. После этого можно получить реалистичную оценку максимального числа наблюдаемых узлов в моделируемом сценарии.

1.2. Оценка числа отслеживаемых узлов в типовых сценариях мониторинга

Предложим два модельных сценария, оценим для них объём поступающих в СМ данных, а также произведем расчёт предельного числа узлов n для каждого из сценариев и при использовании различных типов накопителей.

Первый сценарий отражает типичную нагрузку на СМ образца 2005 года и является базовым для компаний, в которых ИТ и онлайн-сервисы не являются основой бизнеса. Пусть каждый узел отправляет пакет со значениями инфраструктурных метрик $c^f = 0,1$ Мбайта каждые $t = 30$ секунд.

Второй сценарий характерен для компаний с высокой степенью проникновения ИТ в бизнес-процессы (услуги онлайн, ИТ-компании, высокотехнологичные производства, поставщики сервисов с гарантированной работоспособностью). В таком сценарии к инфраструктурным данным добавляются сервисные метрики и частота сбора данных выше, чем в первом сценарии. Пусть каждый узел отправляет значения инфраструктурных и специализированных показателей каждые 10 секунд ($t^f = t^p = t = 10$). Объём пакета данных с инфраструктурными метриками установим для всех узлов одинаковым – $c^f = 0,1$ Мбайт. Размеры пакетов данных специализированных показателей c^p зависят от состава развёрнутых на узле сервисов и могут заметно различаться [2].

Положим для узлов с номерами от 1 до $m_1 = [0.3n]$ объём $c_i^p = 1$ Мбайт; от $[0.3n] + 1$ до $m_2 = [0.6n] - 2$ Мбайта; от $[0.6n] + 1$ до $m_3 = [0.9n] - 3$ Мбайта; а оставшиеся узлы с номерами от $m_3 + 1$ до n , будем считать, не направляют сервисные показатели. Тогда, с учётом вышеприведённых предположений, за одну минуту i -й наблюдаемый узел сгенерирует $c_i = 60(c^f + c_i^p)/t$.

Оценку предельного числа отслеживаемых системой мониторинга узлов будем выполнять по формуле $n = I/(c_{cp}^f + c_{cp}^p/t)$, где I – скорость записи в накопитель, а c_{cp}^f и c_{cp}^p – средние объёмы передаваемых узлами пакетов с данными инфраструктурных и специализированных показателей соответственно.

Значение I максимально при последовательной записи, но при этом редко достижимо на практике. При сохранении данных СУБД осуществляет проверку связанных с ними триггеров, которых может быть довольно много [6], поэтому в реальности накопитель будет работать в режиме смешанной нагрузки, чередуя операции чтения и записи.

В таблице 1 приведён расчёт предельного числа узлов в обоих сценариях для различных накопителей. Тестирование производительности накопителей при различных сценариях работы проводилось при помощи прикладной утилиты FIO (Flexible I/O Tester) [7].

Таблица 1. Оценка предельного числа отслеживаемых узлов в различных сценариях мониторинга

Тип накопителя	Значение I в режиме смешанной нагрузки	Оценка n (сценарий 1)	Оценка n (сценарий 2)
NVMe SSD	130,01 Мбайт/с	39 152	1 849
SATA SSD	86,2 Мбайт/с	25 961	1 225
SAS HDD	4,74 Мбайт/с	1 427	67
SATA HDD	1,58 Мбайт/с	475	23

Таким образом, реальные возможности СМ, использующей для хранения данных РСУБД, в окружении, требовательном к частоте мониторинга и генерирующем значительное число метрик, ограничены сотнями узлов. Одно из возможных решений заключается в модификации архитектуры подсистемы хранения данных.

2 Каскадная архитектура подсистемы хранения данных СМ

Основной идеей повышения производительности подсистемы хранения данных является перенос оперативных данных мониторинга (хотя бы за несколько минут) в динамическую память. В этом случае проверки всех метрик на соответствие заданным условиям произойдут до записи на накопитель, что позволит изменить режим его использования на чистую запись – в таком режиме производительность значительно возрастает.

Для хранения таких данных можно использовать in-memory хранилище типа «ключ-значение», (Redis, memcached, Tarantool и т. п.). Через заданный администратором СМ промежуток времени данные перемещаются в специализированную колоночную СУБД, например, ClickHouse (Yandex LLC, Россия), Kdb+ (KX Systems Inc., США) или Vertica (HP Inc., США). Применение NoSQL решений обусловлено спецификой задачи – данные мониторинга всегда отсортированы по времени.

Оценим потенциал СМ с каскадной архитектурой подсистемы хранения. Предложенная схема позволяет записывать данные на накопитель в пакетном режиме – таким образом, предел производительности подсистемы хранения данных существенно отодвигается (см. табл. 2).

Таблица 2. Оценка предельного числа отслеживаемых узлов в различных сценариях мониторинга для системы с каскадной архитектурой системы хранения

Тип накопителя	Значение I в режиме записи	Оценка n (сценарий 1)	Оценка n (сценарий 2)
NVMe SSD	403 Мбайт/с	121 375	5 732
SATA SSD	194 Мбайт/с	58 429	2 760
SAS HDD	110 Мбайт/с	33 129	1 565
SATA HDD	44,5 Мбайт/с	13 402	632

Аналитические запросы к СУБД на накопителе могут снижать скорость записи, но в любом случае не приведут к потере данных, поскольку те сохраняются в проксирующей СУБД в RAM. Объём оперативной памяти для хранения оперативных данных для одной тысячи узлов во втором сценарии за последние 60 секунд составит около 4 гигабайт.

Использование предложенной архитектуры хранения данных существенно увеличивает производительность подсистемы хранения данных СМ и при высоких нагрузках позволит эффективнее использовать имеющиеся серверные ресурсы.

Литература

1. Julian M. Practical Monitoring: Effective Strategies for the Real World. – O'Reilly Media, 2018. – 271p.
2. Jones D. Creating Unified IT Monitoring and Management in Your Environment. – Realtime Publishers, 2012 – 92p.
3. Alhamazani K., Ranjan R., Mitra K., et al. An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art // Computing Vol. 97. 2015, № 4. – P.357-377.
4. Croll A. Complete Web Monitoring. – O'Reilly Media, 2009. – 672p.
5. Solomon T., Zungeru A. M., Selvaraj R. Network traffic monitoring in an industrial environment // 2016 Third International Conference on Electrical, Electronics, Computer Engineering and their Applications (EECEA), Beirut, 2016, P.133-139.
6. A. Vacche, S. Lee Mastering Zabbix. – Packt Publishing, 2013. – 358p.
7. Github.com [Электронный ресурс]: Flexible I/O Tester – URL: <https://github.com/axboe/fio> (дата обращения: 08.04.2019).