35 YEARS OF PERFORMANCE MANAGEMENT

Visualizing z/VM Performance Data with Modern Tools

Jonathan Castro
GS Forge 26

Software Engineer
Jonathan@velocitysoftware.com

VELOCITY SOFTWARE

Copyright 2026 Velocity Software, Inc. All Rights Reserved.
Other products and company names mentioned herein may be trademarks of their respective owners.

PROVEN PERFORMANCE

Agenda

01

Performance Management

Performance management is essential for IBM Z operating systems.

02

zVPS Visualization

What we collect today, and how customers view it.

03

Customer Challenges

There is too much data, and we need an easier way to view it.

04

VSIPUMP

Getting z/VM metrics off the platform.

05

Grafana Dashboards

Dashboards for z/VM, Linux, z/OS, VSEn, and more.

06

OpenTelemetry

Send data to modern tools like Dynatrace, Datadog, and others.

Performance Management

Performance Management is essential for IBM Z operating systems

Real-time Data

See what the system is doing now.

History Data

See what the system did last month.

Strict Data Validation

Metrics you can validate.

Minimal Overhead

Data collector with a target of less than 1%.

Multiple Data Sources

CP monitor, SNMP, DMF, and SMF.

Single Pane of Glass

One place for z/VM, Linux, VSEn, and z/OS together.

All components of zVPS



VELOCITY SOFTWARE

PROVEN PERFORMANCE

Visualization with zVPS

The traditional display

	<---Users---->			Transact.			<Processor>		Cap-	<--Storage (MB)->			Pages	<------I/O----->			<MiniDisk>		Spool	Communications		SMT
	<-avg number->			per	Avg.		Utilization		ture	Fixed	Active	Stor	/sec	<--DASD-->		Other	<-Cache-->		Page	<-per second->		Prort
Time	On	Actv	In Q	Sec.	Time	CPUs	Total	Virt.	Ratio	User	Resid.	Load	DASD	Rate	Resp	Rate	Rate	%Hit	Rate	IUCV	VMCF	Ratio
10:42:00	66	43	5.0	0.7	0.13	4	2.8	2.4	100.0	372	610	0.3	0	4.8	0.3	1.7	14.9	98.1	0	503.4	0.2	0.521
10:41:00	66	42	6.0	0.7	0.22	4	3.4	2.9	99.97	372	603	0.3	7.3	17.7	0.2	5.0	23.9	98.0	4	531.4	1.2	0.519
10:40:00	66	42	7.0	0.4	0.12	4	3.3	2.8	100.0	372	604	0.3	4.2	14.4	0.2	25.0	22.5	96.8	2	503.3	0.2	0.517
10:39:00	66	43	5.0	0.4	0.08	4	3.2	2.7	100.0	372	614	0.3	0.5	11.4	0.3	11.0	20.5	95.2	0	526.7	1.2	0.517
10:38:00	66	38	3.0	0.3	0.32	4	2.8	2.4	100.0	372	507	0.3	0	4.6	0.3	1.1	13.6	98.7	0	461.2	0.2	0.518
10:37:00	66	41	4.0	0.3	0.10	4	3.1	2.6	100.0	372	527	0.3	0	4.9	0.3	1.4	14.7	98.9	0	448.6	0.2	0.514
10:36:00	66	40	4.0	0.4	1.24	4	3.4	2.9	100.0	372	518	0.3	7.0	16.1	0.2	3.4	19.8	98.6	4	519.1	0.2	0.515
10:35:00	66	40	7.0	0.3	0.08	4	3.3	2.8	99.99	372	519	0.3	4.5	12.9	0.2	0.9	17.6	99.0	2	476.5	0.2	0.514

Traditional way to look at the data

- Easy to use interface
- Detailed and easy to read
- Handles z/VM, Linux, VSEn, and z/OS

Who it's built for

The mainframe team

It takes a 3270 session, knowing what data you want and where to find it.

Everyone else

Want an easier interface to work with, one accessible through the web.

