influxdata®

AN INFLUXDATA CASE STUDY

How Wayfair Uses InfluxDB to Gain Visibility into System Performance and Understand Site User Experience

Jim Hagan
Manager, Wayfair
Mike Bell
Manager, Wayfair

wayfair

APRIL 2018

Company in brief

Founded in 2002 by Steve Conine and Niraj Shah, Wayfair is one of the world's largest online destinations for the home. It carries a broad portfolio of brands including its own Wayfair brand, Joss & Main, All+Modern, Birchlane, Perigold, and many others. Wayfair helps people find the perfect product at the right price. Their extensive selection and superior customer service coupled with the convenience of online shopping, make it easier than ever before to find exactly what you want for your home at a price you can afford.

Wayfair's growth is driving its complex infrastructure. Its technical team is working strategically with InfluxData to ensure the Wayfair implementation is scalable, robust, and in line with the InfluxDB's future direction. Wayfair is also providing a rich set of case studies to help drive InfluxDB further in its support of features that larger enterprises need.

Wayfair has contributed to various open source projects including Graphite and InfluxData's metrics collection agent, Telegraf. In the true spirit of open source, Wayfair's architecture involves a large number of collaborations and ongoing mutual feedback to help shape and improve future releases.

Case overview

Wayfair needed to efficiently monitor performance across their systems, which are spread across three major data centers. The data is used by their developers, business stakeholders, and internal alerting engine. Their 24/7 Ops Monitoring Center is using this data to constantly analyze the vital signs of Wayfair's IT infrastructure and storefront operations. Rapid growth led the company to rethink its time series infrastructure. Their existing Graphite solution failed to scale with growth demands in terms of ingest rate, storage, high availability, and it required major engineering time investment to maintain core functionality.

Wayfair chose InfluxData to monitor system metrics & events across data centers, and to perform real user monitoring (RUM) to understand user experience on their e-commerce site. The goal is to marry these with business process events and provide better business insight and competitive advantage. Wayfair use Kafka MirrorMaker to replicate the data to all three locations and have three six-node InfluxDB Enterprise clusters dedicated to different workloads.

> *“As Wayfair has grown and matured its software development and data center operations over the past decade, and particularly over the last five years, we have embraced the principle of providing maximum visibility into our processes and systems.”*

Jim Hagan, Manager

The business problem

Wayfair’s pursuit of process and system visibility has led to a very large logging and time series infrastructure built to receive a constant stream of application logs and metrics.

As a company that has experienced hyper growth over the last five years, Wayfair often needs to speedily adopt solutions and integrate them into mission critical operations roles. Graphite was relatively easy for their bootstrapped engineering team to adopt and deploy quickly. Its schema-less model meant developers rapidly embraced it as they could send data in a format meaningful to them, without

checking with a central gatekeeper. This schema-less environment also allowed the database to become quite unruly, however, since there was very little means to analyze and track the cardinality of various metrics.

While Graphite is a powerful open source package, its active development cycle has been on the wane for a number of years. During this time, Graphite's user base has drifted away to other solutions such as Prometheus and InfluxDB. Wayfair did their best to have Graphite work for them: their approach was to throw a considerable amount of compute and storage resources at it in order to force scalability, and a considerable amount of technical debt in maintaining arcane configurations and heavy handed cleanup scripts. The amount of engineering resources tied up in keeping the system stable and performant was relatively high, which meant those same engineers have not been able to work on higher value projects.

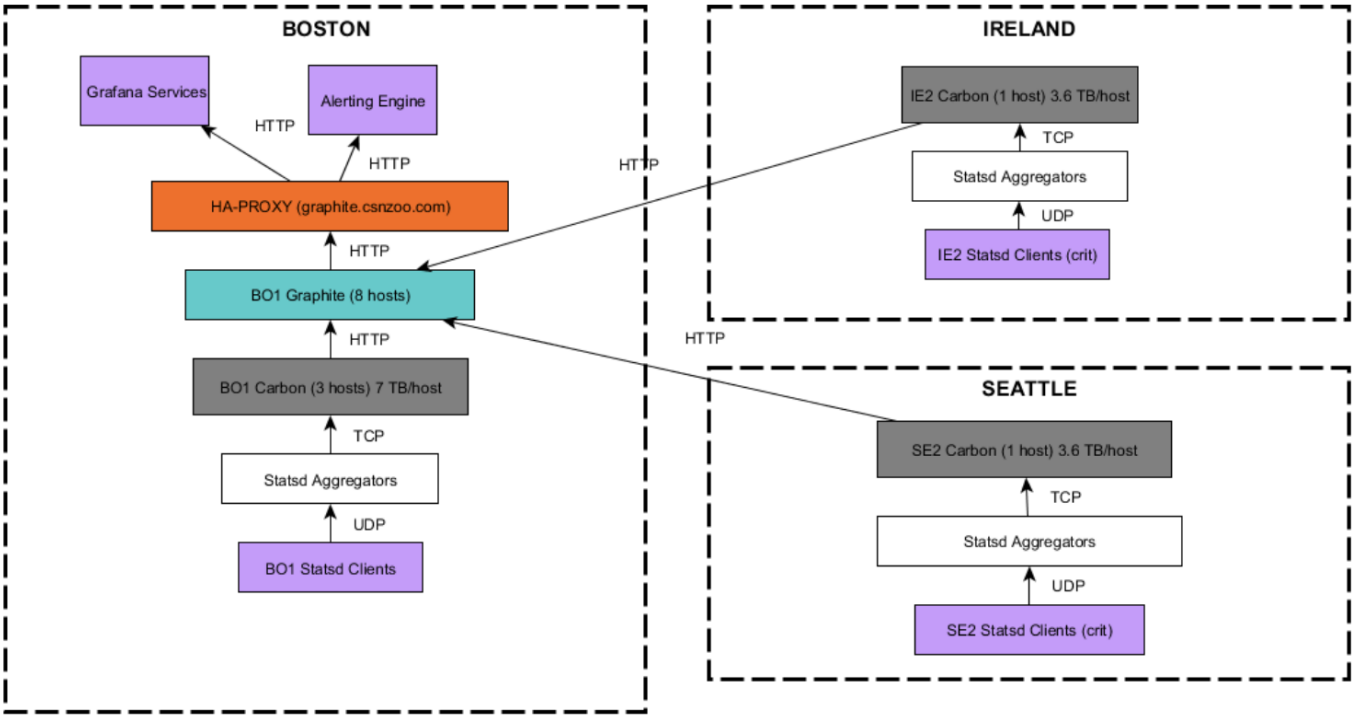
Wayfair decided it was time for a bolder move: to rethink time series data in their architecture and seek newer platforms built on high availability principles — one that was tailor-made for the type of distributed pipeline **they have while providing necessary the resilience and HA.**

