

Monitoring Agents Comparative Study

The tables and lists of questions have been created by [Sridhar Rao](#) <Sridhar.Rao@spirent.com>

There are numerous opensource monitoring solutions available, with varying approaches and architectures. In this study, we compare only the 'agent' component of the monitoring solution, and will not consider the server-side component(s). Because, there can be multiple implementation options of the 'server' - for example, with collectd, it could be simple collectd-web or a timeseries database such as Influxdb, telemetry system based on Apache Kafka, etc. - and considering all the options would be extremely difficult. Typically the server side components could include some or all of the following (a) Metric collection infrastructure - raw-metric receiver, message-queues, etc. (b) Metric Modifier - add contexts, perform-aggregation, filter, etc. (c) Storage solution (d) Alarm/Alerting System (e) Visualization/Graphing - dashboards. (f) Publishing.

Terminology Definition

Term	What we mean by that?
Metric	A Measurement of a particular characteristic. Ex: %ge of CPU used, Amount of Bandwidth used, etc. Complete definition can be found here
Event	A record of something that has happened - A simple immutable fact. Example: Link has gone down. A packet from a flow is dropped, etc. Complete definition can be found here
Agent	Software that runs on a node/system that needs to be monitored.
Client Node	A node that is monitored (Node on which agent runs)
Server Node	A node that collects metrics and events from the client node.
Sampling Interval	How frequently the metrics are sent.
Push Mode	Fetching of events by subscribing
Poll Mode	Fetching of events via polling.
Writing of Metrics/events	sending/outputting of metrics or events.
Reading of Metrics/events	receiving/reading of measurements
Logging of Metrics/events	Logging of monitored/received metric or event
Metric Types (data source types)	Guage: Value stored as-is Derive: Derivative - Change of the value (rate) Counter: Similar to Derive - but it is NEVER negative (due to wrap-around) Absolute:

Parameter Table

Parameters\Tools	Collectd	Ceilometer Polling agent.	Monasca	SNAP	node-exporter and other exporters	sensu client: metric collection plugins	munin	telegraf	NRPE + Plugins (NSClient++, ICINGA, OpenNMS)	diamond	Reimann	Elast Beat:
CPU metrics	idle, system, wait, stolen, user (% & time), util, vcpus	idle, system, wait, stolen, user (% & time), util, vcpus	idle, system, wait, stolen, user (% & time)	idle, system, wait, stolen, user, guest, irq, nice (% & jiffies)	idle, system, wait, stolen, user (% & time), util, vcpus	idle, system, wait, stolen, user (% & time), util, vcpus	Freq. usage - idle, system, wait, user, vcpus.	Same as ceilometer or monasca	user, system, iowait, idle in (% and time). average-load	idle, system, wait, user, nice.	idle, system, wait, user, nice, stolen, irq	idle, system, wait, us nice, stc irq

