Open-Falcon
A Distributed and High-Performance Monitoring System

Yao-Wei Ou & Lai Wei
2017/05/22
Let us begin with a little story...
Grafana PR#3787

[feature] Add Open-Falcon datasource
“I'm sorry but we will not merge any new datasources unless they are very popular.”

- Grafana
[feature] Add Open-Falcon datasource #3787

Closed

hibripad wants to merge 4 commits into grafnan:main from hibripad:feature-openFalcon

Open-Falcon is a monitoring system initiated by Xiaomi.

CLassistant commented 7 days ago

You should sign our Contributor License Agreement in order to get your pull request merged.

bergquis commented 7 days ago

I'm sorry but we will not merge any new datasources unless they are very popular. I suggest you create a plugin instead. Please take a look at https://github.com/grafana/datasource-plugins-kairosdb as an example.

I would suggest you open an issue next time before doing all this work so we could have told you earlier :(
chuanal commented 5 days ago

Open-falons is very popular monitoring system. Please reconsider this pull-request.

berqrsa commented 5 days ago

@htrlpod you can create plugins under your own name. But I suggest you wait a few weeks since we are currently working a lot on the plugin systems and things might change. After that you can expect some documentation about how to do it.

Catchet commented 5 days ago

+1

thuhak commented 5 days ago

+1

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

ProTip! Add *patch* or *diff* to the end of URLs for Git’s plain-text views.
Open-Falcon, now is one of the most popular monitoring systems in China.
About us

Laiwei

- Technical director of DiDi, China.
- Founder of Open-Falcon software.
- Core maintainer of Open-Falcon community.
- Focus on service reliability, DevOps, Cloud computing, etc.

Yaowei

- Director of WiFire oversea R&D center.
- Core maintainer of Open-Falcon organization.
- Leads the development of CDN monitoring system in Fastweb.
- Focus on CDN and Blockchain.
Outline

- Motivation
- Features
- Architecture
- Comparison
- Community
There are already so many outstanding open source monitoring systems. Why do we reinvent a wheel? And how does it become the most popular monitoring system in China?
Motivation(2/3)

Zabbix

- difficult to scale out (~2000)
  - Database can grow very large, very quickly if not tuned properly.
  - not very resource friendly: a lot of connections will be made.
  - we have to build several zabbix cluster to deal with the rapid business grow.
- a bit of a learning curve
  - Tuning and tweaking can be a lot of work
  - more complex to setup
  - hard to extend
  - manually configuration

2015 open sourced by Xiaomi SRE, China, under Apache License Version 2.0

- to replace Zabbix
key factors of an enterprise class monitoring system.
Features
Scalability

- Scalable monitoring system is necessary to support rapid business growth. Each module of Open-Falcon is super easy to scale horizontally.

- Supports up to hundreds of million transactions per minute (query/judge/store/search).

- Can easily support over 100,000 hosts.
Performance

- With RRA (Round Robin Archive) mechanism, the one-year history data of 100+ metrics could be returned in just few seconds.
- Stores 10+ years historical metrics.
High Availability

- No critical single point of failure
- Easy to operate and deploy
Flexibility

- Falcon-agent has already 400+ built-in server metrics. Users can collect their customized metrics by writing plugins or just simply run a script/program to relay metrics to falcon-agent.
- Extensive architecture.
- Customizable metrics.
- Abundant APIs.
For easier management of alerting rules, Open-Falcon supports strategy, expression, template inheritance, and multiple alerting method, and callback for recovery.

- Auto discovery of endpoints and counters.
- API support.
User-Oriented

- Supports Grafana Datasource.

- Open-Falcon could present multi-dimension graph, including user-defined dashboard/screen.
Grafana with Open-Falcon Screenshot (1/3)
Grafana with Open-Falcon Screenshot (2/3)
Architecture
Components

COLLECT → STORE → PRESENT

JUDGE → NOTIFY
Components

COLLECT -> STORE -> JUDGE

PRESENT -> NOTIFY
Components

1. COLLECT
2. STORE
3. PRESENT
4. JUDGE
5. NOTIFY
Components

COLLECT
- Falcon-Agent
- Proxy-Gateway
- aggregator
- nodata

STORE
- graph
- redis
- MySQL
- hbs

PRESENT
- Falcon-Dashboard
- Grafana
- Falcon-API

JUDGE
- judge

NOTIFY
- alarm
Before: Too many modules

![Diagram showing various modules and their connections.](image)

- **Agent1**, **Agent2**, **Agent3**, **Agent4**, ..., **AgentN**
- **UIC Dashboard Portal FE**
- **Alarm Query Graph Judge Transfer**
- **Transfer** has an input of 60 and an output of 5.