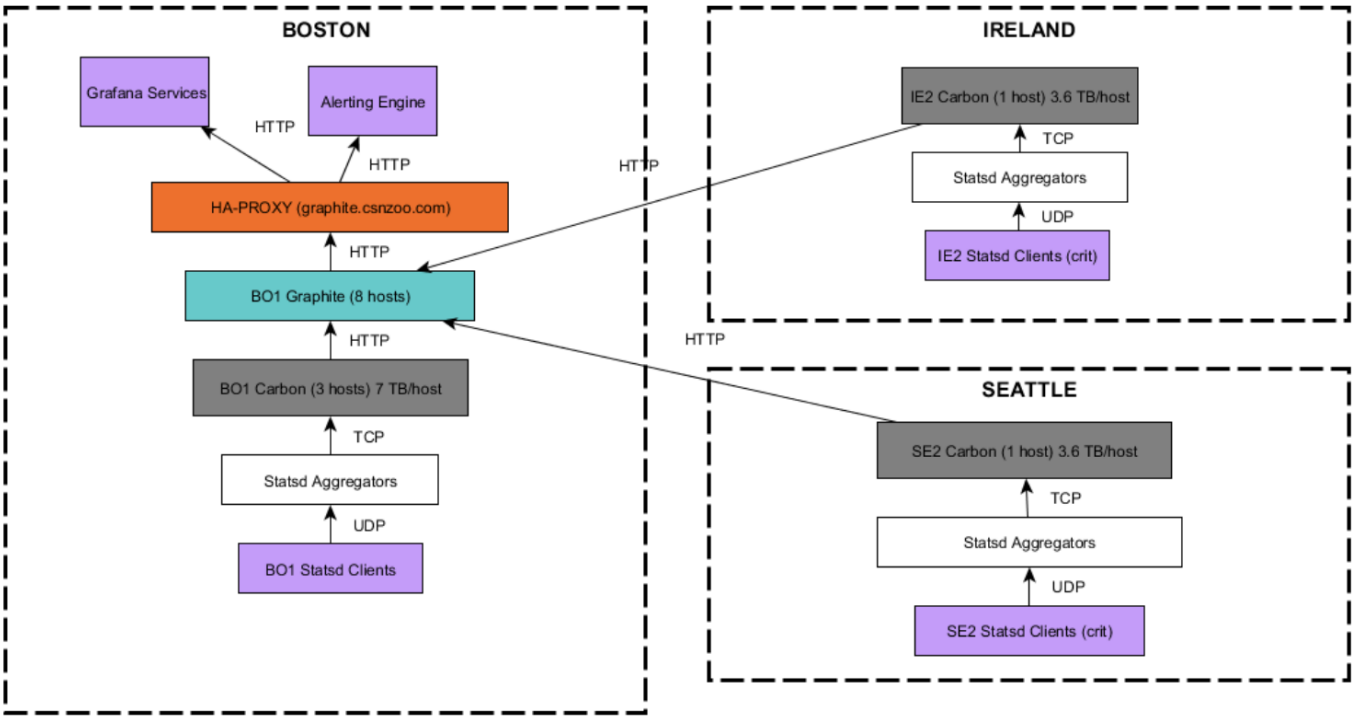
The technical problem

As with many open source projects, Wayfair's teams were able to move rapidly with Graphite. It can be very fast for write and read, but works within very fixed scalability constraints. Their implementation is coupled with the StatsD aggregation process. Since this uses an upstream hash ring to route metrics, it had always been a challenge to refactor the storage footprint of a Graphite/carbon cluster. There aren't "relocate shard", "backup shard", or "restore shard" operations in the Graphite/carbon world (mainly because there is no proper sharding concept; each metric is essentially a shard).

Having no replication/redundancy with this stack hurt them in several ways: primarily, it was impossible to do the kind of on-the fly maintenance and hardware upgrades that require them to bring a node down for any length of time. This also meant that there was always high contention for a given metric as it only resides on a single host.

Though they were “getting by” with fewer than 10 carbon storage hosts across three data centers and using a multi-layered Graphite proxy to merge data from remote data centers into a single web service, “getting by” wasn’t adequate. They had many slow queries with frequent timeouts (the opposite of high availability). They were also continually hitting the storage wall until they had each carbon host attached to nearly 7 TB of direct attached storage. Not surprisingly, their storage team eventually indicated that this model wasn’t scalable.





The "Legacy" Graphite Cluster in Service Until Early 2017

The cluster shown above relied heavily on cross-data center HTTP calls using their Boston Graphite layer as a direct proxy for remote data center carbon storage layers. Provisioning compute and storage was more of a challenge in the remote data centers.

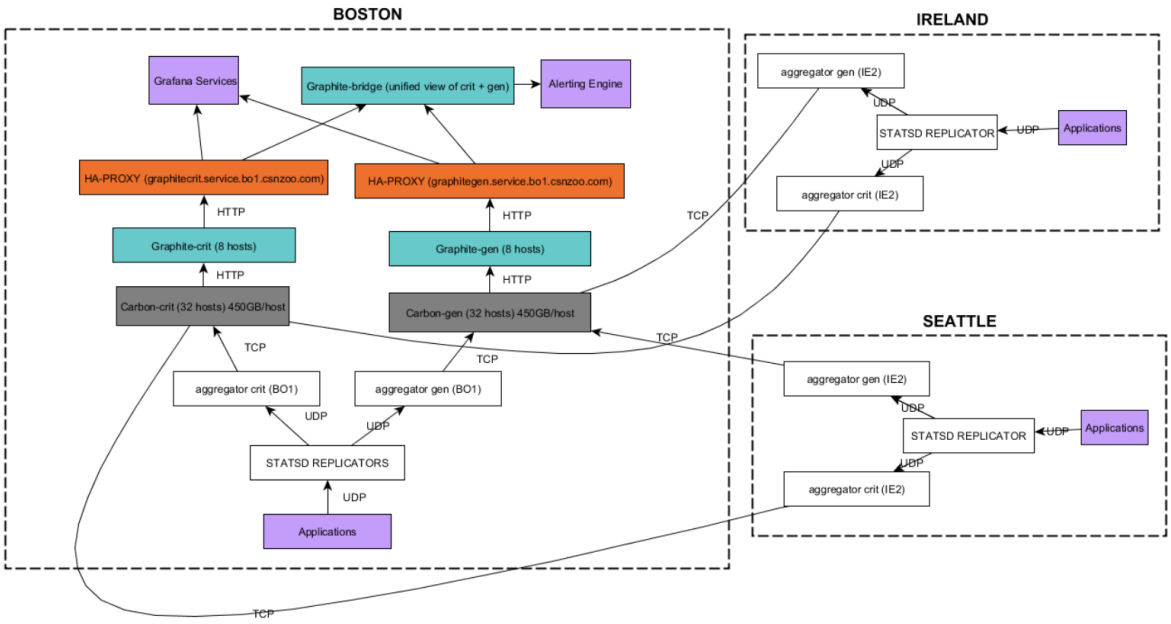
In an effort to prepare for the massive traffic they received during 2017's Cyber 5 weekend, they built out the cluster to 64 data nodes with only 0.5 TB attached to each node. Further, they split this group of nodes into two clusters of "critical" and "general" data. The critical data mostly fed their 24/7 network operations center team and their most customer-critical web service alerts.

That's a fair amount of compute power, but it addressed their main areas of system failure in the form of large contention for data at the TCP level of a given carbon host.

As there is no central index for each measurement, the carbon architecture requires a fair number of redundant searches, which leads to network and file system contention. In order to scale so wide (as Wayfair needed to do), the storage in its central Boston, Massachusetts data center had to be consolidated; meaning their data centers in Seattle, Washington and Ireland would not have their own Graphite/carbon storage instances. Instead, they sent their StatsD metrics cross-data center via UDP into the main data center. This element itself had a lot of overhead to monitor and wasn't robust against a major network outage or latency.

Specifications of their Graphite implementation (refactored for Holiday 2017) were as follows:

- Data points/second: Up to 1 Million
- Total storage allocated: 35 TB
- Carbon storage hosts: 64
- Graphite web server hosts: 40
- StatsD pipeline hosts: 50
- Retention: 2 Months (at 10 sec. resolution)



The "Super Graphite" Cluster Built for Holiday 2017

Through their efforts at scaling Graphite, they reached concrete conclusions:

1. Graphite/carbon does not support true clustering (replication and high availability).
2. Graphite/carbon has a unique storage model, and although often fast due to its fixed temporal resolution, requires pre-allocation for each series, leading to storage infrastructure that is difficult to manage. It can even lead to rapid and catastrophic depletion of disk resources.
3. Graphite/carbon does not provide an out-of-the-box data pipeline solution. They've relied on their highly optimized in-house StatsD package. While this robust service is very good at what it does, the technical debt and opportunity costs in maintaining such a tool for internal consumption is high. For example, it currently lacks any sort of I/O support for Kafka. UDP is its only I/O mechanism. If they wanted more I/O options, they would need to build them.
4. The Python code underlying Graphite and carbon has maxed out its performance potential.
5. There is no straightforward way to move metrics from one data node to another (modern clustered solutions support this via shard relocation and backup/restore operations).

With the above conclusions in mind, Wayfair set out to evaluate a number of replacement systems. They found that InfluxDB from InfluxData offered the right combination of attributes that would take them to the next level. They had a number of core technical requirements that were honed over the years of working with Graphite:

- Support hundreds applications sending metrics from multiple data centers
- Handle millions of points per second
- Support for non-blocking I/O
- Data availability within seconds (for alerts and graphs)
- Tolerance for rapid spikes in traffic at certain times of day (3x to 5x average peak traffic)
- Easy for developers to integrate into their code
- Retention periods of two months to over a year
- Support for data replication and shard management.