Disk IO metrics	Read and write (bytes, rate, time, sectors) disk-free	read and write (bytes, rate, req)	read and write (bytes, rate, req)	read and write (ops, octets, merged, time) disk-free	read and write (bytes, rate, req)	Read and write (bytes, rate, time, sectors)	read and write (bytes, rate, req)	Same as ceilometer or monasca	read and write (ops, octets, merged, time) disk-free	read and write (bytes, rate, req)	read and write (merged, sector, time, req) io- reqs, time, weighted	read an write (count, ' and byt
Memory metrics	free, swap, total, used (bytes and perctages)	usage, bandwidth	free, swap, total, used	free, available, total, used.	free, swap, total, used	free, swap, total, used (Mb and percentages)	free, swap, total, used, slab.	Same as ceilometer or monasca	free, available, total, used. (bytes, %ges)	free, total, swap, active, dirty, inactive, buffers.	free, used, (bytes and % ges) actual-used.	free, us (bytes a %ges) actual-u
Process metrics	I/O, memory, CPU-Usage, read-write (bytes and count)	NO	NO	I/O, memory, CPU-Usage, (bytes and count).	Same as collectd.	status, thread-count, uptime. IO, memory, cpu-usage, connections.	Cpu and memory, read-write (bytes, count), and various other fields	Cpu and memory, read-write (bytes, count)	CPU, memory, uptime,	btime, cxtt, processes, blocked, running	I/O, memory, CPU-Usage, read-write (bytes and count)	I/O, memory, CPU-Usage, read-wr (bytes a count)
Network Interface Metrics	Interface plugin: Standard 4 fields of rx/tx (octets, packets, errors, dropped). Netlink plugin: uses netlink sockets and covers others	Standard 4 fields of rx/tx (octets, packets, errors, dropped).	Standard 4 fields of rx/tx (octets, packets, errors, dropped).	sent and recv : bytes, compressed, drops, errors, fifo, frame, multicast, packets	Standard 4 fields of rx/tx (octets, packets, errors, dropped).	Standard 4 fields of rx/tx (octets, packets, errors, dropped). Also includes, fifo, compressed, and frame stats.	rx/tx (octets, packets, errors, dropped).	Same as ceilometer or monasca	rx/tx (octets, packets, errors, dropped). SNMP (3)	Rx and Tx. MBs	Standard 4 fields of rx/tx (octets, packets, errors, dropped)	Standar fields of /tx (octe packets errors, droppec
Libvirt Metrics	YES -	YES	YES	YES	YES	NO	NO	NO	YES	YES	NO	NO
Container resource usage Monitoring (memory, restarts, status, uptime, etc)	YES	NO	NO	Docker	Docker	Docker	NO	Docker	YES (Docker, LXC)	Docker	YES (Docker)	YES (4)
Databases Monitoring : [Influxdb, MongoDB, MySql, PostgreSQL, Carbon (graphite), Prometheus, RRDCache,Redis, TSDB]	YES for all	MySql, PostgreSQL, MongoDB	Influxdb, Vertica, MySql, PostgreSQL, Cassandra	Influxdb, mysql, mongodb, Cassandra	ALL (4)	All	NO	All.	YES for all	MongoDb, mysql, postgresql, and Redis	YES for all	YES for (4)
Publish metrics to databases - (influxdb, mysql, TSDB, PostgreSQL, MongoDB, Carbon, Elasticsearch)	YES for all	NO	NO	YES for all.	NO	NO (1)	NO	Yes for all	NO	Yes for All	YES for all.	YES (4)
Encryption Support	YES	NO	NO	YES	NO	NO	NO	NO	YES	YES	YES	YES
Language (written)	C	Python	Python	Go	Go	Ruby	Perl	Go	perl, shell, c, (varies)	Python	Varies - ruby, c, c++, etc.	Go
Extensibility - multilanguage support [Python, Java, Golang, C/C++, Lua]	YES for all	Java	Java	Python C++	Java, Python, Ruby	Go, Python.	Python, Ruby	None.	Perl, shell, C.	None	Multiple	NO?
Interoperability [with other monitoring solutions]	Sensu, statsd, telegraf?	Nagios zabbix	ceilometer	Ceilometer , Facter, Reimann, Prometheus	Collectd	Nagios, Zabbix.	NO	Reimann	NSClient, Icinga.	Nagios	Collectd	Collectc
Write to Message Queues and protocols (AMQP, Kafka, MQTT, NSQ)	YES for ALL	AMQP	Kafka	AMQP, Kafka.	NO	AMQP	NO	kafka, MQTT, NSQ	NO	Yes for ALL	YES for all	YES for (4)
Metrics Pub/sub Mode Support (Metrics push/pull mode support ?)	YES	YES	YES	YES	YES	YES	NO	YES	NO	YES	YES	YES
Metrics Req/Resp Mode Support	NO	NO	NO	YES	NO	YES	YES	NO	YES	NO	YES	YES
Support for Events (polling, Pushing)	Yes	NO (1)	NO (1)	NO	NO	YES	NO	YES	YES	NO	YES	YES
Notification Support	YES	NO (1)	NO (1)	NO	NO (1)	YES	NO	NO	YES	NO	YES	YES
Logging Support	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Hypervisor metrics	YES	NO	NO	YES (KVM)	YES	YES (XenTop)	NO	NO	YES	XEN, KVM.	NO	NO
Log-File Analysis	YES	NO	NO	YES	YES (mtail)	NO	NO	YES	YES	NO	YES	YES

