GitLab

Prometheus Deep Dive

2019-11-21

Ben Kochie

- Site Reliability Engineer / GitLab
- Prometheus team
- Exporter maintainer

Prometheus 101

- Started in late 2012
- Created to solve real-world problems
- Metrics-based monitoring system
- Polling pre-aggregated event metrics

So I installed Prometheus, now what?

Reading material

- RED Method
  - https://grafana.com/blog/2018/08/02/the-red-method-how-to-instrument-your-services/
- USE Method
  - http://www.brendangregg.com/usemethod.html
- Monitoring Distributed Systems
  - https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems/
- Practical Alerting
  - https://landing.google.com/sre/sre-book/chapters/practical-alerting/
- Prometheus official docs
  - https://prometheus.io/docs/introduction/overview/

Distributed system

Designed to be distributed

- Minimal dependencies
  - Local disk
  - Network
- Intentionally un-coordinated distributed system
- Run Prometheus close to your targets
- Vertical sharding before horizontal sharding

Prometheus Time-Series Database

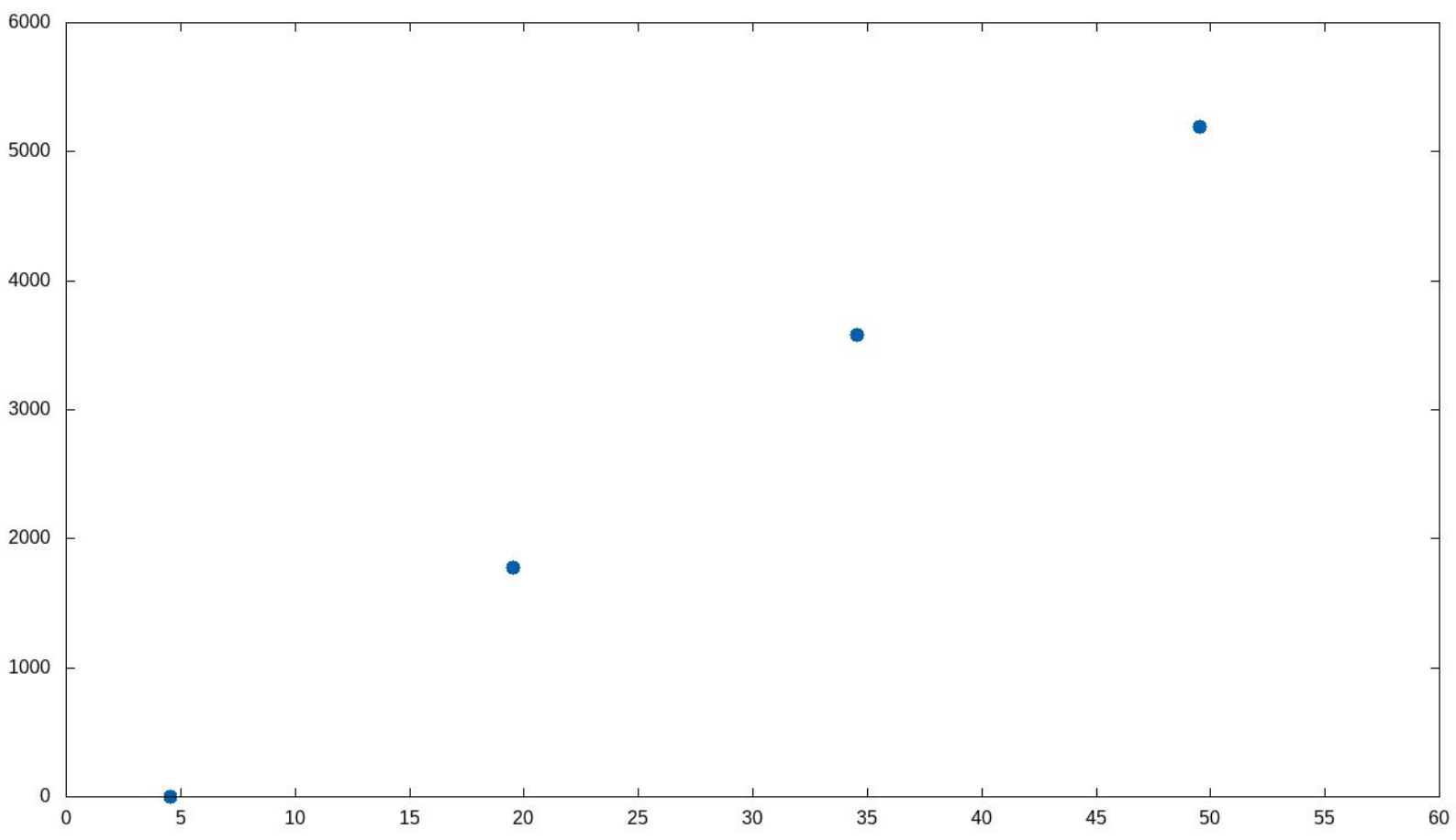
- Write-Ahead-Log (WAL)
- Immutable blocks of data
  - Inverted Index
  - Compressed Chunks
- Compaction
- Simple and robust