The solution

> *We set out over a year ago to evaluate a number of replacement systems. After an initial investigation, we concluded that InfluxDB from InfluxData offered us the right combination of attributes that we felt would take us to the next level."*

Jim Hagan, Manager

With the above conclusions in mind, Wayfair set out to evaluate a number of replacement systems. They found that InfluxDB from InfluxData offered the right combination of attributes that would take them to the next level. They had a number of core technical requirements that were honed over the years of working with Graphite:

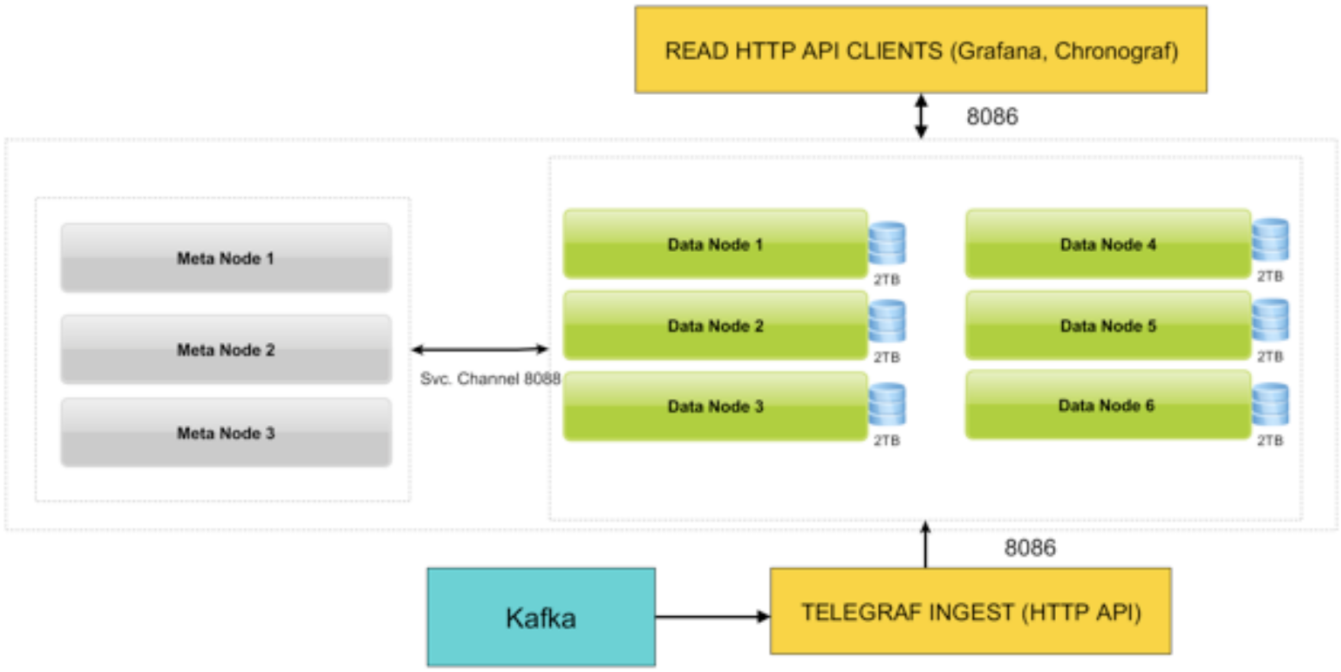
- Support hundreds applications sending metrics from multiple data centers
- Handle millions of points per second
- Support for non-blocking I/O
- Data availability within seconds (for alerts and graphs)
- Tolerance for rapid spikes in traffic at certain times of day (3x to 5x average peak traffic)
- Easy for developers to integrate into their code
- Retention periods of two months to over a year
- Support for data replication and shard management.

Why InfluxDB?

Through Wayfair's evaluation and proof-of-concept phase, InfluxDB proved capable of meeting their core technical requirements. It is architected in such a way that allows balancing horizontal and vertical scaling approaches for both compute and storage. Wayfair's in-house time series technology review identified key attributes that offered substantive advantages over the status quo:

1. InfluxDB is undergoing rapid performance improvements and hardening.
2. InfluxDB supported true clustering technology (replication, high availability, shard management).
3. InfluxDB and all related packages from InfluxData are written highly optimized in Go, a language that fully exploits multi-core environments.
4. InfluxDB was built with an emphasis on efficient and scalable storage. Knowing that many of the metrics in a time series system will be sparse and/or ephemeral, InfluxDB per data point compression will generally be higher than carbon's as there is no pre-allocation of storage for stored data points.
5. InfluxDB has an active community and an ecosystem of related tools for building enterprise-wide installations. Wayfair found it more appealing, as an organization, to contribute to an occasional Telegraf plugin (a plugin-based data streaming utility) that others in the industry are using and improving, than to maintain a completely proprietary codebase for the backbone of their data pipeline.
6. InfluxDB has a growing customer base from a diverse set of companies in IT, IoT, financial analytics, and data science. Wayfair felt this diverse customer set will allow this platform to evolve into a general analytics platform that can support Wayfair applications as broad as IT infrastructure monitoring, application performance monitoring, as well as data science and market research.

In the proof-of-concept chart below, data nodes contain shards, meta nodes manage the cluster state (shard assignments). Telegraf is the gateway into the cluster.



General Architecture for InfluxDB Proof of Concept Cluster at Wayfair

As part of their migration process, Wayfair worked closely with their development teams to retool their code so that they could send metrics to both Graphite (via StatsD format) and InfluxDB (via line protocol format). Using feature toggles, each application can write in either format or both at the same time. This allowed them to do head-to-head testing of query performance as well as load test their new InfluxDB cluster.

Why replace Graphite with InfluxDB

As part of their migration process, Wayfair worked closely with their development teams to retool their code so that they could send metrics to both Graphite (via StatsD format) and InfluxDB (via line protocol format). Using feature toggles, each application can write in either format or both at the same time. This allowed them to do head-to-head testing of query performance as well as load test their new InfluxDB cluster.

Graphite Limitations

- No clustering
- Storage infrastructure that is difficult to manage
- No out-of-the-box data pipeline solution
- Graphite maxed out its performance potential
- No shard relocation & backup/restore

InfluxDB Advantages

- Rapid performance improvements & hardening
- Supports true clustering
- Written in Go, a language that fully exploits multi-core environments
- Built with an emphasis on efficient & scalable storage
- Active community & an ecosystem of related tools for building enterprise-wide installations
- Growing customer base from a diverse set of companies