Other Writing (output) Support: [CSV, HTTP, RRD, UnixSocket, Multicast]	ALL that are listed.	NO	NO	NO	HTTP	NO	RRD	Socket,	NO	HTTP	NO	YES?
Transport Protocol	Depends on the end point it's communicating with.	TCP*	TCP*	TCP	TCP, UDP. (5)	TCP	TCP	TCP, UDP	TCP	TCP	TCP, UDP	TCP, UI
Data-Format [XML, JSON, etc]	JSON, Custom, XML	JSON XML	JSON	JSON	JSON ?	JSON	Custom	Custom	Custom	JSON	Custom	Custom JSON
Data-model	Custom	KVP	KVP	KVP	KVP	KVP	Custom	Custom	Custom	KVP	KVP	KVP
Hardware: IPMI, Battery, Sensors,	YES for all	IPMI	IPMI	IMPI	YES for all	YES - IPMI	YES (3)	IPMI sensors	YES	NO	NO?	YES for
Metric Types: Gauge, Derive, Counter, absolute	YES for all	Gauge cumulative delta	Gauge, rate, counter.	gauge, derive, counter.	Gauge, Counter, Histogram, summary	Gauge, Counter, derive.	Gauge, Counter, derive.	Gauge, Counter.		Gauge, Derivative, delta	Gauge, sum, counter, derive	Gauge, sum, counter derive
Last-Updated	2017	2017	2017	Varies(5)	Varies (5)	Varies (5)	Varies (5)	2017	varies(5)	Varies (5)	Varies(5)	Varies(5)
Commercial Versions?	NO	NO	?	NO	NO	YES	NO	No	YES	YES?	YES?	YES?
Run-Time Analysis [^]	CPU: 14.8% VM: 958205952 RSS: 2066 Code: 14442496 Data: 831705088 StackSize: 2288							CPU:17.5% VM: 345899008 RSS:7880 Code: 15036416 Data: 321084800 StackSize: 416				
License	MIT/GPL v2 or later	Apache License, Version 2.0	Apache License, Version 2.0	Apache License, Version 2.0	Multiple (5)	MIT	GPL V2.	MIT	GPL V3	MIT	MIT	Apache License Version
Webserver monitoring [Nginx, Apache]	YES for all	Apache	Apache	YES for all.	Nginx, Apache, Passenger varnish	Apache, Nginx, Unicorn.	NO	Yes for all	YES for all	NO	YES for all.	Yes for
Platforms - OS? Linux (unix'es), Windows.	Supports windows, linux, freebsd, etc.	Linux	Linux	Linux, MAC, Windows (soon)	Linux Windows(3)	Linux, Windows,	Linux, Windows	Linux	ALL	Linux	ALL	ALL
Configuration Tool support [Puppet, Chef, Ansible, Salt]	YES for all	Puppet Chef	Puppet, Chef, Ansible,	Yes for all.	Yes for all.	YES for all	NO	Yes for All.	Yes for all	Puppet	ALL	ALL
Deployments: servers, VMs, containers,	ALL	ALL	ALL	ALL	ALL	ALL.	ALL	All	ALL	ALL	ALL	ALL
Openstack Modules	NO (2)	NO	ALL.	CEPH, Cinder, Glance, Keystone, Neutron, Nova	NO	NO	NO	NO	YES (All)	NO	NO	NO
Intel PCM and SSDs SMART metrics	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO
Cluster Mgmt. (Kubernetes, Mesos, Swarm)	NO	NO	NO	Kubernete s and Mesos	Kubernetes and mesos	Kubernetes and mesos	NO	Kubernetes and Mesos	YES	NO	YES	YES
Modifiers - (filtering, threshold, tags, contexts)	Filtering and threshold - yes. Tags - YES. Contexts - No. (1)	NO	YES	YES for all.	Tags, Filtering and threshold.	NO(1)	NO	Tagging	YES	Tags	YES	YES
Dynamic Loading of plugins.	NO	NO	NO	YES	YES	YES.	YES?	NO	YES	NO	YES	YES
Intervals:	LSI: can go down to a nano second resolution NTI: Cannot be specified - depends on size of the buffer and reading interval	PI: Configurable default: 60s	CF: Configurable Further controlled by per-plugin "collect_period"	Based on Task Configuration. Interval: Can go down to ns resolution.	Configurable scrape interval.	Command specific check-interval.						

