Quick comparison of Webjive vs Grafana

2020-05-26 Giorgio Brajnik

Introduction

This short memo is triggered by the demo done by Matteo Di Carlo on a <u>solution based on</u> <u>Tango devices</u>. <u>Prometheus</u>. <u>Grafana</u>. I see that as a potential alternative to Webjive</u>.

Outline of the architecture

As far as I can tell the architecture involving Grafana comprises:

- a Tango server
- Prometheus, a tool that collects timed data points from "exporters" and stores them
 into an efficient time series database. Each metric in the database is basically a
 timestamped tuple with whatever attribute we want, a float value and a millisecond
 timestamp. For example a metric can be made of a collection of <timestamp, name of
 device, name of attribute, value>. There can be more than one metric in the
 database.
- an exporter, written in python or c++ that uses Tango deviceproxy. The exporter is an ad-hoc http server that Prometheus reads; it can do polling on selected devices/attributes/states.
- Grafana, the editor and runner of visual dashboards.
- Some existing and some yet-to-be-developed plugins in React for Grafana.
- TangoGQL (or TangoREST) to let dashboards send commands to devices.

Extension points include:

- Grafana plugins (in React)
- Prometheus exporters (Python, c++, ...)

Pro's and con's wrt WebJive

	WebJive	Grafana, Prometheus + REST
web-based	I	
user management	minimal	
dashboard management: access privileges, show/hide, folders	minimal	I
dashboard navigation	todo	I
performance assessment	partially done (js dispatcher to reassess)	to do
plugins (in React)	none	I
plugin repository	none	I
attractive look & feel of running dashboards	none	I
flexible grid layout	none	I
rich set of widgets/charts	not there yet	I
usability of dashboard editor	poor	S
conditional styles and expressions	very limited	I
open source	⊘	✔ Apache, MIT
introspection on Tango devices/server	0	limited to what can be expressed as

		Prometheus metrics
Extendability of data that can be displayed		by writing new Prometheus exporters
ability to send commands	>	Needs the Ajax plugin and an http server
interactive synoptic viewer	none	seems to be provided by existing plugin
import/export of dashboards	todo	I
dropdown menu	todo	available in grafana/ui (beta)
visual grouping of widgets	todo	✓ available in grafana/ui (beta)
structural grouping of widgets	todo	✓ available in grafana/ui (beta)
table	todo	✓ available in grafana/ui (beta)
alerts/alarms	todo	grafana has tools, via http, slack, prometheus has push-mode components
log viewer	todo	with existing plugin
digital assets (logo, color theme, L&F)	todo	probably by developing a "Grafana app"

Where we want to go

Examples:

s001-b-0 sb004d s001-b-0.evla.nrao.edu Ant/St:1	. BB:4/5 GUI s/w version: 290ct2010 11:45AM									
Screen Board Control FPGA GUI ErrorCounts Logging Misc										
Name/IP: s001-b-0.evla.nrao.edu 🔴 BOARD 💮 Temperature	Startup Filter Gain Clear All Errors Refresh OFF									
State: running Serial #: 0x004d 🔵 CMIB 🕘 Voltage	CPU Set Filters Set Logical Configuration sec: 10 Refresh									
Version: 20101119-1154 OPWER Temp/Volt Info	Registers Crossbar									
	Hale Configuration Configuration Configuration Configuration Configuration Configuration Configuration Configuration									
Main Logica Configuration Configueue CRC Bitstreamhies Lookupiaoles Confighies Outputbata IIC										
Common Data Path 0 Data Path 1 View Both Data Paths View All Filters										
Data Path 0 Data Path 1 Cardingstip Name V area 037 55597 05014504039										
configuration warne. <u>Costo_027.55507.55514664026</u>	Configuration Name, A_0570_027.555507.55514064020									
SID. 1 BDID. 4 Polarization. R Sunds 1 Success. 2 (1)-01-24T00:00:00.000	SID. 1 DDD. 3 POINTZATION. R CANAS. 1 PRODUCT ACC									
Source of input data: FOPM stream() swPwrinteg: 1000	Source of input data: EODM stream() swPwrinteg: 1000									
Input bandwidth (Hz): 1024000000 Use Test LUT (write-only)	Input bandwidth (Hz): 1024000000 Use Test LUT (write-only)									
LO_o (MHz): 0.0 Load LUTs (write-only)	LO_0 (MHz): 0.0 Load LUTs (write-only)									
fshift (kHz): 0.0 Gating Mode: Invalid	fshift (kHz): 0.0 Gating Mode: 1nval1d									
Net_LO (MHz): 0.0 Pulse period: 0	Net_LO (MHz): 0.0 Pulse period:									
Radar Mode enabled for filter:	Radar Mode enabled for filter.									
Offset for TEX (Hz): Dispersion: Not implemented	Offset for TEX (Hz): Dispersion: 0									
Inv. duration:	Inv. duration:									
Action: Use Filter 0 as template, place SBs in subsequent slots 💌 Act	Action: Use Filter 0 as template, place SBs in subsequent slots									
Set Baseband + Subbands	Set Baseband + Subbands									
Filter SBID SET StdBv swIndex Width CentFreq Valid Filter Wafer	Filter SBID SET StdBy swIndex Width CentFreq Valid Filter Wafer									
2 17 2 3 12800000 32000000 17 2	2 17 2 20 12800000 32000000 17 17 2									
3 17 2 4 12800000 44800000 17 3	3 17 21 12800000 44800000 7 17 3									
4 17 🖌 🖌 5 12800000 576000000 🖌 17 4	4 17 🖌 🖌 22 12800000 57600000 🖌 17 4									
5 17 🖌 6 12800000 704000000 🖌 17 5	5 17 🗹 23 12800000 70400000 🗹 17 5									
6 17 🗹 7 12800000 83200000 🗹 17 6	6 17 V 24 12800000 832000000 V 17 6									
7 17 2 8 12800000 96000000 7 17 7	7 17 25 12800000 96000000 17 7									
8 1/ 8 12800000 6400000 17 8 0 17 8 12800000 1020000 17 8	8 1/ 26 12800000 6400000 2 17 8 0 17 8									
10 17 2 10 12800000 32000000 9 9	10 17 2 28 12800000 32000000 2 9 9									
	11 17 29 12800000 44800000 11 11 11									
12 17 12 12800000 57600000 17 12	12 17 2 30 128000000 576000000 17 12									
13 17 🖌 🖌 13 128000000 704000000 🖌 13 13	13 17 🖌 🖌 31 12800000 704000000 🖌 13 13									
14 17 🗹 14 12800000 83200000 🗹 17 14	14 17 🗹 🗹 32 12800000 83200000 🗹 17 14									
15 17 🖌 15 128000000 960000000 🖌 15 15	15 17 🖌 33 128000000 960000000 🖌 15 15									
16 17 🗹 16 128000000 832000000 🗹 16 16	16 17 V 34 12800000 83200000 V 16 16									
17 17 17 17 128000000 960000000 17 17 17	17 17 2 35 12800000 960000000 17 17 17									