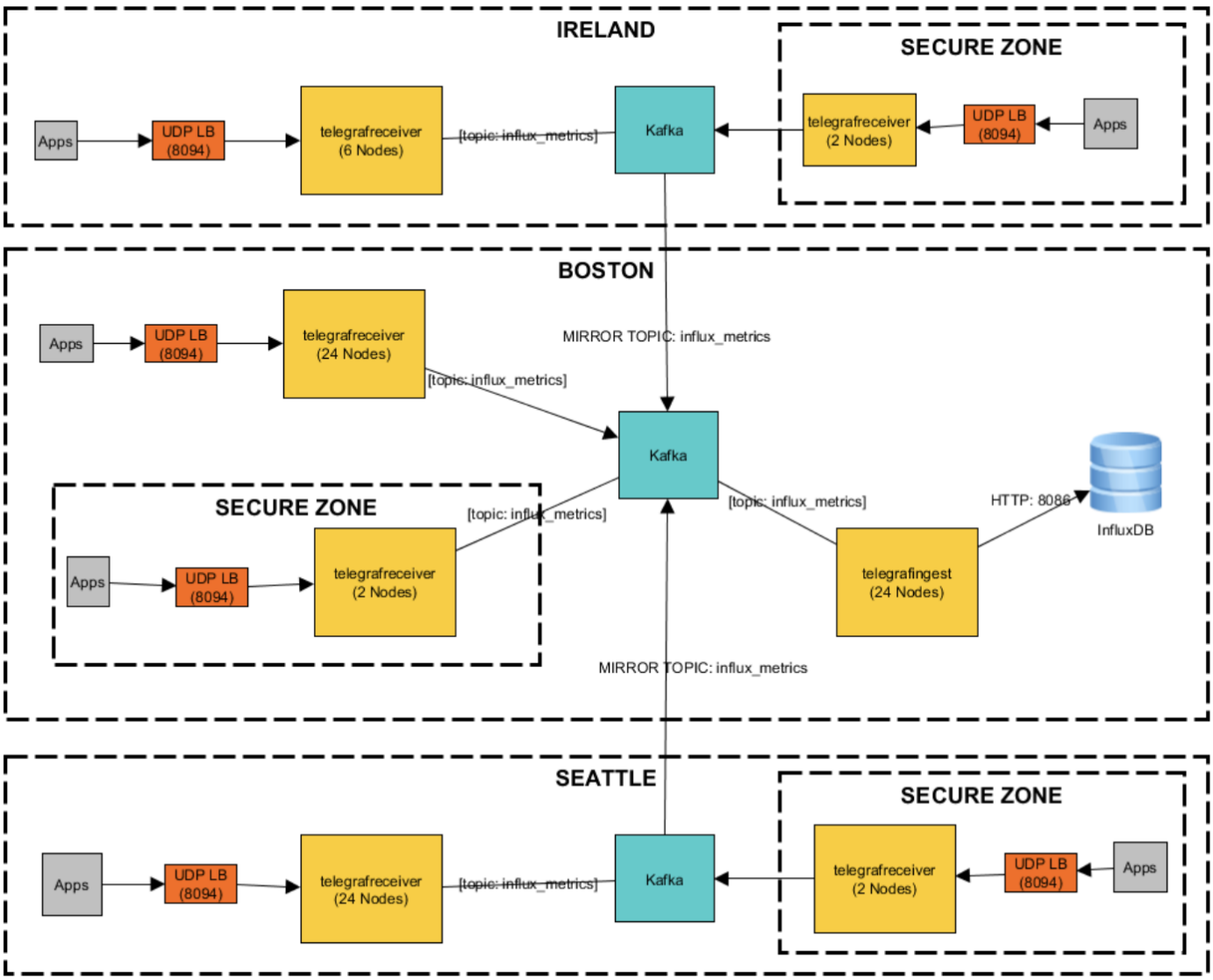
Grafana Graph Showing a Data Comparison in Graphite and InfluxDB

Technical architecture

> *We recently introduced InfluxDB as our first-class time series database system, where we had the opportunity to work directly with InfluxData to ensure we were on a path that is scalable, robust, and in line with the future direction of their platform.”*

Jim Hagan, Manager

Reference Architecture for Wayfair's InfluxDB Metrics Data Pipeline



Wayfair use this Kafka-centric pipeline to integrate metrics data from three different data centers (including secure and non-secure zones in each).

A modern metrics data pipeline

Wayfair are using their migration to InfluxDB as an opportunity to build a more flexible and robust data architecture with Kafka as an intermediate metrics buffer. This is modeled after a paradigm they've used successfully with their logging system.

InfluxData's Telegraf service made it relatively easy to configure a multi-layered pipeline by which applications could send data to Telegraf and allow Telegraf to pipe it into Kafka for later consumption. In practice they are using Telegraf twice:

- One layer to receive raw InfluxDB line protocol from applications via UDP and forward it on to Kafka
- An additional layer to consume metrics from the Kafka buffer and write to InfluxDB

Wayfair's buffered model:

1. Allows them to connect multiple data centers by mirroring Kafka topics to shuttle metrics, rather than through cross-data center database replication
2. Allows them to use fast, non-blocking data protocols such as UDP upstream (at the top of the data funnel) and more transactionally robust protocols such as TCP as they get downstream
3. Gives them the ability to inject various processing hooks into the data stream as their business needs evolve.
4. Makes it easy to write the same data to multiple instances of InfluxDB (simply by consuming the same topic as the main cluster)
5. Gives them multi-day tolerance against a severe network connectivity incident (dependent on the configured age limit of messages in Kafka)

InfluxDB is a major component in Wayfair's Cyber 5 Holiday weekend monitoring and alerting systems for monitoring data center metrics as well as e-commerce site user metrics:

- Data center metrics (100s of apps sending metrics from multiple data centers):
- At the three data centers, Wayfair use Kafka's MirrorMaker to replicate the data to all three locations
- They have three six-node InfluxDB Enterprise clusters, which are dedicated to different workloads: storefront metrics, general monitoring of Kafka queues, containers, etc., and all other application monitoring.

Real user monitoring - RUM (client-side monitoring):

- Understand user experience - deploy 100s of code changes to the app, each change has the potential to impact performance for better or worse
- Daily, 20 million RUM measurements across eight stores, hundreds of page types, & thousands of device types (phones, tablets, laptops & PCs)

What's next for Wayfair?

Wayfair is in the process of fine-tuning their data pipeline architecture to take advantage of Kafka in more sophisticated ways. For example, they are working on dynamically routing data into separate Kafka topics based on tags present in the incoming data. This will enable them to route data into separate databases within a single instance, or even separate InfluxDB clusters. This decoupling of the movement of data from its production and consumption forms the basis for their high availability and disaster recovery strategies. It will also help them break the cycle of monolithic unscalable systems early on.

Wayfair is working strategically with InfluxData to ensure they are on a path that is scalable, robust, and in line with the future direction of their platform. Wayfair also think their massive e-commerce platform provides a great set of rich case studies to help drive InfluxDB further in its support of features that larger enterprises need. The issues which they'll continue to hammer concern things such as monitoring, high availability, support for even greater cardinality, and more elegant solutions for multi-tenant instances.

Technical architecture

> *Our next-generation pipeline takes advantage of Kafka and the Telegraf streaming service to create a more robust data topology. Essentially this allows us to explicitly implement the four R's: routability, retention, resilience, and redundancy."*

Jim Hagan, Manager

Wayfair's time series infrastructure using InfluxData is delivering visibility into their infrastructure as well as client-side monitoring in an efficient and scalable manner.

The four R's of metrics delivery

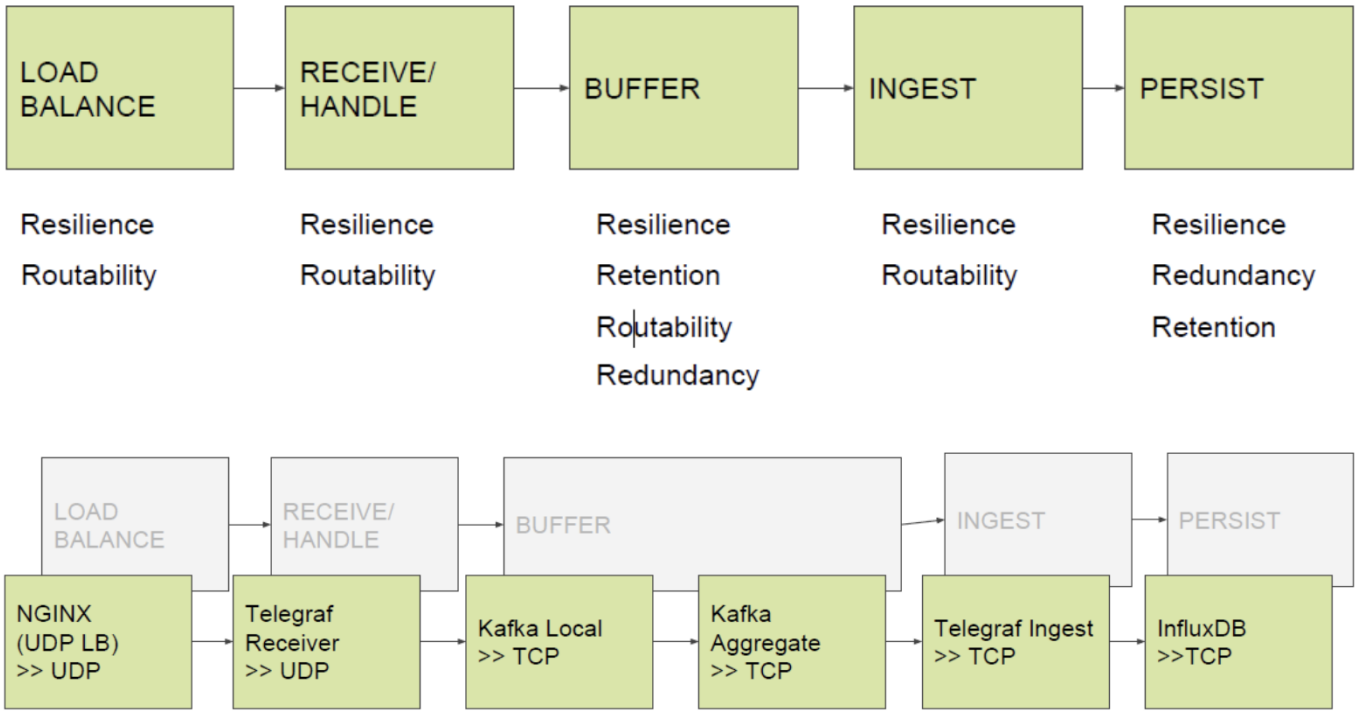
Wayfair's data pipeline built so far for metric collection, delivery, and use utilizes both Graphite and InfluxDB as a time series platform and sends a diverse set of event trackers, timers, and other system metrics from over 2,000 VMs running hundreds of applications.