VELOCITY SOFTWARE

PROVEN PERFORMANCE

Visualization with zVPS

The web display

Tables or graphs

Each window holds tabular data or a graph

Standard graphs

A standard set of graphs and graph types

Many systems at once

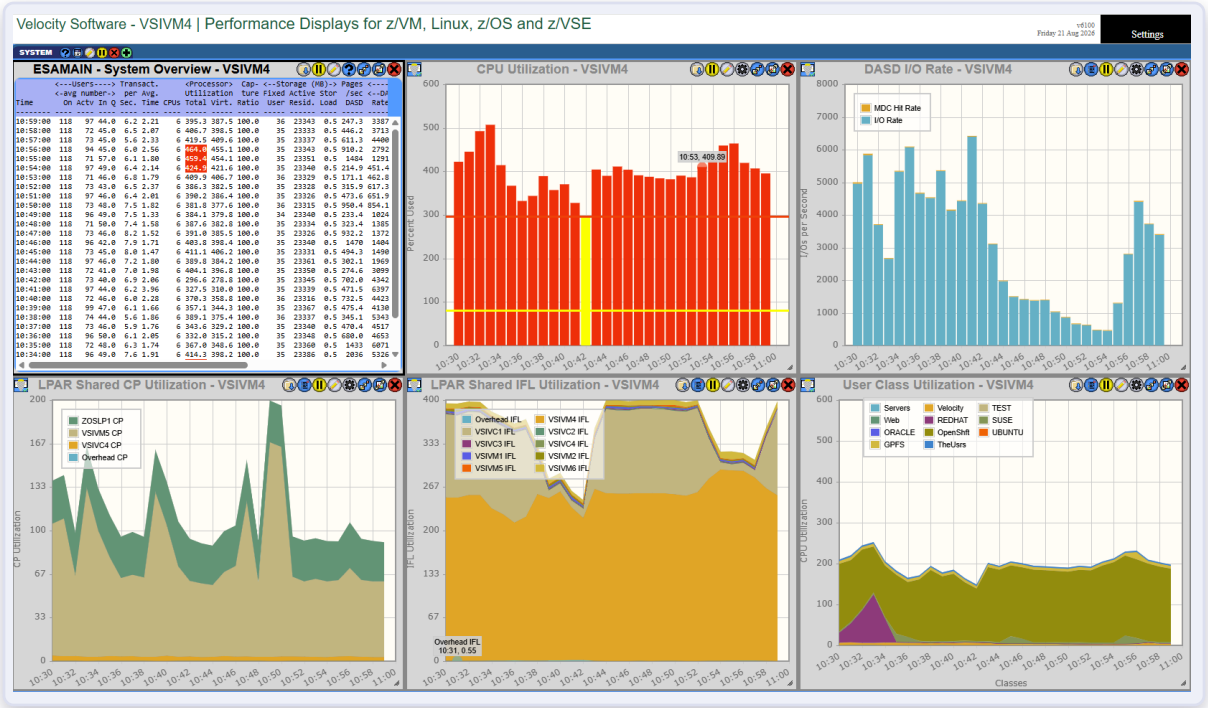
Display several systems on one screen

Mix any data

z/VM, VSEn, Linux, and z/OS side by side

Your own view

Users save their own preferred layout



Customer Challenges

Large volume of data

- The CP monitor and SNMP provide data every minute
- Hundreds of screens and thousands of metrics
- They need the important numbers on one screen

Want modern tools

- Up to date displays
- Better graphics
- Familiar to a wider audience

The three names that have come up

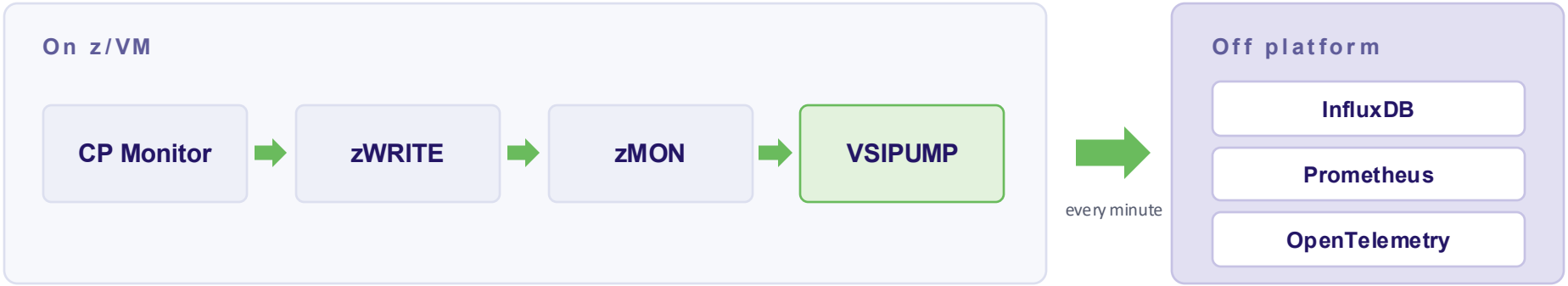
Prometheus

influxdb

OpenTelemetry

Introducing VSIPUMP

Data flow



Deployment

- Delivered as a zVPS product package
- Runs in its own virtual machine
- Collects real time data every minute

Integration

- Connects to InfluxDB or OpenTelemetry endpoints
- Sends data every minute via an HTTP POST request
- Optional collection for Linux, VSEn, CICS, Kubernetes, and MongoDB

Where the Data Lands

Two destinations, two models

InfluxDB

Push · line protocol



VSIPUMP opens the connection. Nothing on z/VM has to listen.

Pick it when

Starting from scratch, and want the shortest path to getting data into Grafana.

Prometheus

Pull · OpenMetrics



Prometheus opens the connection. Your endpoint has to be reachable from it.

Pick it when

Already using Prometheus in your environment, z/VM endpoint becomes one more target to add.