What seems to be possible now

Ø	🗱 DevOpsProdigy KubeGraf Node's Dashboard -		🗚 🕁 🖻 🕸 🖵 💿 Last 30 minutes - Q 🗘 1m -				
+	cluster Test - node minikube -		C [*] DevOpsProdigy C [#] Kub <mark>b</mark> araf ≡ Dashboards				
	Server Stats						
•	CPU Bay RAM lace	N/A Root F3 Used	Sys Load (Sin wa) Sys Load (Sin wa) 16% 26%				
-	> Server Info (6 panels)						
	~ Node Stats						
1	Node CPU Allocation	Node Memory Allocation	Node Pods Allocation				
~	13.55%	11.61%	20.00%				
	Node "minikube" CPU Requested	Node "minikube" Memory Requested					
	26.38%	2.90%	Requested resources				
	> Node Info (3 panels)						
	> Basic CPU / Memory Graph (4 panels)						
	> CPU / Memory / Net / Disk (7 panels)						
8	> Memory Meminfo (15 panels)						
(?)	> Memory Vmstat (3 panels)						
	System Timesvnc (4 panels)						

© +		DevObsPro	drav KubeGraf DevOpsProdigy KubeGraf / Nodes C DevOpsProdigy KubeGraf	verview			
	Cluster Status Applications Over	view Nodes Overview				Dashboards	Edit Plugin Config
•	Overview: Production	. Nodes		✓ k8s-node1	✔ k8s-node2	✓ k8s-node3	Show all Hide all ()
*	Node: k8s-node1 • IP: 192168:101.31 CPU Core: 40 RAM Total: 92.925 CiB SWAP Total: 93.8361 CiB RourFS Total: 439.8361 CiB Sys Load (Im avg): 1.38 Uptime: 55.2 day						
		Pods limits		CPU (cores)		Memory	
	Used	51 (46.36 %)					
	Requested	51 (46.36 %)		4.908 (12.33 %)		10.903 GiB (11.81 %)	
	Allocatable						
	Capacity					92.926 GiB	
	Open Dashboard Hide Application	ions					
	Namespace: inore	seginx (1)↓		CPU usage †	CPU requested †	Memory usage †	Memory requested
	nginx-ingress-controller	-5fb84c58d6-s71bh 👁				970.234 MiB	
8			Summary	145m	0m	970.234 MiB	0 B

