

Оптимальный выбор системы мониторинга для различных типов ИТ-инфраструктур

Н.Б. Лазарева

Тихоокеанский государственный университет, Хабаровск

Аннотация: В данной статье рассмотрена проблема выбора системы мониторинга для ИТ-инфраструктуры. Проведен сравнительный анализ систем мониторинга, применительно к различным типам ИТ-инфраструктур и выбрано решение под конкретную ИТ-инфраструктуру компании.

Ключевые слова: мониторинг, инфраструктура, Zabbix, Grafana, Kubernetes, метрики.

В рамках инициативы по внедрению системы мониторинга для ИТ-инфраструктуры внутри компании возникла проблема выбора системы мониторинга. На рынке существует множество решений [1], из которых необходимо выбрать и внедрить одну или несколько систем мониторинга, покрывающих всю инфраструктуру компании.

Инфраструктура компании базируется преимущественно на базе Kubernetes-кластера, где развернута основная часть всех приложений. Также есть ряд виртуальных машин, в которых запущены различные приложения, работающие как в среде контейнеризации, так и в виде стандартно установленных пакетов.

Сетевой и серверный уровень полностью скрыт и недоступен, иначе говоря, имеем дело с определенным облачным решением.

Это довольно стандартная схема современной ИТ-инфраструктуры, поэтому вопрос выбора системы мониторинга в данном случае не является задачей, специфической для конкретной компании, а носит универсальный характер, позволяя применить этот опыт во многих других компаниях с подобными инфраструктурами.

В современном ИТ-мире существует большое количество систем, созданных для сбора и/или хранения метрик с различных устройств,

сервисов, приложений и так далее. Однако, в некоторых случаях они не являются взаимозаменяемыми и часто под определенные типы ИТ-инфраструктур следует выбирать ту или иную систему.

Рассмотрим примеры инфраструктур, для которых далее будем определять наиболее подходящую систему мониторинга.

Тип 1. Классическая инфраструктура с разнородными устройствами, такими как персональные ПК, серверы, сетевое оборудование, устройства печати, сканирования и прочее. Сюда же можно отнести программное обеспечение, установленное на ПК и серверах. В этом случае способ получения метрик с каждого устройства и программного продукта может быть абсолютно уникальными и не похожим на другие.

Тип 2. Облачная инфраструктура с виртуальными машинами, в которой сетевой уровень обычно скрыт, и из доступного аппаратного обеспечения могут быть сервера, иногда только с доступом через специальное ПО. Виртуальные машины содержат в себе операционные системы и ПО, с которого можно получать метрики. Это могут быть как внутренние сервисы операционной системы, так и разного рода приложения. В такой инфраструктуре прослеживается унификация способов получения метрик из-за меньшего разнообразия объектов.

Тип 3. Облачная инфраструктура с системами оркестрации контейнеров (например, Kubernetes [2]). Как и при использовании типа 2, сетевой уровень обычно скрыт, и из доступного аппаратного обеспечения могут быть сервера, иногда только с доступом через специальное ПО. В такой инфраструктуре система оркестрации контейнеров может быть выполнена поверх виртуальных машин или использоваться, как готовое облачное решение без доступа на нижний уровень. В таком случае достигается максимальная унификация способов сбора метрик, но сами способы несколько отличаются от способов, типичных для типов 1 и 2.

Часто существуют комбинированные инфраструктуры, однако использование для их мониторинга какой-либо одной системы обычно не является хорошей практикой. Рекомендуется разделять системы сбора метрик для гибкости и оптимизации конфигурации.

Рассмотрим некоторые наиболее известные системы мониторинга и определим, для какого типа инфраструктур они подходят.

Система мониторинга Zabbix [3] появилась достаточно давно и постепенно обрела поддержку сбора метрик с широкого ассортимента устройств, что делает ее крайне удобной для систем 1-го типа. Так, система позволяет реализовать Push и Pull модели сбора метрик (рис.1).