Setting Up InfluxDB

Path 1 · InfluxDB, then Grafana

STEP 1 · IN INFLUXDB

Create a user and a database

1. Create a user
2. Create a database

```
create user vsi
  with password 'velocity'
  with all privileges

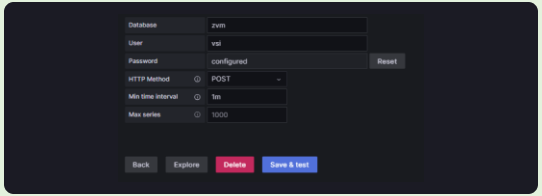
create database zvm
```

VSIPUMP pushes to this database. Grafana logs in with the same user in step 2.

STEP 2 · IN GRAFANA

Add the data source

1. Open Connections
2. Search for InfluxDB
3. Enter database, user, and password
4. Save & test



Setting Up Prometheus

Path 2 · Prometheus, then Grafana

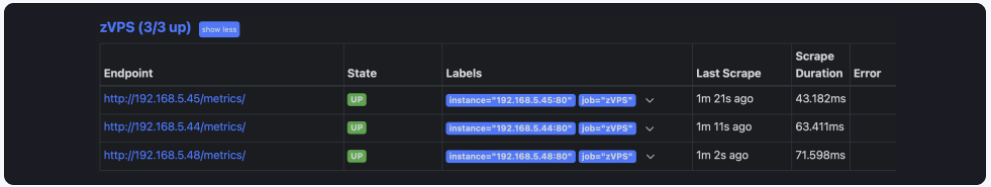
STEP 1 · IN PROMETHEUS

Add a scrape job

1. Edit prometheus.yml
2. Reload Prometheus
3. Check Targets Page

```
# prometheus.yml
scrape_configs:
  - job_name: 'zVPS'
    metrics_path: /metrics/
    static_configs:
      - targets:
['192.168.5.45','192.168.5.44','192.168.5.48']
```

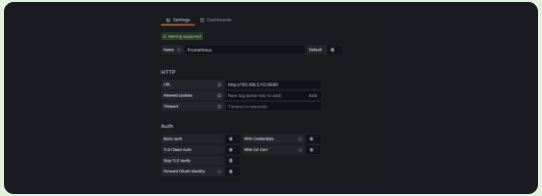
WHEN IT WORKS



STEP 2 · IN GRAFANA

Add the data source

1. Open Connections
2. Search for Prometheus
3. Enter the Prometheus server URL
4. Save & test



Grafana Dashboards

z/VM Overview



Grafana Dashboards

z/VM Overview

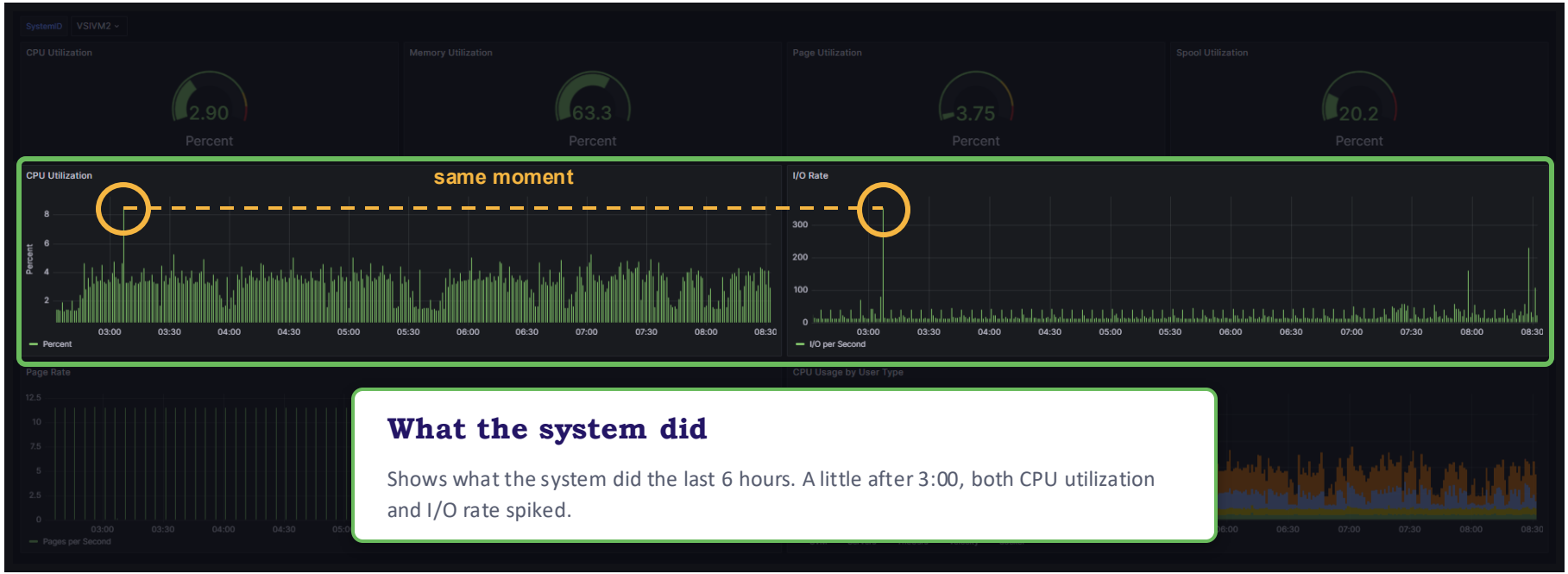


Where the system currently stands

Shows the current CPU, memory, page, and spool utilization for the selected system.