Wayfair's legacy data pipeline was an elaborate series of services relaying data from data center to data center over UDP (used to avoid blocking calls from the client). This configuration works very fast (and even supported replication) but lacks a number of the elements they are looking for in our metrics pipeline.

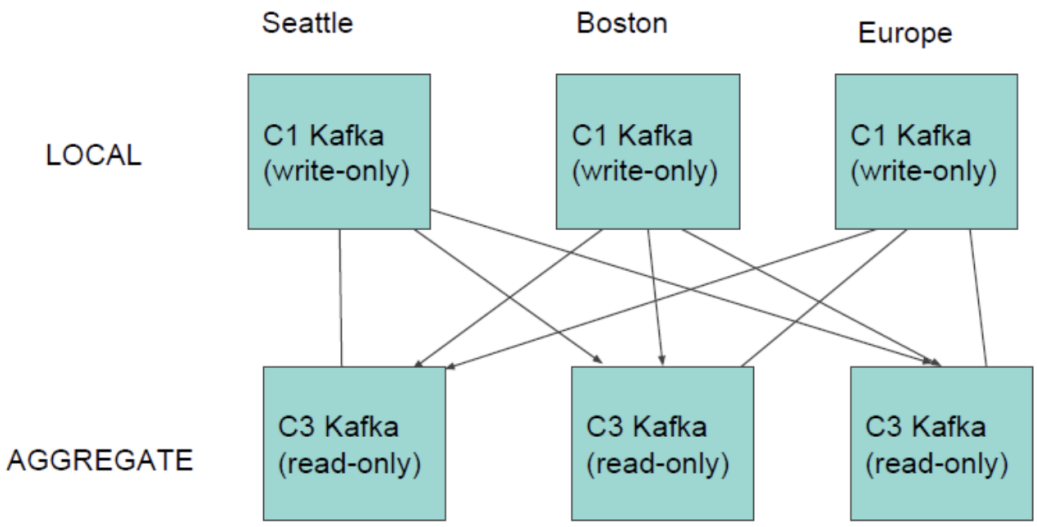
Wayfair's next-generation pipeline using Kafka and Telegraf has allowed them to create a more robust data topology, and effectively, to implement the four R's of metrics delivery:

- **Routability** (keep, drop, redirect)
- **Retention** (keep data in pre-digested, or final format for some time)
- **Resilience** (survive network or DC failure, recover data after a DB failure, survive massive flood of data)
- **Redundancy** (replication of raw data for failover and purpose built dbs)

Conceptual Architecture to Support the Four R's

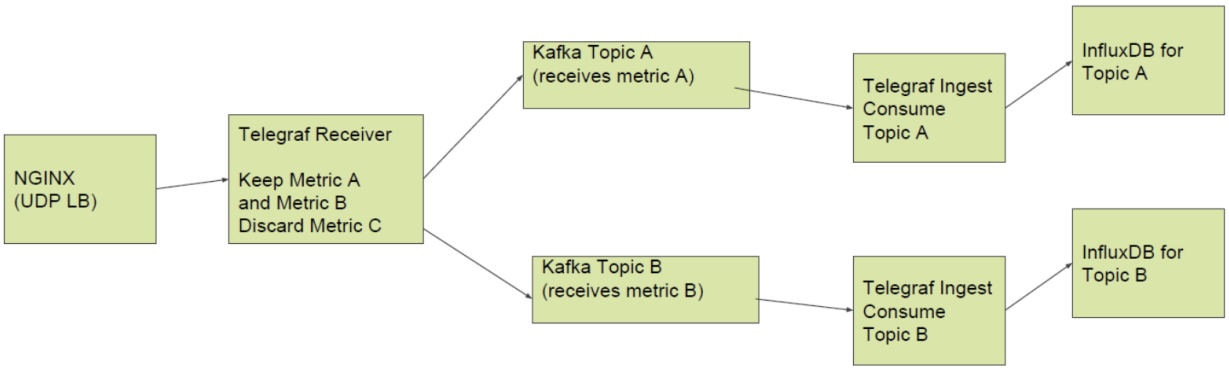


1. **Wayfair use the UDP load balancing plugin for Nginx**. This allows them to take very high data rates including 3 to 5 x spikes and efficiently route them to an array of telegraf hosts; 3 UDP load balancers can feed into dozens of telegraf hosts. In addition, Wayfair can use different port designations to route traffic. This is giving them RESILIENCY and a means of top-level ROUTABILITY.
2. **Wayfair use several features of Telegraf to perform basic traffic shaping**. They use tag and measurement filters to drop, keep, or route certain metrics to a specific Kafka topic, and route to specific Kafka brokers. They get both RESILIENCE and ROUTING in this layer.
3. **The local Kafka layer gives Wayfair an immediate place to store metrics coming into the system.** No expensive processing needs to be applied to the data yet. They configure several different "mirroring" services to copy different topics from local Kafka instances to what they call "Aggregate" Kafka instances. All of this communication happens with minimal transformation. Wayfair get RESILIENCE, ROUTING and RETENTION in this layer.



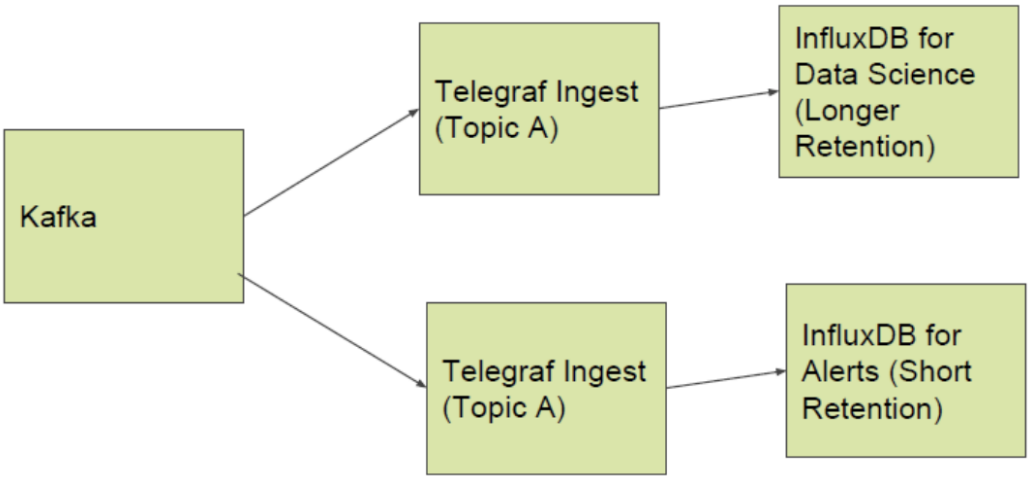
The C1 Clusters are used for write-only in the local data centers. In addition, each data center has an “aggregate” cluster (C3) for consumption of the integrated data stream.