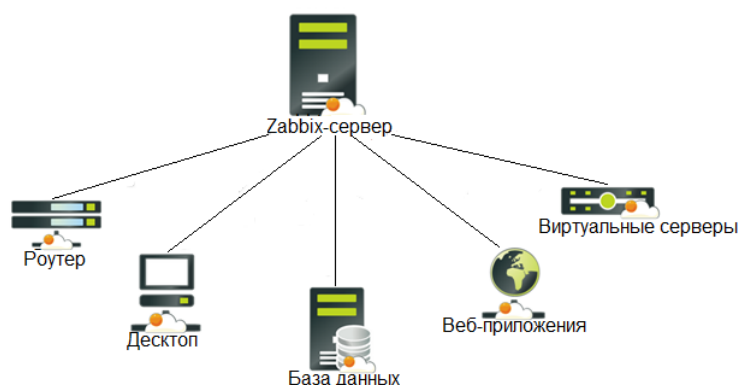


Рис. 1. – Высокоуровневое представление сбора метрик Zabbix

Это значит, что метрики можно как отправлять в систему мониторинга внешними средствами, так и собирать их, отправляя запросы от имени самой системы мониторинга. Для сбора метрик с серверов и персональных ПК очень удобен нативный Zabbix-агент, компонент, который устанавливается в систему и начинает сбор тех метрик, которые указаны в конфигурационном файле. Агент очень гибок и позволяет расширять его функциональность внешними скриптами. Zabbix может отправлять запросы на получение метрик любым устройствам, которые поддерживают внешние запросы подобного типа. Для типа 2 такая система так же подходит, однако часто в

таких инфраструктурах программное обеспечение самостоятельно генерирует метрики и готово их отправлять в систему сбора метрик. При этом, количество таких метрик может быть довольно большим, и использовать Zabbix с агентом становится сложнее из-за проблем с производительностью – Zabbix использует классическую базу данных для хранения метрик, и его производительность напрямую зависит от производительности базы данных. При больших размерах инфраструктуры использование Zabbix становится не оптимальным. Кроме этого, Zabbix ориентирован на настройку через Web-интерфейс и плохо интегрируется с подходом IaaS, когда вся конфигурация находится, например, в Git [4].

Graphite - это система хранения и обработки метрик. Для того, чтобы собрать метрики, нужно развернуть точку доступа к этому хранилищу, например Carbon, и отправлять в нее метрики. Для отправки можно использовать промежуточное ПО, такое, как Statsd или Telegraf [5], которые могут быть установлены непосредственно рядом с приложением, которое отправляет метрики к ним. В этом случае приложению ничего не известно о том, как настроен мониторинг, известно только, что метрики нужно отправлять этому промежуточному компоненту. Statsd или Telegraf также могут производить предварительную обработку метрик перед отправкой в систему хранения. Graphite [6] обладает высокой производительностью, напрямую зависящей от дисковой подсистемы. Быстрые SSD-диски позволяют Graphite обрабатывать огромное количество метрик (рис.2). Очень важное преимущество состоит в мощнейшем математическом аппарате – благодаря Graphite, в популярном средстве визуализации данных Grafana [7] возможно использовать различные преобразования метрик, которые, благодаря производительности системы, выполняются очень быстро. Именно поэтому модуль работы с Graphite всегда поставляется с Grafana по умолчанию. Не менее важным преимуществом Graphite является простота

обслуживания – все метрики в системе хранятся как файлы и их легко модифицировать/удалять.

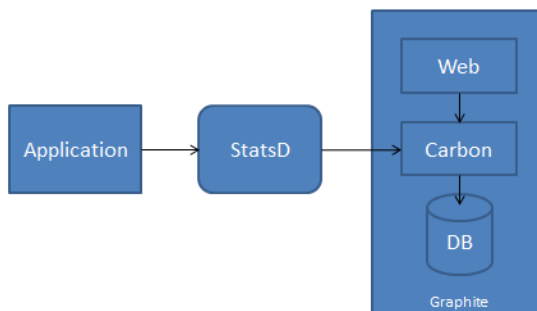


Рис. 2. – Высокоуровневое представление сбора метрик Graphite

Недостатками Graphite является сильная зависимость от производительности системы хранения: SSD-диски высокой производительности довольно дороги, поэтому при ограниченном бюджете такая система может работать медленно на более медленных дисках. Также нет возможности собирать метрики с различных устройств – для этого нужны промежуточные решения, способные запрашивать метрики. Такая система больше подходит для инфраструктур типа 2, в которых приложения генерируют метрики и готовы их отправить в совместимом формате. Стоит учитывать, что развертывание Graphite и Carbon - это нетривиальная задача, требует больше времени и опыта.

Система InfluxDB также, как и Graphite, является в первую очередь системой хранения и обработки метрик, и может быть использована в связке с Statsd или Telegraf (рис.3).