ې +		DevOpsProdigy KubeGraf / Applic DevOpsProdigy KubeGraf	ations Overview	
•	Cluster Status Applications Overview Nodes Overview			Dashboards Edit Plugin Config
	- Overview: Production. Applications			1/69 Show all Hide all 🕥
*	 Namespace: kube-system Deployments 	Statefulsets	Daemonsets	SHOW Cron Jobs (0) SHOW Jobs (0) Status: Active Other
	calico-kube-controllers & • calico-kube-controllers 79043c6658 k22tz & contents 644c686c9-kgbbh @ • contents 644c686c9-kgbbh @ Services		Californade en • californade efford e • californade e • californad	
	· coredni: S3/UDP S3/TCP 91S3/TCP dns-autoscaler ↔ • dns-autoscaler;866f58b8bfÅg8b2 ◆		merceat * • fields ck94c as • fields: v39x as • fields: vvild-as • fields: vvild-as	
	kubernetes dashboard 🐲 • kubernetes-dashboard \$457;55(89.ccmaz 👁 Services • kubernetes-dashboard 443/7CP		Nucley Ny ≪ o skole rang-2nat2 ● o skole rang-4227 ● o skole rang-4277 ●	
	tiller-deploy & • tiller-deploy-55558bbcf-95rog & Services • tiller-deploy-44134/7CP			
3				
	The plugin consists of th	area main into pages with detailed information about the Kubern	Links:	

() +		DevOpsProdigy KubeGraf / Appli DevOpsProdigy KubeGraf	ications Overview	
	Oluster Status Applications Overview Nodes Overvie			Dabboards Edt Plugin Config
₩ ₩ ₩	Status Pod Name - Namespace: prometheus Deployments Dometheu-operator gafana //////////////////////////////////	Error message Statefulsets alertmanager prometheus-operator alertmanager # alertmanager operated selectmanager # Berlor alertmanager operated selectmanager # alertmanager operated selectmanager # alertmanager operated selectmanager # alertmanager operator prometheus #	Demonsets Demons	Stow consider (1) Stow Conside
8 7				



ADS State Agent Initalized Computer Started Agent (version)			Run 19 days ago 20 days ago 232a7e7					
2/20	2/22	2/24	2/26					
RUNNING (64%) PAUSI	Syster ED (35%) STARTING (2%)	n State ANTI_ICING (0%)	RUNNING					
System State RUNNING (64%) PAUSED (35%) STARTING (2%) RUNNING 017-02-13 08:05:00 to 1017-02-14 08:35:00 a day								
System State RUNNING (64%) PAUSED (35%) STARTING (2%) ANTI_ICING (0%) PAUSED 24 times for a day total								
	System	n State						
	PAUSED	RUNNING	PAUSED					







Topology Panel is an Sigma.js-based plugin for Elasticsearch and InfluxDB.

SVG Builder

Panel Title												
SVG _G	eneral Metrics SVC	Builder SVG	Events Time rar	уge								×
Use SVG Builder	ß											
Canvas		Elements										
Width	100%	Name				rotation	r-center x	r-center y	scale			
Height	100%	plant	plant	100	100							
Viewport X		square	square	200	600	45					٥	
Viewport Y		light-green	light-green	220	500						۵	
Viewport Width	1000	Add new										
Viewport Height	1000	Repository	rtmaster									
Categories Indicators												
		SVG	light-green									
		+ Add										

"Navigating" through Tango devices:

uses a simple combo box:



The question is "pivot or persevere".

Shall we continue working on Webjive as planned? or shall we explore the potential of this "new" solution? If so, shall we do it in PI7?

Giorgio thinks that:

- 1) The solution based on Grafana is more mature and tested than Webjive.
- 2) It yields GUIs that are visually more attractive, with a richer set of widgets from which to choose.
- The dashboard editor provides many of the features that are missing and planned for Webjive.
- 4) It needs more development in the area related to controlling devices and in the area of associating Tango devices/attributes/states to widgets.
- 5) We will have more difficulties in changing the internals (of Grafana, Prometheus) in case something will appear not to be working (for example in terms of performance bottlenecks).

- 6) We will need to redo the performance assessment.
- 7) We will need to carefully assess the reliability of exporters.
- 8) The Tango community will have a web-based GUI toolkit in any case.

All things considered, Giorgio thinks that the current goal in PI7 (<u>building attractive</u>, <u>connected and complex dashboards</u>) could be achieved also with this new solution, instead of using Webjive. If we can achieve that, then this fact could be the demonstration that such a solution works.

After all we already have the basic building blocks (Grafana, Prometheus, TangoGQL, an exporter, an example of a button widget).

We need to focus on building the needed dashboards and 1-2 new plugins for Grafana for entering commands.

In doing this work we can assess the possible difficulties due to the poor Tango introspection capabilities, but for the moment live with what is now possible with Prometheus and Grafana.

Decision

After a discussion between Giorgio and Nick:

- 1. PERSEVERE with webjive.
- Explore potential strong negative risk with Grafana + Prometheus that has to do with inability to sustain a high data rate for monitoring (the 1000 updates in less than a second) and a small latency when issuing commands. See feature <u>https://jira.skatelescope.org/browse/SP-1089</u>.