Other Services monitoring: (DHCP, DNS, FTP, NTP, HAProxy, Consul)	HAProxy, DNS, NTP	NO	HAProxy, NTP.	HAProxy	DHCP, HAProxy, NTP, Consul.	YES for all.	NO	HAProxy, NTP, Consul, DNS,	YES	NO	YES (4)	YES(4)
--	-------------------	----	---------------	---------	-----------------------------	--------------	----	----------------------------	-----	----	---------	--------

Legends

- (1) This aspect is realized either as a server-side component or by a 'customized' agent.
- (2) Custom solution exist, and may not be part of main distribution.
- (3) Support with strong dependency on additional tool/library.
- (4) Supports more-options than the ones provided in column-1
- (5) A single value cannot be entered due development of logically-independent modules by different community groups.

[^]: Runtime analysis process and considerations:

1. Isolate the CPUs on the monitoring node. [Add isolcpus option in the grub. CPU0]
2. Run the agent on the isolated CPU (CPU0). [Use taskset command to run agent-processes with appropriate CPU-mask: 0x01]
3. Plugins: Configure agent to monitor following metrics - CPU, Memory, Disk, Interface, IPMI, processes, libvirt, Caches, OVS, hugepages.
4. Output: Make agent to send metrics over network (Ex: influxdb running on separate node)
5. Workload: stress-ng + iperf.
6. Monitoring duration: 5 minutes.
7. Frequency: 1sec.
8. Collect Metrics (using any other tool) to analyze agent's runtime performance [Ex: Used Snap to collect 'collectd-process' metrics and CPU and memory data]
9. Note the iperf performance (to study any effect on it due to collectd]

#: Interval Definitions:

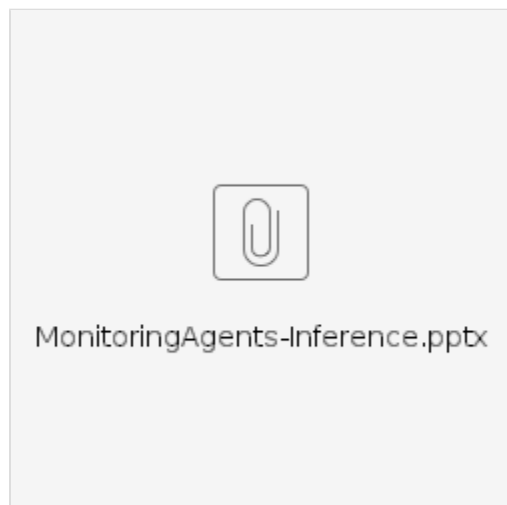
Lowest Sampling Interval (LSI) - How frequently the plugins can read values from source(s) of truth.

Network Transmit Interval (NTI) - Interval at which the metrics are sent over the network.

Polling Interval: Frequency at which metrics are read.

Check Frequency(CF): frequency at which all plugins are run. This may map to LSI and NTI.

Inference Questions



The Questions	The Answer
Lowest Interval: Which agent supports the lowest sampling interval, and what is the value?	
Interoperability: Which agent is 'most interoperable'? (Work with maximum of 'servers' (collection node)	
Large-scale deployment: Which agent is ideal for large-scale monitoring (Provide description in a separate page, if needed)	
Low-footprint: Which agent has the lowest footprint (memory and CPU)?	

[illegible]