Grafana Dashboards

z/VM Overview

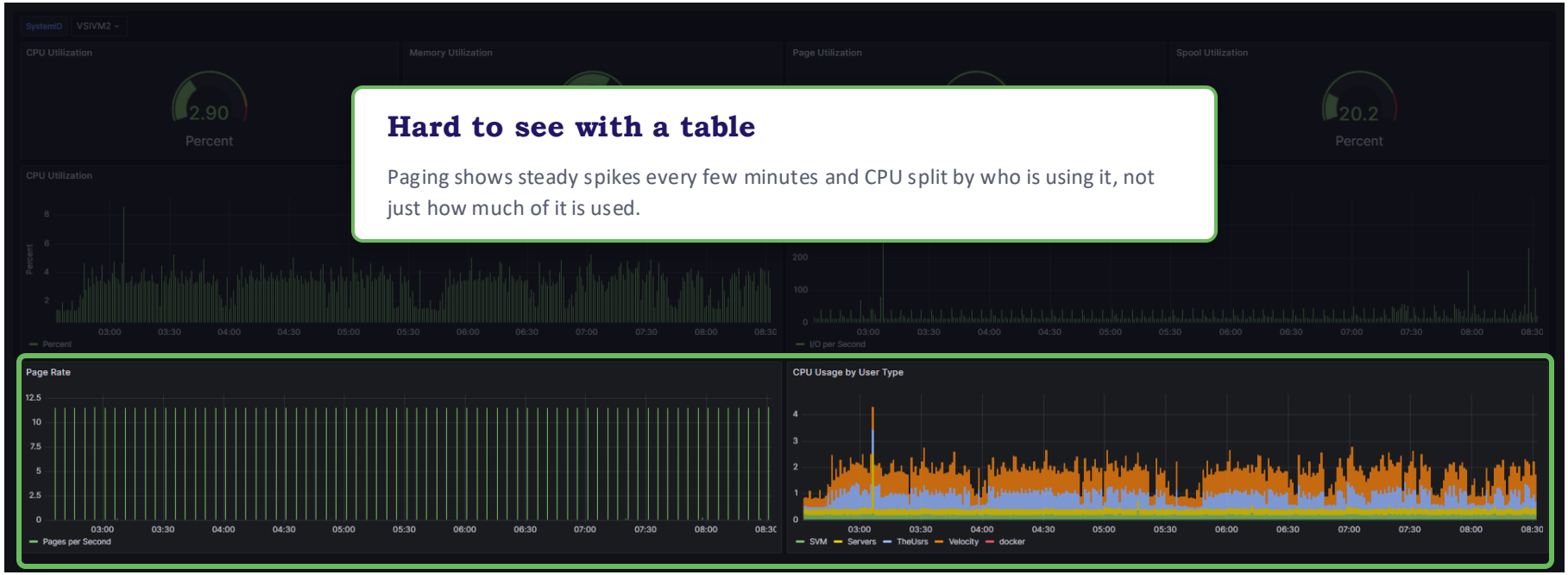


What the system did

Shows what the system did the last 6 hours. A little after 3:00, both CPU utilization and I/O rate spiked.

Grafana Dashboards

z/VM Overview



Hard to see with a table

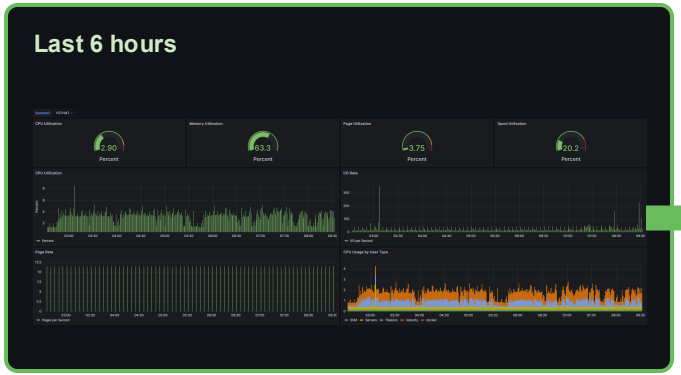
Paging shows steady spikes every few minutes and CPU split by who is using it, not just how much of it is used.

Grafana Dashboards

Not just six hours

TIME RANGE

- Last 5 minutes
- Last 1 hour
- **Last 6 hours**
- Last 24 hours
- **Last 7 days**
- Last 30 days
- Last 1 year





Drag across any chart to zoom into a single moment — the whole dashboard follows.

How far back you can go is not up to Grafana

Grafana stores nothing. The window is set by retention in InfluxDB or Prometheus — pick a range longer than they keep, and the chart simply runs out.