Module connections and data flow are indicated by arrows.
After
After

Alarm
Dashboard
Portal
FE
UIC
After

Dashboard

Alarm

FE

Portal

UIC

Dashboard

Alarm

API Gateway

HBS
(Control)

Judge

Graph

Transfer (Queue)
After

Central Status

Alarm

Dashboard

Judge

FE

FE

UIC

Portal

Transfer (Queue)

HBS (Control)

Dashboard

Falcon-Plus

Agent1

Agent2

Agent3

Agent4

⋯

AgentN
Design Philosophy
Design Philosophy

MICROSERVICES
10 major modules
individual deployed
Design Philosophy

MICROSERVICES
10 major modules
individual deployed

PUSH
agent as a push proxy
Design Philosophy

**MICROSERVICES**
10 major modules
individual deployed

**PUSH**
agent as a push proxy

**RRDTOOL**
with consistent hash
Design Philosophy

MICROSERVICES
10 major modules
individual deployed

PUSH
agent as a push proxy

RRDTOOL
with consistent hash

BINARY
Go static binary
Design Philosophy

MICROSERVICES
10 major modules
individual deployed

PUSH
agent as a push proxy

RRDTOOL
with consistent hash

BINARY
Go static binary

DEPLOYMENT
mass deployment by ops-updater
Design Philosophy

**MICROSERVICES**
10 major modules individual deployed

**BINARY**
Go static binary

**PUSH**
agent as a push proxy

**RRDTOOL**
with consistent hash

**DEPLOYMENT**
mass deployment by ops-updater

**GRAFANA**
open-falcon datasource
Comparison
## Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers grouping, deduplication and silencing functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
## Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers <strong>grouping, deduplication and silencing</strong> functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
## Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers grouping, deduplication and silencing functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
## Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers grouping, deduplication and silencing functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
# Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers <em>grouping, deduplication, and silencing</em> functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
# Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers grouping, deduplication and silencing functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
## Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td>Easy to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>simple alert management of own dashboard</td>
<td>Alertmanager offers <strong>grouping, deduplication and silencing</strong> functionality</td>
</tr>
<tr>
<td>Faster query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td>Simple shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
# Compared to Prometheus

<table>
<thead>
<tr>
<th>OPEN-FALCON</th>
<th>PROMETHEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abundant</strong> APIs</td>
<td>Metrics API</td>
</tr>
<tr>
<td><strong>Push</strong> Model: Auto Discovery</td>
<td>Pull Model: Manual configuration</td>
</tr>
<tr>
<td><strong>Easy</strong> to scale out</td>
<td>Harder to scale out</td>
</tr>
<tr>
<td>Simple alert management of own dashboard</td>
<td>Alertmanager offers <em>grouping, deduplication and silencing</em> functionality</td>
</tr>
<tr>
<td><strong>Faster</strong> query performance of RRA</td>
<td>Slower, Recording rules</td>
</tr>
<tr>
<td><strong>Simple</strong> shellscript as plugin</td>
<td>A bit learning curve to write exporter and collector</td>
</tr>
<tr>
<td>Limited expression</td>
<td><strong>PromQL</strong></td>
</tr>
</tbody>
</table>
Community
Ecosystem (1/2)

- Banking
- IaaS, SaaS
- CDN
- O2O
- Social
- Entertainment
- ...
Ecosystem (2/2)

OS
- Windows
- Apple
- Linux

UI
- Grafana
- Falcon

Plugins
- Switch
- Hadoop
- HBase
- Docker
- Redis
- MongoDB
- GPU
- RabbitMQ
- HAProxy
- Nginx
- JMX
- LVS
- Tomcat
- WebSphere
- IIS
Join us

- Github: https://github.com/open-falcon
- Homepage: http://open-falcon.org
- Contact us: openfalcon-users@googlegroups.com
- Wechat: 

![QR Code](image-url)
Summary
100+ companies
40,000+
servers
400+
built-in metrics
5,000+ users
3 seconds
https://github.com/open-falcon/open-falcon