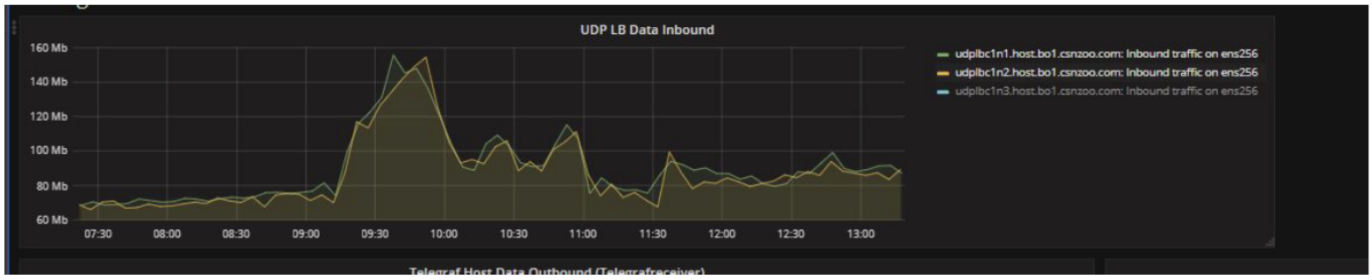
4. **Wayfair have a second layer of Telegraf acting as a Kafka consumer.** This allows them to subscribe to topics that they care about and route them to the DB of their choice. They can also deploy multiple layers of ingest and populate multiple databases. They are getting both RESILIENCE and ROUTING and REDUNDANCY in this layer.



Some Architectural Recipes (Dynamic Topic Routing)

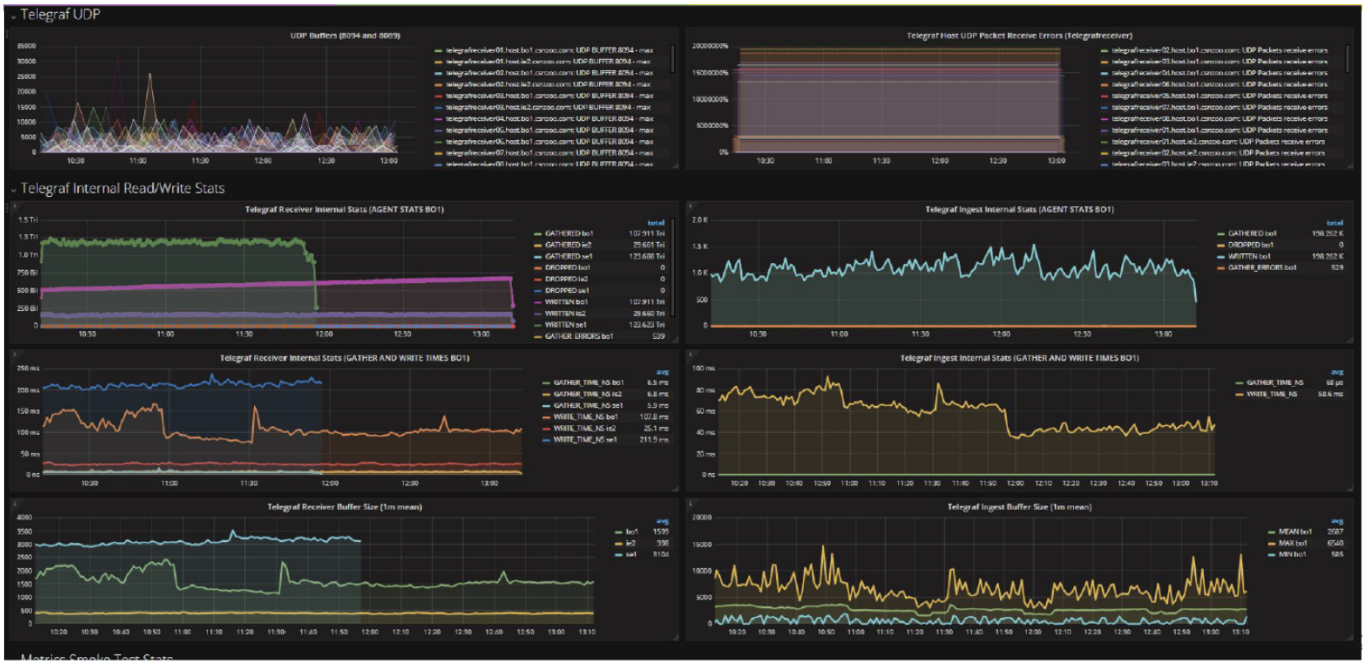
Some Architectural Recipes (Redundancy)





Monitoring it All (Load Balancer Layer)

Monitoring it All (Telegraf Layers)



Monitoring it All (Telegraf Layers)

Real user monitoring to track actual performance

InfluxDB helps deliver accurate performance monitoring, which is crucial for Wayfair's Storefront Engineering team. Each day they deploy hundreds of code changes to the web application for their customer-facing websites, and each change has the potential to impact performance for better or worse. For this reason, they carefully monitor KPIs such as page load times to catch regressions, identify opportunities for speedups, and verify that improvements work in the real world.

As part of their migration from their legacy Graphite backend to next-generation InfluxDB, Wayfair were able to give their client-side monitoring of page load times in the browser, known as real user monitoring (RUM), a major upgrade.

RUM systems use JavaScript and standard browser APIs like PerformanceTiming to record load times experienced by real customers. By tracking performance in real world conditions, Wayfair can identify issues that would otherwise be invisible. This is increasingly important as pages become more complex, including resources, images, fonts, and scripts from a range of third-party servers and networks.

Wayfair's RUM system was first created using a Graphite time series database for real-time charts and alerts. Over the years, their RUM dashboards have been incredibly useful at identifying issues, but there are quirks that make it difficult to interpret the data. RUM is particularly challenging to measure and visualize due to the wide range of values and outliers. For example, a page with a median load time of 5 seconds will commonly see some page loads of 200+ seconds. Just a few outliers can drastically skew the mean, and the legacy Graphite backend was particularly susceptible to this problem.

Graphite estimates and samples in a few places during data ingestion. In some common cases those errors compound, resulting in misleading graphs that waste time. The new InfluxDB back-end doesn't suffer from that problem, since it uses a true median calculation, one of many available operators, aggregators and selectors.

InfluxDB schema for RUM

Schemas are the biggest change for application developers who are familiar with Graphite. InfluxDB supports defined fields and indexed tags for each measurement, while Graphite uses a plain, unstructured dot-delimited string to identify each metric.

An example templated Graphite metric:
`rum.client_timers.$device_type.$store.$page.$metric.$dc.timer.mean`

An example query:
`rum.client_timers.desktop.wayfair_com.index.speed_index.bo1.timer.mean`

The simple structure used by Graphite makes it easy to get started, but complexity grows quickly as you add facets. For example, in order to break out by browser, the same measure was stored multiple times, but it was limited to only certain pages to keep disk usage reasonable on Graphite hosts. Dashboards have to be created with these limitations in mind, and new views would often require deploying new metrics to duplicate measures.

An example templated Graphite metric with browser:

```
rum.client_timers.$device_type.$store.$page.$browser.$metric.$dc.timer.mean
```

In contrast, designing InfluxDB schema takes a little more time upfront, but it gives you more power and flexibility. Dashboards are generic and easy to customize. Introspection queries let you explore data in a natural way to find interesting cross-cuts and correlations.

Similar to a table in a SQL database, the schema has a field for each value that Wayfair measure on a page load:

connection_time	load_event_time
dns_lookup_time	receiving_time
dom_content_loaded_time	redirect_time
dom_interactive_time	secure_connection_time
dom_processing_time	speed_index
first_meaningful_paint_time	total_page_load_time
first_paint_time	waiting_for_response_time

Tags (similar to indexed columns in SQL) are used to filter and slice:

browser	is_login_recognized
client_type	os
customer_shard	script_name
dc	store

Optimizing the schema

When Wayfair initially deployed the InfluxDB schema and started building dashboards, they ran into a problem: slow queries and timeouts. This made it difficult to explore the data and impossible to view a time range of more than a couple of hours. Due to the daily patterns, the inability to chart the last 24 hours was a dealbreaker, so they had to find a solution. Engineers from InfluxData visited the Wayfair offices in Boston to put on a workshop for their application developers. This helped Wayfair's team quickly get up to speed on the InfluxDB data model and best practices. They learned that query performance scales with series cardinality, and were able to diagnose the issue with their RUM schema.