Data Retention

How long the data is kept

Prometheus

Default: 15 days

```
# startup flags
--storage.tsdb.retention.time=90d
--storage.tsdb.retention.size=50GB
```

Restart Prometheus for them to take effect.

InfluxDB

Default: keeps everything

```
-- what's set now
SHOW RETENTION POLICIES ON "zvm"
-- keep 90 days
ALTER RETENTION POLICY "autogen" ON "zvm" DURATION 90d
```

Older data starts getting removed in each retention check (default 30 min).

Dashboards

Between the time range and retention, you get to decide how far back your dashboards can go.

Grafana Dashboards

More Dashboards!

z/VM Overview



z/OS



Linux



VSEn



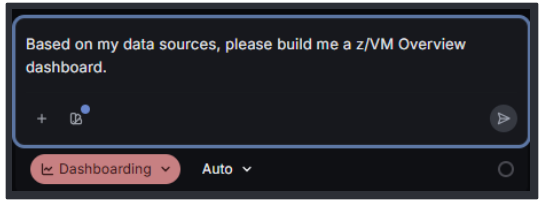
MongoDB



Grafana Assistant

From one sentence to a first draft

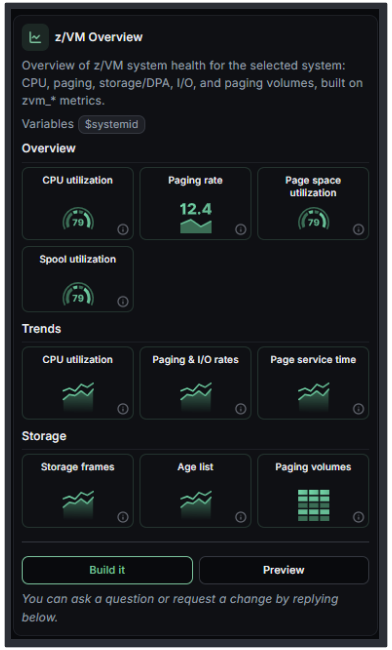
1 · ASK



One sentence, in dashboarding mode.

No metric names and no queries. It looked at our data sources and found the z/VM metrics itself.

2 · REVIEW THE PLAN



3 · BUILD IT



Ten panels in three rows, with a System picker at the top.

Grafana Assistant

It tells you what it guessed



ITS REPLY, AFTER BUILDING THE DASHBOARD

I guessed some settings that you should check:

CPU utilization gauge: it's set to 0–300%, with warning at 240% and critical at 270%, because VSIVC1 has peaked at about 246% since its data started this morning. If CPU is summed across processors, the real ceiling is 100% times the number of processors, so the scale may need to change.

Page space and spool gauges: they're set to 0–100%, with warning at 70% and critical at 90%. VSIVC1's values (about 54% and 18%) show these metrics report percentages, not fractions.

A first draft: check what it guessed, then save it.

Grafana Alerting

From a threshold to a notification

1 THE RULE

A query and a condition

QUERY
CPU utilization

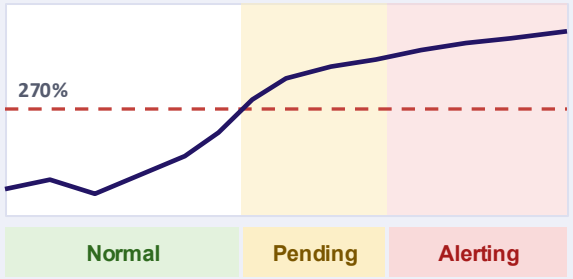
CONDITION
Above 270%

PENDING PERIOD
5 minutes

CHECKED
Every minute

2 THE STATE

Pending first, then Alerting



It only alerts if the value stays above 270% for the whole pending period.

3 THE NOTIFICATION

Sent to a contact point

Email
Slack
Microsoft Teams
PagerDuty
Discord

It sends another message when the value is back to normal.

One rule covers every system. Each system gets its own alert, and it works with both InfluxDB and Prometheus.

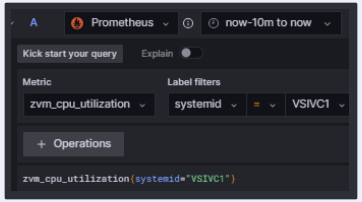
Grafana Alerting

Setting up the CPU alert

Alerting → Alert rules → Create alert rule

1 THE QUERY

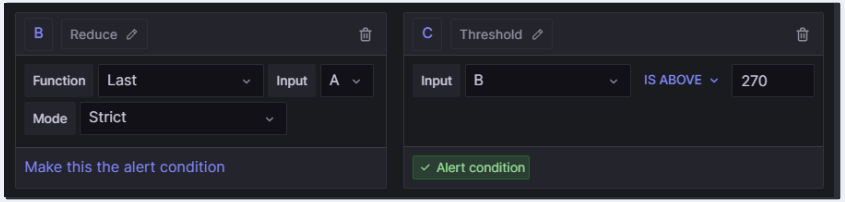
Pick the data source and metric



Our Prometheus data source and the zvm_cpu_utilization metric.