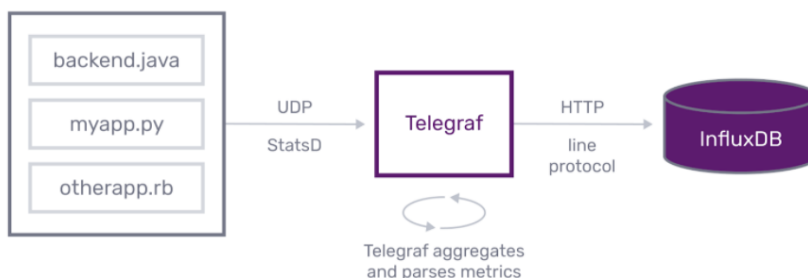


Рис. 3. – Высокоуровневое представление сбора метрик InfluxDB

Основное отличие от Graphite состоит в том, что InfluxDB является TSDB [8], написан на языке Golang и при большом количестве метрик показывает лучшую производительность. Отсюда вытекают и недостатки – TSDB сложнее обслуживать, математический аппарат предоставляет намного меньше функционала и построение визуализации в Grafana может быть сложнее. Хотя InfluxDB предоставляет SQL-совместимый интерфейс, он упрощен и не предоставляет всей функциональности. Как и Graphite, InfluxDB больше подходит для инфраструктур типа 2, в которых приложения генерируют метрики и готовы их отправить в совместимом формате, но имеет большую производительность при меньшей гибкости.

Система мониторинга Prometheus [9] является относительно новой на рынке и была разработана специально под инфраструктуры типа 3 – инфраструктуры, использующие Kubernetes в качестве оркестратора контейнеров. Идея состоит в том, что каждое приложение, развернутое внутри Kubernetes, предоставляет метрики по определенной ссылке – Prometheus же опрашивает эти ссылки и забирает метрики (рис.4).

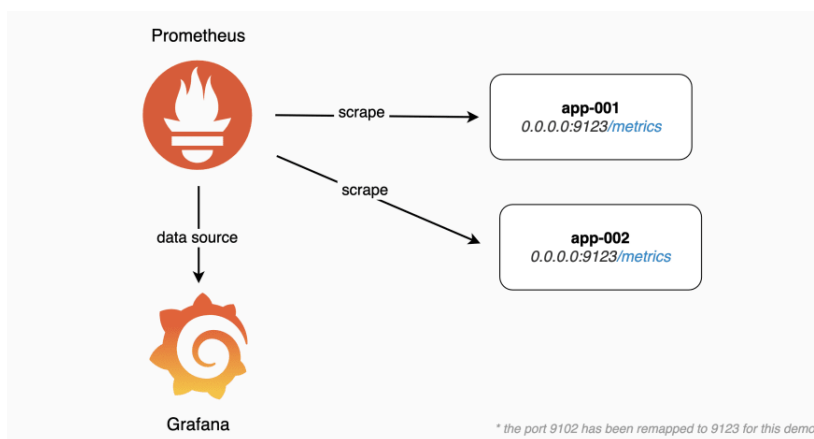


Рис. 4. – Высокоуровневое представление сбора метрик Prometheus

Как и InfluxDB, Prometheus является TSDB и написан на Golang, поэтому обладает высокой производительностью, но меньшим удобством в обслуживании и простым математическим аппаратом. Данную систему

мониторинга можно использовать в инфраструктурах типа 2, однако отправлять метрики в Prometheus нужно с использованием дополнительных компонентов, таких, как Prometheus Gateway. К недостаткам данной системы можно отнести необходимость подготовки приложений для отправки метрик – добавление специальной ссылки, что не всегда возможно. Также необходимо внести изменения в настройки объектов Kubernetes, чтобы Prometheus мог искать приложения с доступными для сбора метриками.

Применение Zabbix, Graphite и InfluxDB для инфраструктур типа 3 возможно, но только с помощью дополнительного ПО, которое будет забирать метрики у приложений, работающих в Kubernetes [10].

Проанализировав исходные данные, подберем систему(ы) мониторинга. В данном случае стоит выбрать комбинацию из двух различных систем мониторинга, поскольку есть условно две разные задачи – сбор метрик с кластера Kubernetes и приложений, запущенных в нем, а также сбор метрик с виртуальных машин и приложений, установленных и запущенных внутри них. Для сбора метрик с кластера Kubernetes выберем систему мониторинга Prometheus, так как она хорошо подходит под эту задачу: не требует дополнительного ПО, сама может забирать метрики, имеет высокую производительность и простоту обслуживания. Для сбора метрик с виртуальных машин выберем Graphite в качестве системы хранения данных и Telegraf в качестве клиента для их сбора и отправки. Telegraf позволяет гибко настроить сбор метрик с помощью разнообразных плагинов и отправлять их в Graphite с определенным интервалом, причем не только в сыром виде, но и в обработанном (дополнительные метрики вроде различных перцентилей, среднее значение за период и так далее). Graphite обеспечит отличные возможности для математических операций над метриками, а также простоту поддержки. Для развертывания Graphite обязательно нужно провести отдельный анализ количества метрик с оценкой объемов ресурсов и

характеристик инстанса, на котором он будет развернут, так как производительность Graphite прямо зависит от этих параметров. Так как Prometheus и Graphite позволяют интегрировать их с Grafana, не возникнет никаких проблем для отображения метрик в различном виде из обеих систем мониторинга.