They had inadvertently added a host tag to the schema which recorded the web server that received the RUM metrics from the browser. There's no value to filter on that tag, and removing it reduced cardinality by 600x. After resetting the schema by moving the RUM measurement to a new cluster, query performance was dramatically improved.

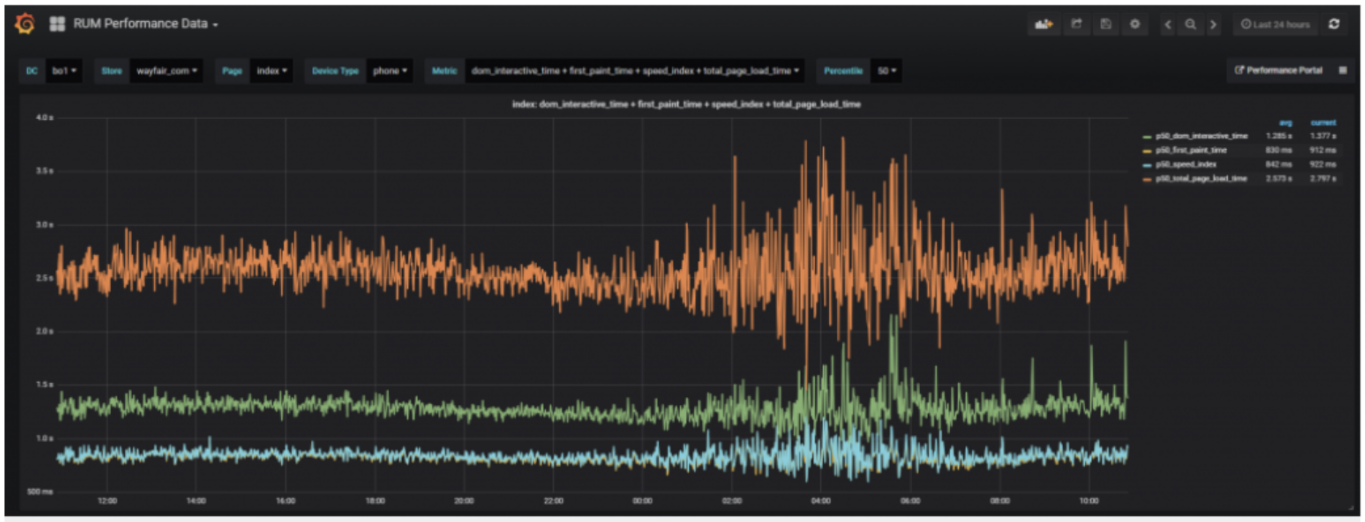
Dashboards

Wayfair use Grafana to build dashboards and visualize data from a variety of sources. Grafana's extensive InfluxDB support helped make this a smooth transition.

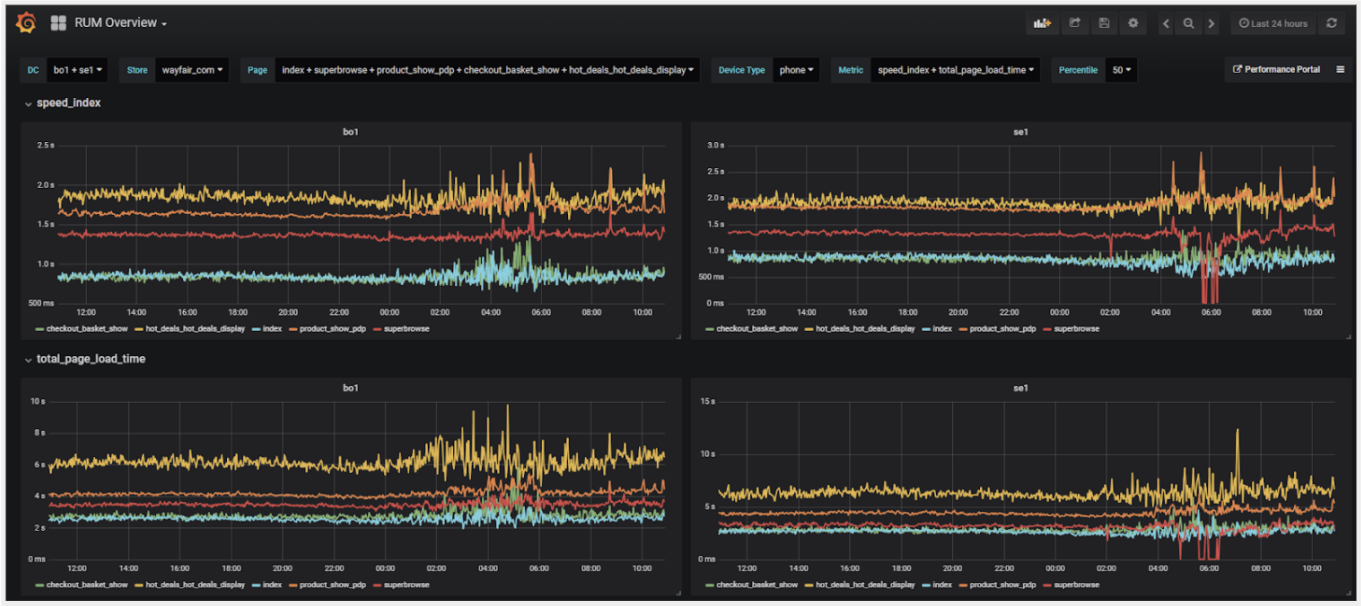
With Graphite RUM, it was confusing and unintuitive to interpret each graph without a certain amount of background expertise. InfluxDB has made these graphs more accessible to a broad audience of engineers and product managers across the Wayfair organization. On the same token, some filters in Graphite (for example, break out by browser) were limited to certain pages. With InfluxDB, Wayfair have visibility of full tags on every page by default.

Dashboards are used to improve customer experience. The Login Status dashboard, for example, displays when a page is slower for logged in customers, enabling Wayfair to investigate the cause and fix the bottleneck.

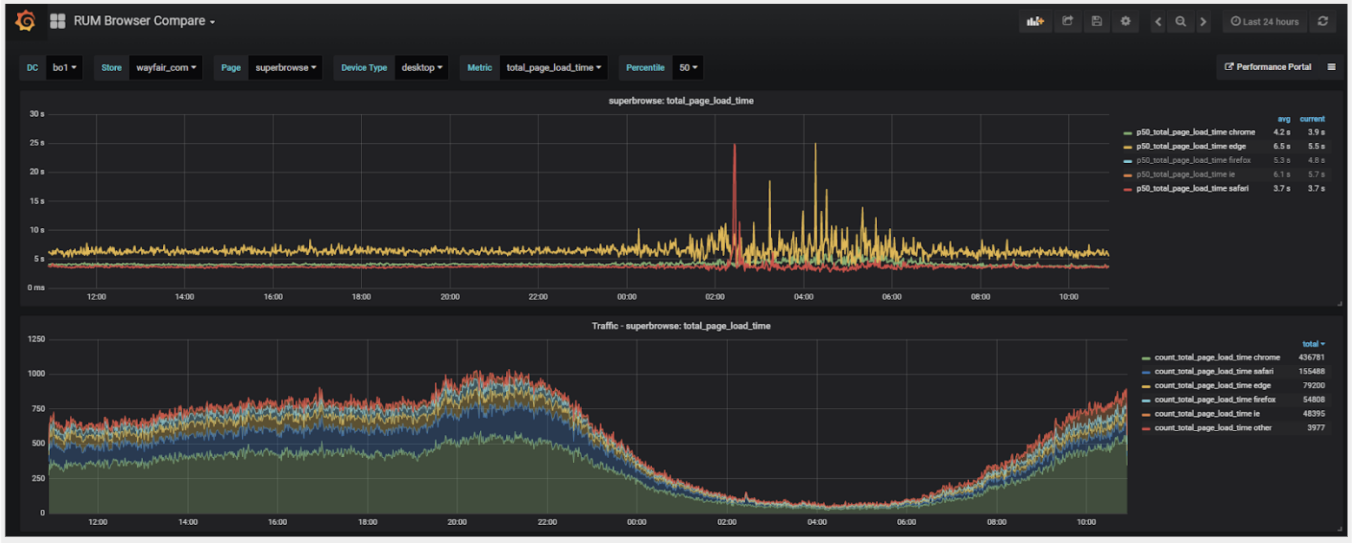
Generic RUM Dashboard to View a Single Page and Store