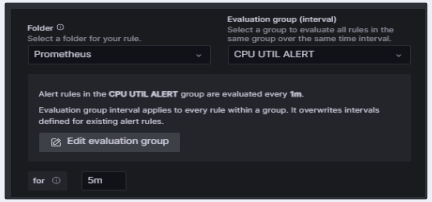
2 THE CONDITION

Last value is above 270%



3 THE EVALUATION

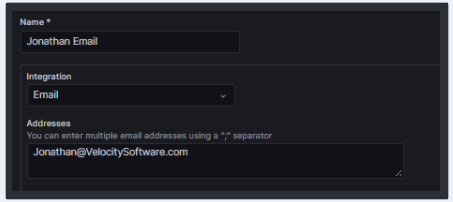
Checked every minute



The evaluation group runs every minute. The “for” field is the pending period.

4 THE NOTIFICATION

Sent to a contact point

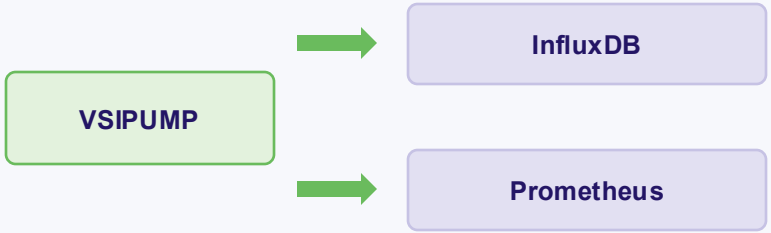


No custom labels, so the default policy sends it to this email contact point.

The Next Question

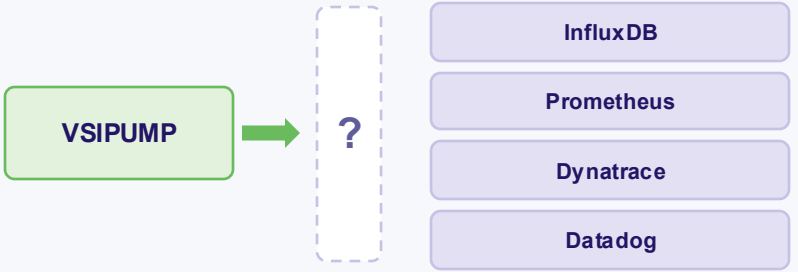
What if I want to send my data elsewhere?

VSIPUMP 1.0



Every new tool meant a new program due to different formats.

VSIPUMP 2.0



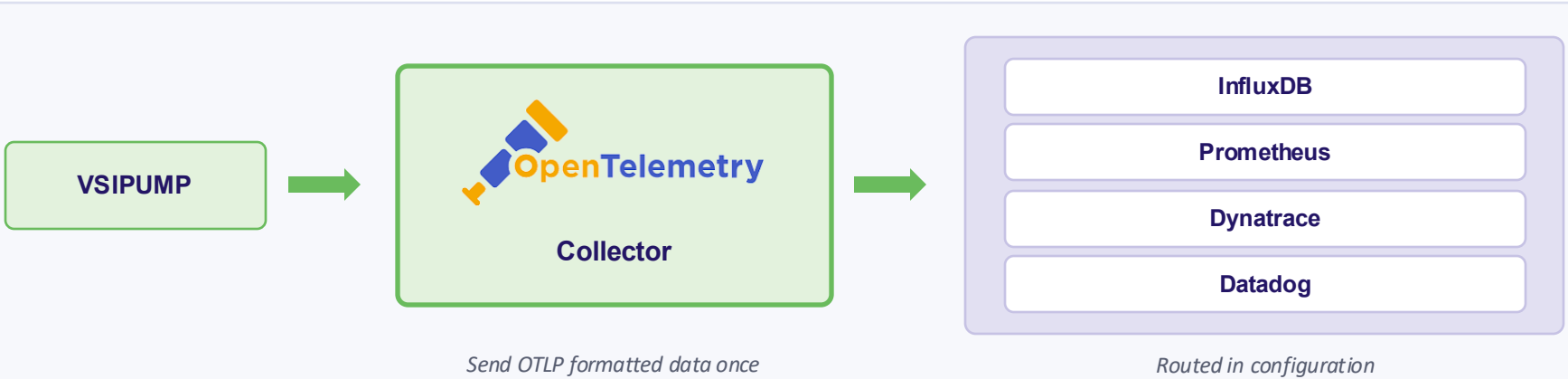
Every new tool is now just a change in the configuration.

So, what goes in the box?

One piece in the middle that can talk to every one of these tools — so VSIPUMP only has to talk to it and it will be able to distribute the data.

Introducing OpenTelemetry

The missing piece



RECEIVE

Send data to one spot

An OTLP receiver takes in everything VSIPUMP sends.

PROCESS

Processes data on the way

Optional: batch, filter, rename, or tag metrics before they leave the collector.