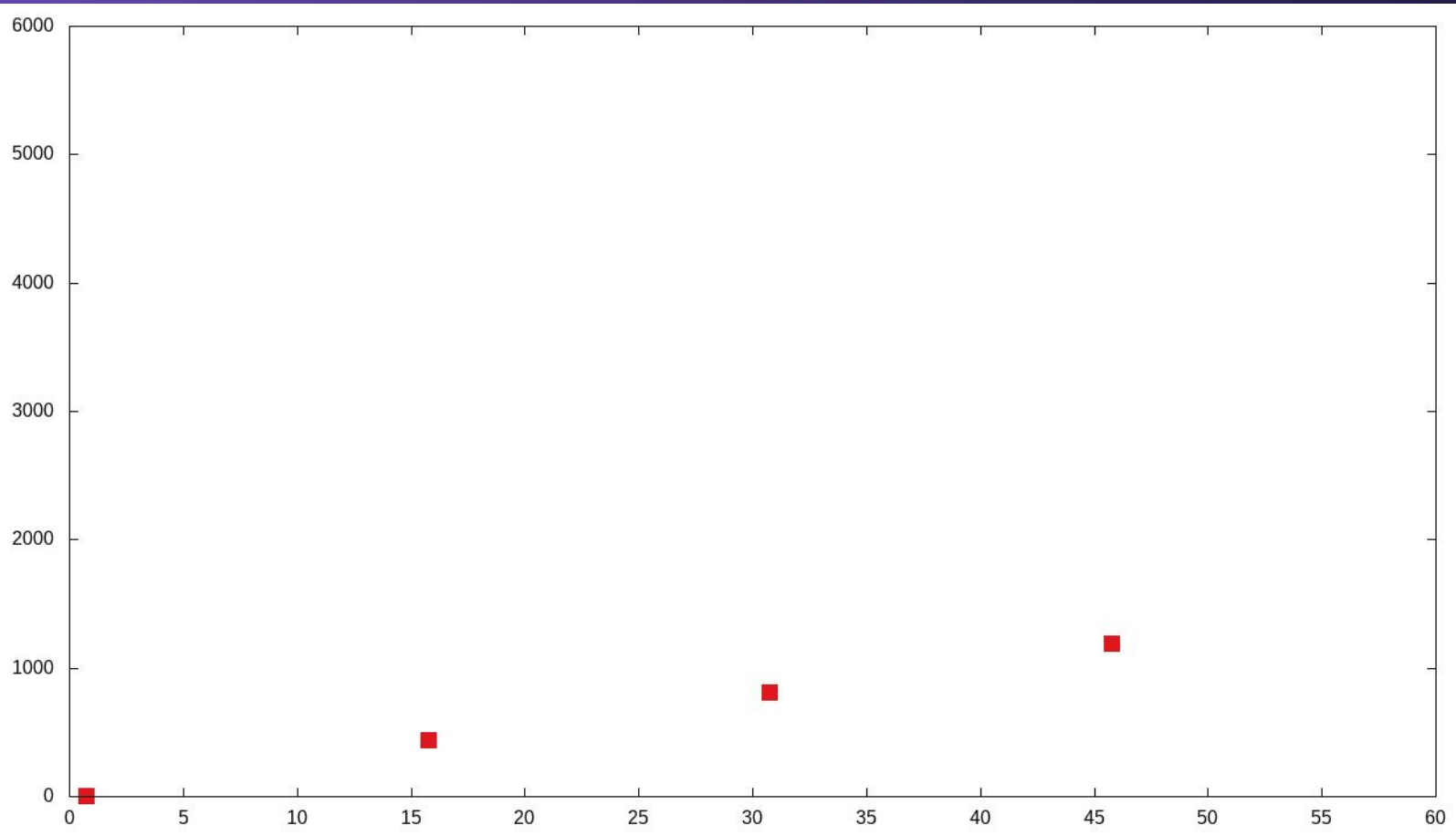
Reference: https://promcon.io/2017-munich/talks/storing-16-bytes-at-scale/