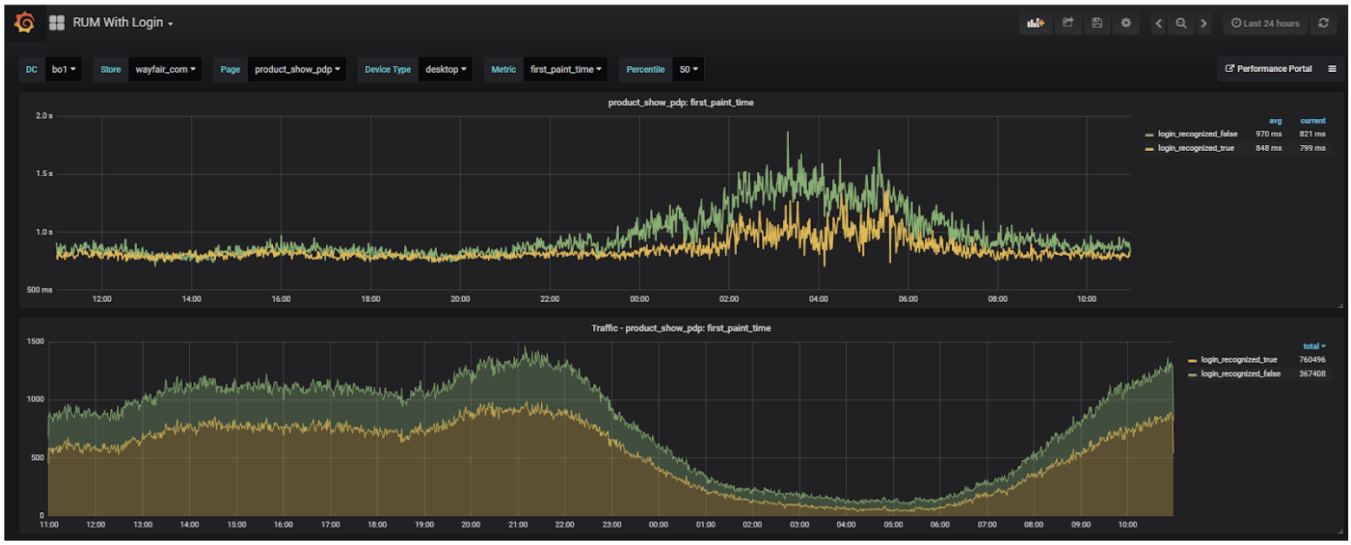
RUM Overview Dashboard
(To View a Configurable Selection of Important Pages)



RUM Browser Compare Dashboard

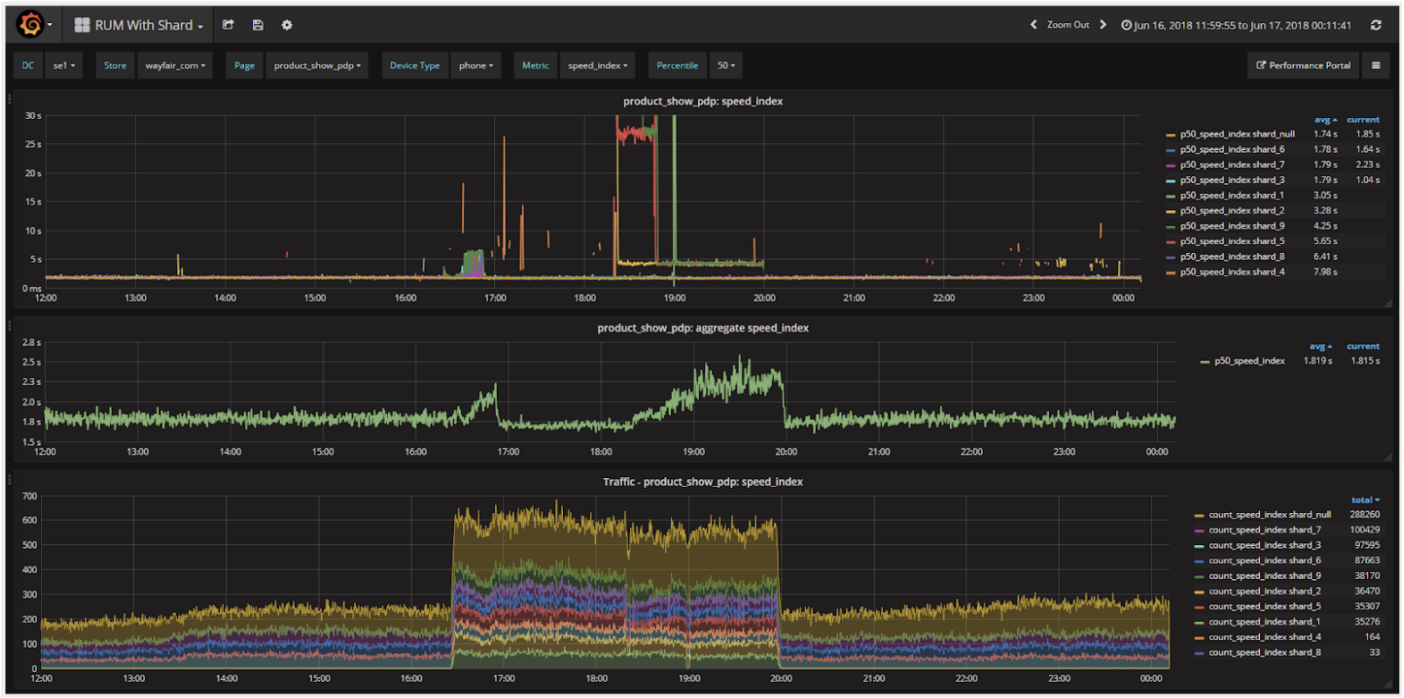


RUM with Login Status Dashboard



RUM with Store Dashboard
(To Compare a Single Page Across all Stores and Data Centers)

RUM with Customer Shard
(Load Times During Database Maintenance Work in Which Traffic for Certain Groups of Customers Was Shifted Between Data Centers)



With InfluxDB, Wayfair is able to represent the full granularity of some of these metrics due to the different storage architecture used; with Graphite they ran up against the issue of pre-allocated storage (which consumes space in direct proportion to the number of series). They had to lower their original metrics granularity or they would have exhausted all storage in their Graphite cluster. They are also able to hone in on specific areas of consideration using the where clause.

In short, the following InfluxDB strengths provided the functionality that Wayfair needed both to monitor performance across data centers and to perform RUM in the context of rapid growth:

- Central scalable repository for general, critical and business KPIs
- Resilience and HA
- Cloud-friendly
- Configurable retention policies and sharding options
- Support for granular raw data to capture even rare events
- Support for RUM wide range of data, values and outliers with real-time dashboarding and alerting

By aligning their time series data infrastructure with their growth requirements and powering their architecture with the InfluxData modern time series platform, Wayfair have gained system-wide visibility spanning backend and storefront performance in support of fulfilling their mission and maintaining their promise to partners, suppliers and customers.

About InfluxData

InfluxData is the creator of InfluxDB, the leading time series platform. We empower developers and organizations, such as Cisco, IBM, Lego, Siemens, and Tesla, to build transformative IoT, analytics and monitoring applications. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by sensors, applications and computer infrastructure. Easy to start and scale, InfluxDB gives developers time to focus on the features and functionalities that give their apps a competitive edge. InfluxData is headquartered in San Francisco, with a workforce distributed throughout the U.S. and across Europe. For more information, visit influxdata.com and follow us @InfluxDB.

influxdata®

Try InfluxDB

Get InfluxDB

Contact us for a personalized demo influxdata.com/get-influxdb/