EXPORT

Use an exporter

One exporter per tool, each writing the data in the format that tool expects.

The Collector

collector.yaml file

```yaml
receivers:
  otlp:
    protocols:
      http:
        endpoint: 0.0.0.0:4318
```

```yaml
exporters:
  influxdb:
    endpoint: http://influx:8086
  prometheus:
    endpoint: 0.0.0.0:8889
  dynatrace:
    endpoint: "https://<env-id>.live.dynatrace.com/api/v2/otlp"
    headers:
      Authorization: "Api-Token ${env:API_TOKEN}"
  datadog:
    endpoint: "https://otlp.datadoghq.com"
    headers:
      dd-api-key: "${env:API_KEY}"
```

```yaml
service:
  pipelines:
    metrics:
      receivers: [otlp]
      exporters: [influxdb, prometheus, dynatrace, datadog]
```

RECEIVE

One way in the collector

VSIPUMP sends everything here, over OTLP.

EXPORT

One block per tool

Add a block per tool that you want to send data to. The block will tell the collector how to reach that tool.

PIPELINE

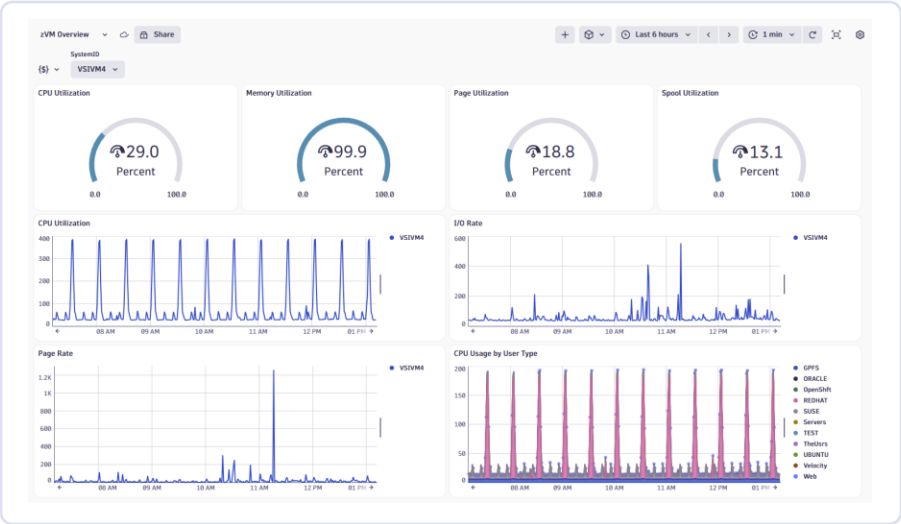
The one line that decides

Whatever is listed here is where the data goes.