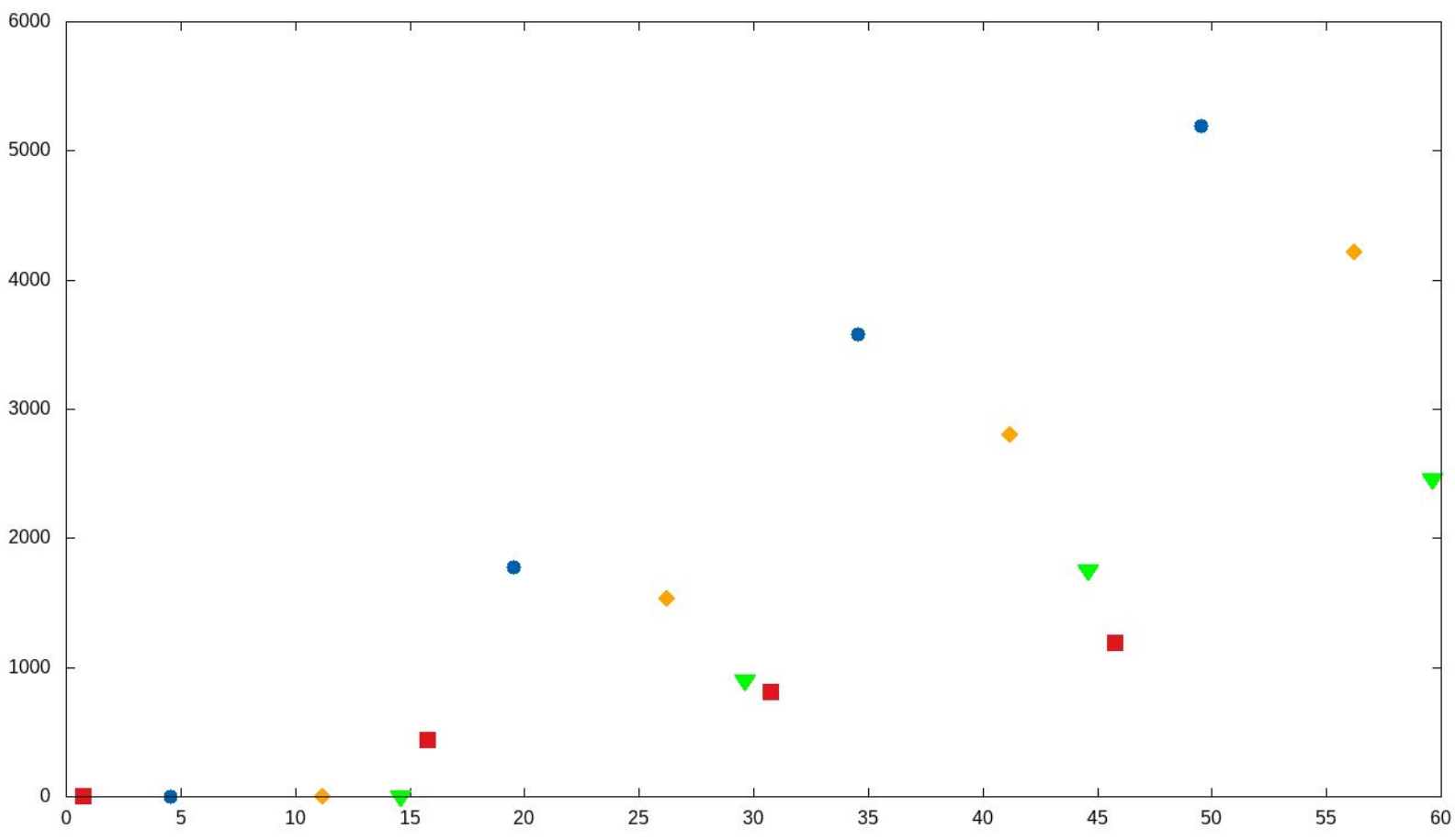
FAQ:

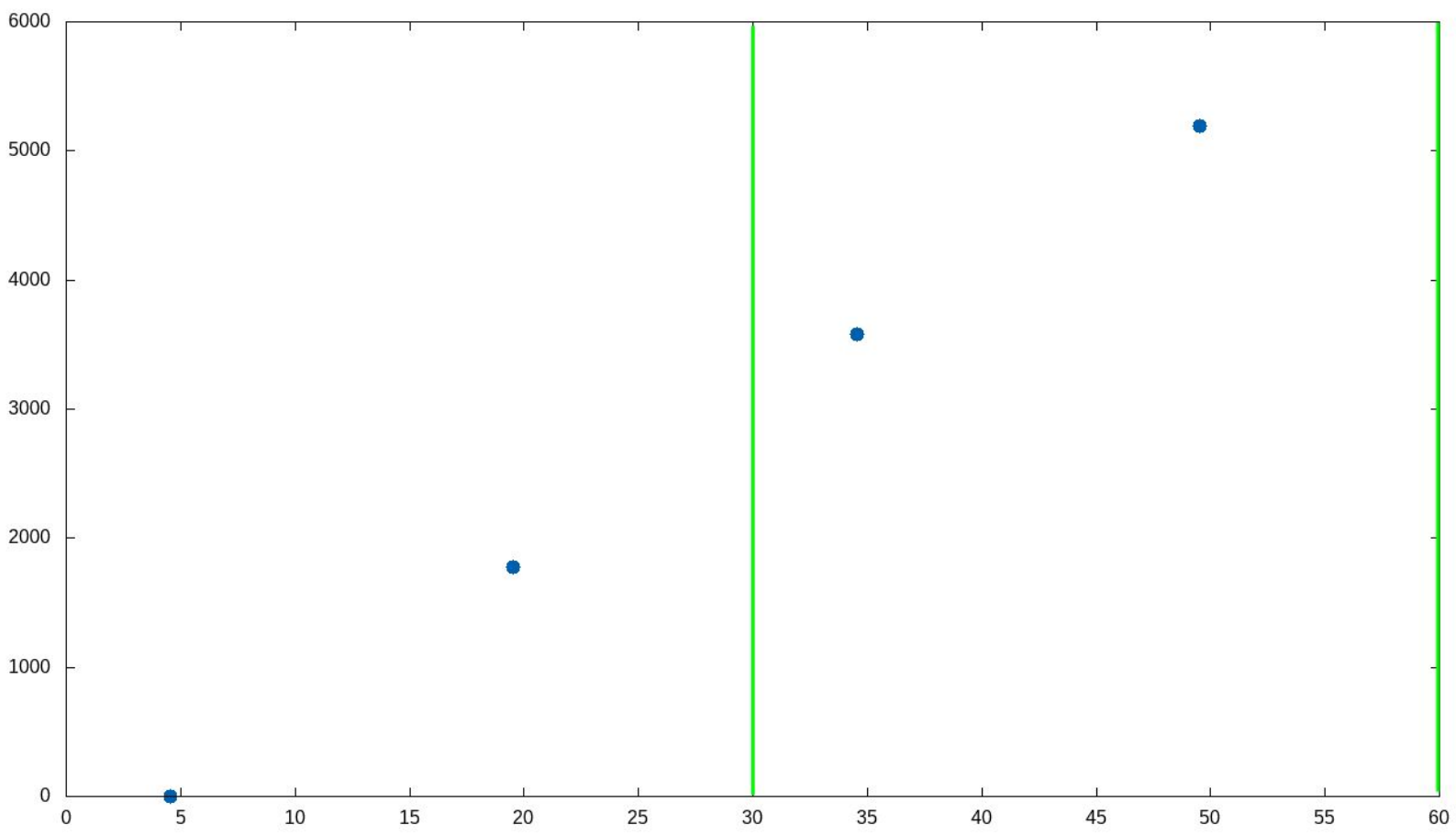
Q: Why do I get float values for increase() when my counter goes up by integers?