В качестве дополнительного варианта можно рассмотреть систему, в которой Prometheus будет работать только на сбор метрик с Kubernetes-кластера, но не будет их хранить, а будет отправлять их в Graphite. Плюсы такого решения - единое хранилище метрик Graphite, мощные математические возможности для всех метрик, простота поддержки. Минусы - большая нагрузка на инстанс Graphite, требующая больших ресурсов и грамотного планирования, а также необходимость в промежуточном компоненте, который должен забирать метрики у Prometheus и отправлять их в Graphite.

Литература

1. Верещагина Е.А., Рудниченко А.К., Колесникова Д.С. Windows Management Instrumentation как способ мониторинга и аудита ИТ-инфраструктуры предприятия // Инженерный вестник Дона, 2019, №8. URL: ivdon.ru/ru/magazine/archive/n8y2019/6125/.
2. Принципы и архитектура Kubernetes. enterprise-k8s.arcgis.com URL: enterprise-k8s.arcgis.com/ru/latest/deploy/kubernetes-concepts
3. Смушкин В. А. Zabbix для мониторинга в ИТ – инфраструктуре // Форум молодых ученых, 2019, №4(32). URL: forum-nauka.ru/_files/ugd/b06fdc_41ab8211ee6c4bf4b18af1a9485eb145
4. Тармосина А.С., Пчелинцев А.Н. Внедрение системы контроля версий GIT – компании // Синергия наук, 2017, №18. URL: synergy-journal.ru/archive/article1497

5. Soppelsa, F., Kaewkasi, C. Native Docker Clustering with Swarm. – Packt Publishing Ltd, 2016. – P. 280.
6. Dixon, J. Monitoring with Graphite: Tracking Dynamic Host and Application Metrics at Scale. – O'Reilly, 2017. – P. 290.
7. Quintero, D., Dhar, S., Huertas, L. C. Software Defined Data Center with Red Hat Cloud and Open Source IT Operations Management. – IBM Redbooks, 2020. – P. 420.
8. Time series database (TSDB) internetofthingsagenda. techtarget.com URL: internetofthingsagenda.techtarget.com/definition/time-series-database-TSDB (дата обращения 19.04.2022)
9. Bastos, J., Araujo, P. Hands - On Infrastructure Monitoring with Prometheus. – Pakt Publishing Ltd, 2019. – P. 430.
10. Лазарева Н.Б., Ловцова Н.Н. Автоматизация получения доступа к базам данных для компонентов в среде Kubernetes предприятия // Инженерный вестник Дона, 2021, №3. URL: ivdon.ru/ru/magazine/archive/n3y2021/6844/.

References

1. Vereshhagina E. A., Rudnichenko A. K., Kolesnikova D. S. Inzhenernyj vestnik Dona, 2019, №8. URL: ivdon.ru/ru/magazine/archive/n8y2019/6125/.
 2. Principy` i arxitektura Kubernetes. [Kubernetes architecture and principals] URL: enterprise-k8s.arcgis.com/ru/latest/deploy/kubernetes-concepts
 3. Smushkin V. A. Forum molody`x ucheny`x, 2019, №4 (32). URL: forum-nauka.ru/_files/ugd/b06fdc_41ab8211ee6c4bf4b18af1a9485eb145
 4. Tarmosina A.S., Pchelincev A.N. Sinergiya nauk, 2017, №18. URL: synergy-journal.ru/archive/article1497
 5. Soppelsa, F., Kaewkasi, C. Native Docker Clustering with Swarm. – Packt Publishing Ltd, 2016. P. 280.
-



6. Dixon, J. Monitoring with Graphite: Tracking Dynamic Host and Application Metrics at Scale. O'Reilly, 2017. P. 290.
7. Quintero, D., Dhar, S., Huertas, L. C. Software Defined Data Center with Red Hat Cloud and Open Source IT Operations Management. IBM Redbooks, 2020. P. 420.
8. Time series database (TSDB) internetofthingsagenda. URL: internetofthingsagenda.techtarget.com/definition/time-series-database-TSDB (accessed 19/04/2022)
9. Bastos, J., Araujo, P. Hands On Infrastructure Monitoring with Prometheus. Packt Publishing Ltd, 2019. P. 430.
10. Lazareva N.B., Lovczova N.N. Inzhenernyj vestnik Dona, 2021, №3. URL: ivdon.ru/ru/magazine/archive/n3y2021/6844/.