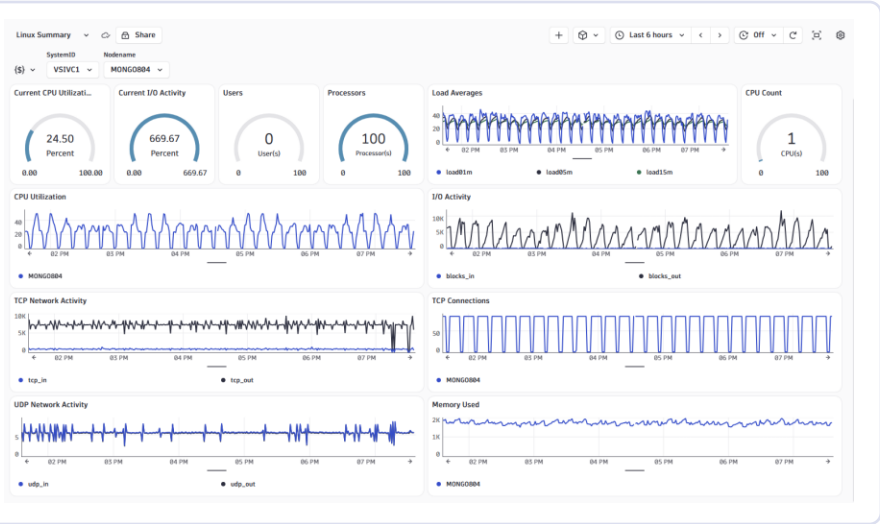
Dynatrace Dashboards

The same metrics & dashboards, in Dynatrace

z/VM Overview



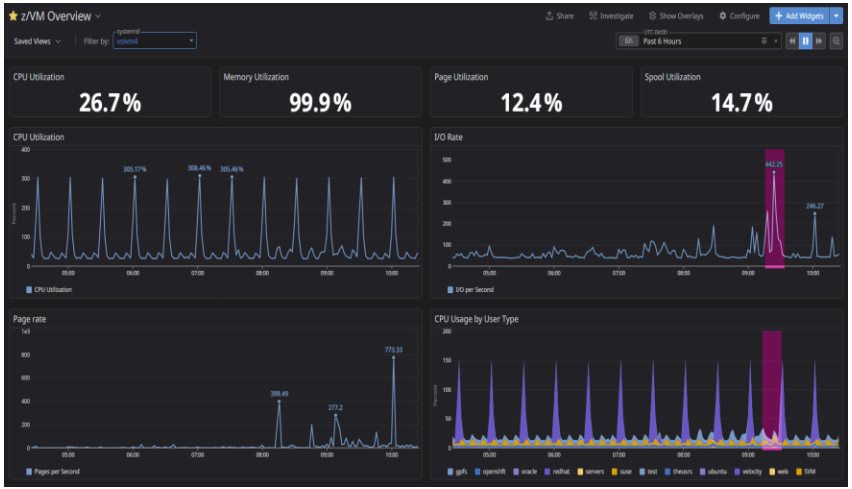
Linux Summary



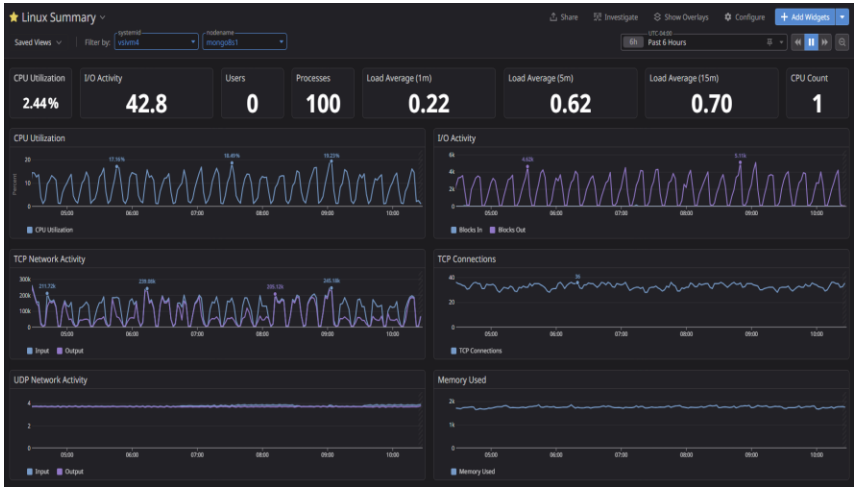
Datadog Dashboards

The same metrics & dashboards, in Datadog

z/VM Overview



Linux Summary



OpenTelemetry Exporters

Many tools to send metrics to via OpenTelemetry

Metrics databases

- Prometheus
- InfluxDB
- Graphite
- ClickHouse

Observability platforms

- Dynatrace
- Datadog
- Splunk
- New Relic
- Grafana Cloud

Cloud monitoring

- AWS CloudWatch
- Azure Monitor
- Google Cloud

Pipes and files

- Kafka
- File
- Another collector

Each one is another block in the same collector file.

Which Path to Pick

Which path should you pick?

InfluxDB

Push · line protocol

PICK IT WHEN

You're starting from scratch and you want Grafana.

The shortest path: VSIPUMP pushes, nothing on z/VM listens, two setup steps.

TRADE-OFF

It only feeds InfluxDB. Other tools need another path.

Prometheus

Pull · OpenMetrics

PICK IT WHEN

You already run Prometheus for other systems.

z/VM becomes one more target in the scrape configuration.

TRADE-OFF

Prometheus connects in to z/VM, so that port has to be open.

OpenTelemetry

OTLP · Tool Based

PICK IT WHEN

You use a tool that has a direct exporter in OpenTelemetry.

VSIPUMP sends once, and the Collector routes it in configuration.

TRADE-OFF

One more piece to run and keep up to date.

Why not always the Collector? It can feed InfluxDB and Prometheus too, but if Grafana is all you need, going direct is simpler. Add the Collector when a second tool needs the data.

The Future of Observability

Findings from Grafana's 2026 survey

Grafana Labs Observability Survey 2026 · 1,363 responses

77%

say centralizing observability has saved them time or money

IN THIS TALK

Putting z/VM data in the same tools as everything else.

65%

invest in both Prometheus and OpenTelemetry

IN THIS TALK

The two paths we covered today, and both work with z/VM data.

47%

increased their OpenTelemetry investment in the past year

IN THIS TALK

The path we used for Dynatrace and Datadog.

About the survey: Grafana Labs' fourth annual survey, with responses from engineers, SREs, and technology leaders in 76 countries.

The Future of Observability

What people want from AI

Grafana Labs Observability Survey 2026 · 1,363 responses

92% see value in AI spotting issues before they cause downtime

Our CPU alert uses a threshold we picked ourselves.

92% see value in AI generating dashboards, alerts, and queries

What the Assistant did for our z/VM dashboard.

77% see value in AI taking action on its own

But 15% don't trust it to do that yet.

95%

say it's important for AI to explain its reasoning

IN THIS TALK

The Assistant told us what it guessed.

The takeaway: z/VM data doesn't have to stay on 3270 screens that only the mainframe team looks at.

Conclusion

Thank you for listening!

Jonathan Castro
Software Engineer
Jonathan@velocitysoftware.com