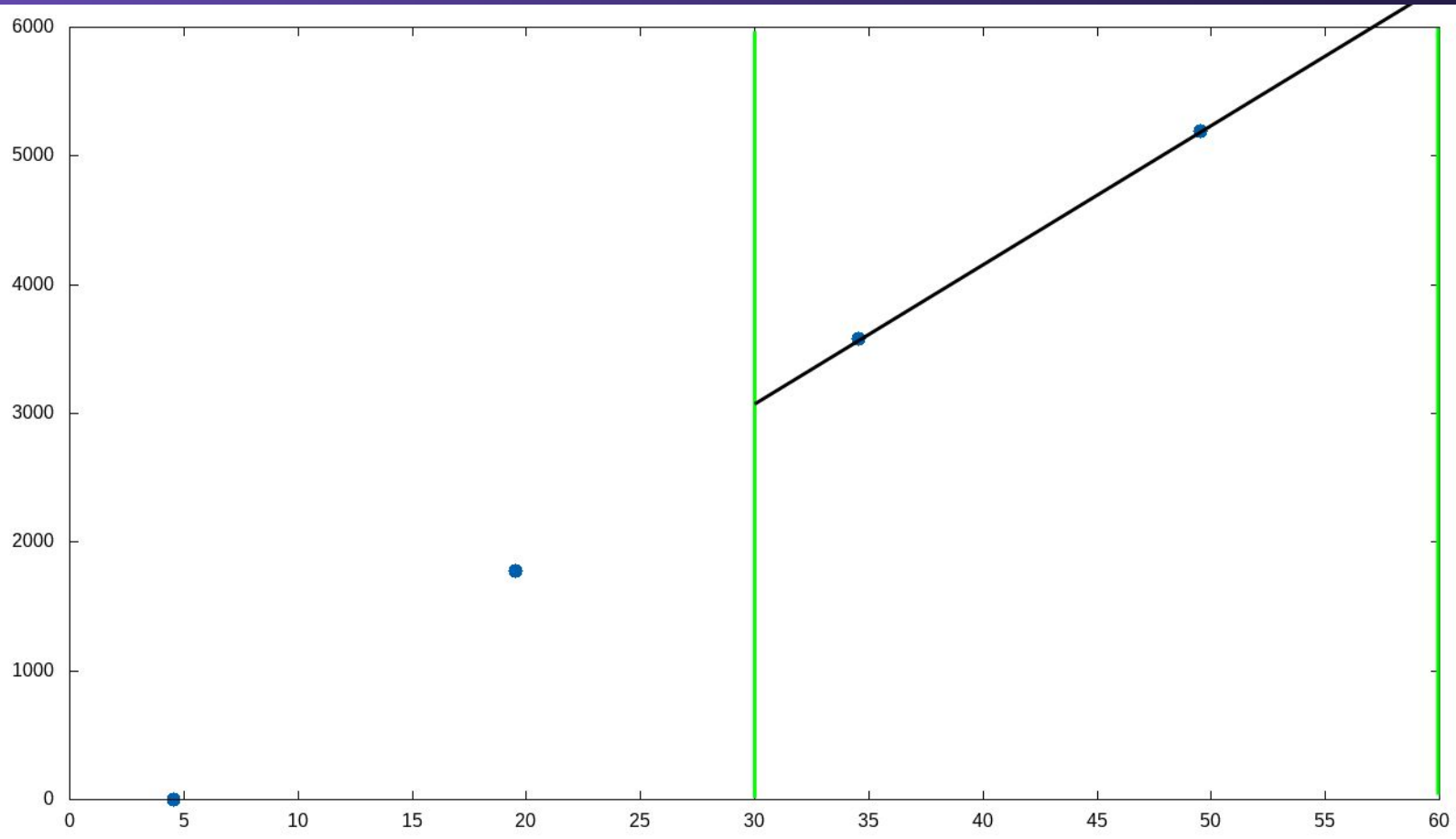
A: Interpolation

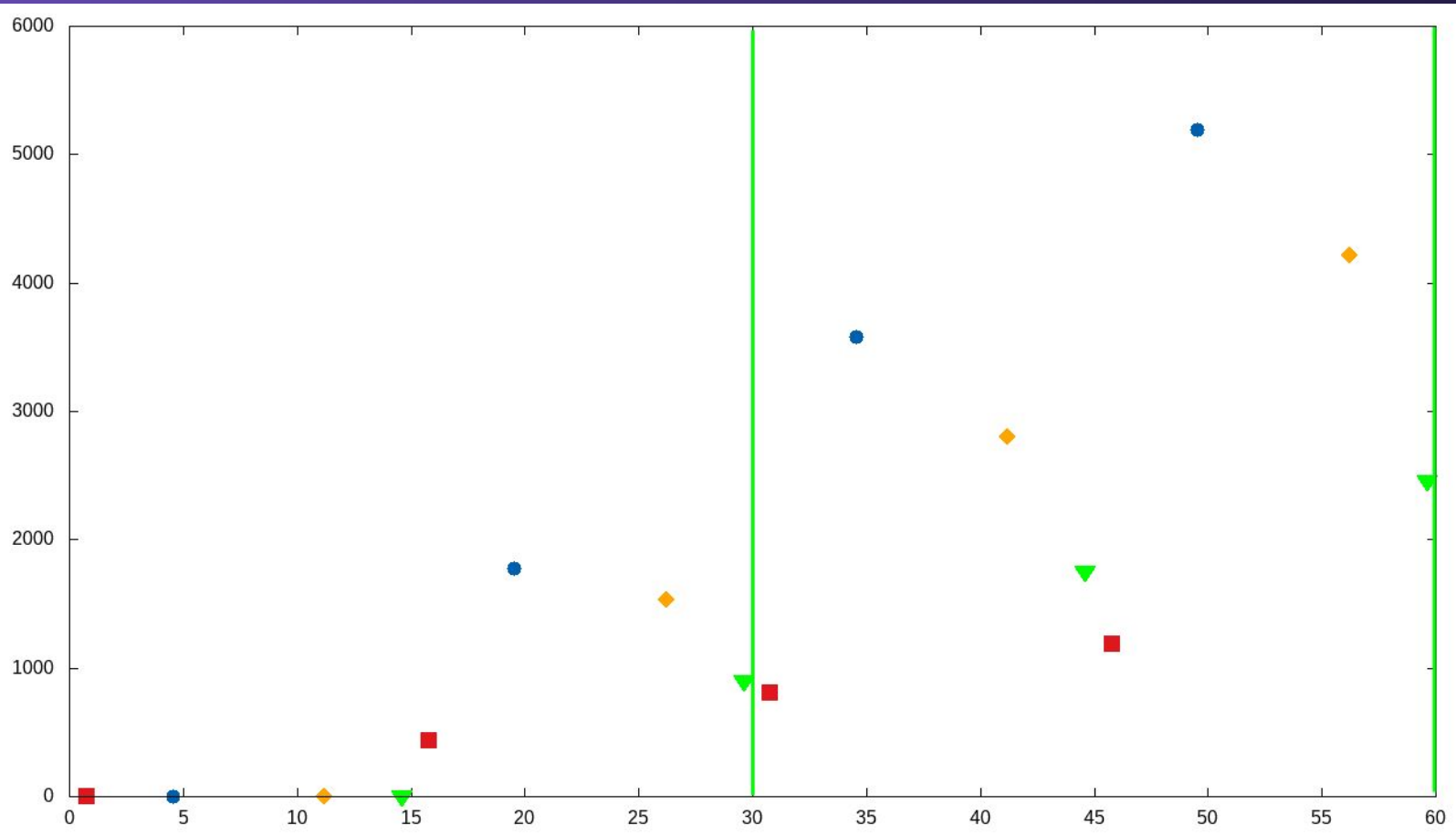

6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60

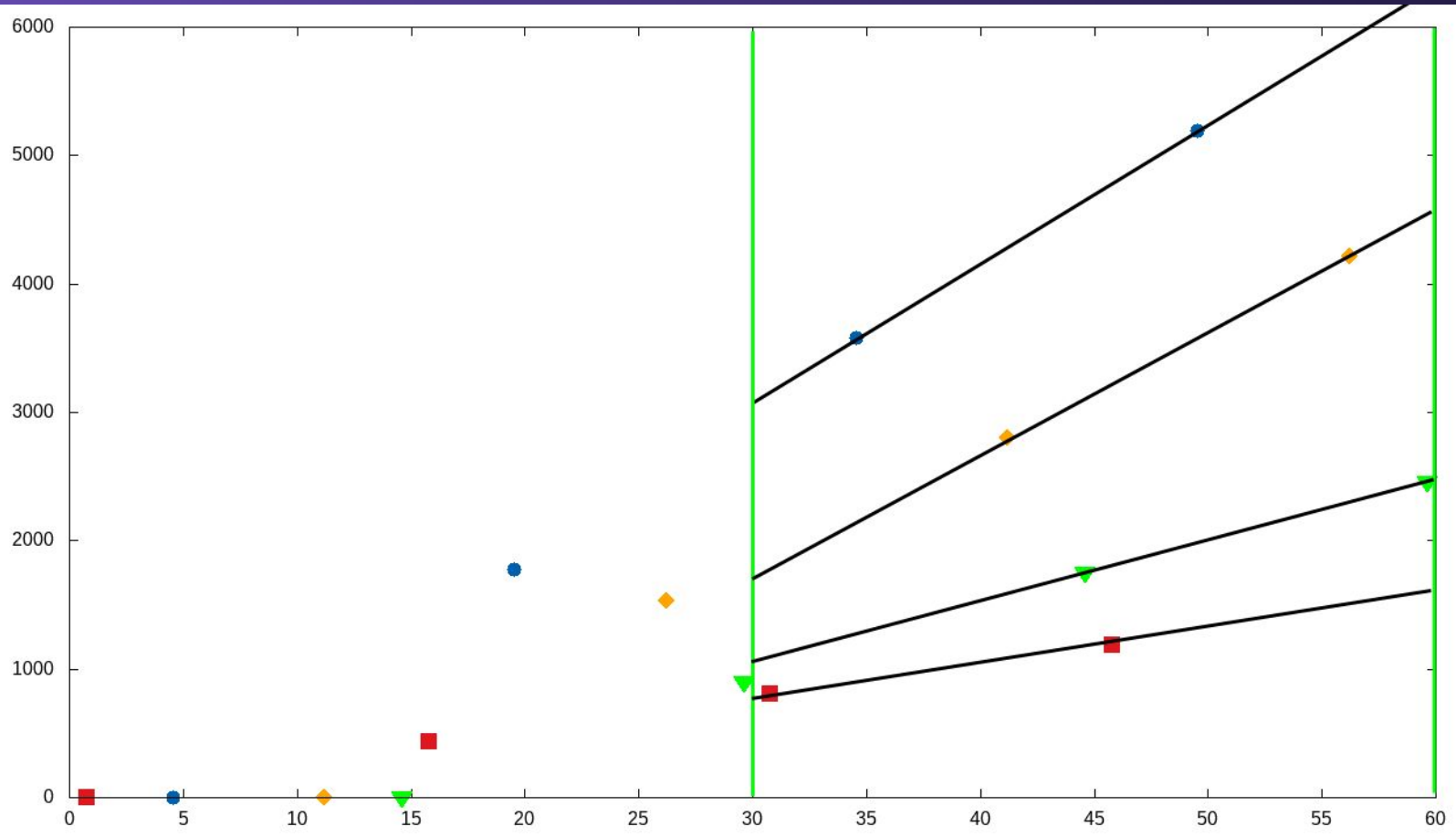

6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60


6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60


6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60


6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60


6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60


6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30
35
40
45
50
55
60

Scaling

FAQ:
Q: How much space do I need for Prometheus?
A: It depends on your ingestion rate and data.

Typical formula: 1.5 bytes per sample per second

Example server:

- ~1700 targets
- ~700 metrics/target
- 15 second scrape interval (some 5 second scrapes)
- lots of recording rules.

100,000 samples / second * 1.5 bytes* 60 seconds * 60 minutes = **0.5GB/hour**

Scaling

FAQ:

Q: How do I deal with multiple instances?

A: External clustered storage

- Cortex - https://github.com/cortexproject/cortex
- M3DB - https://www.m3db.io/
- Thanos - https://thanos.io/
- Others

Thanks!

Questions? Demo?