Arion Health framework

Wei Yue

Futurewei Technologies

5/2022

Arion Health/Anomaly check framework

By: Wei Yue
wyue@futurewei.com

Key components

eBPF probes: A set of eBPF probes. Core ones are pre-installed, others could be dynamically deployed based on needs change.

Arion Health Agent(AHA): installed per host. It takes events from AHD and deploys eBPF probes to collect and analysis selective data and send triggered events over to AHD.

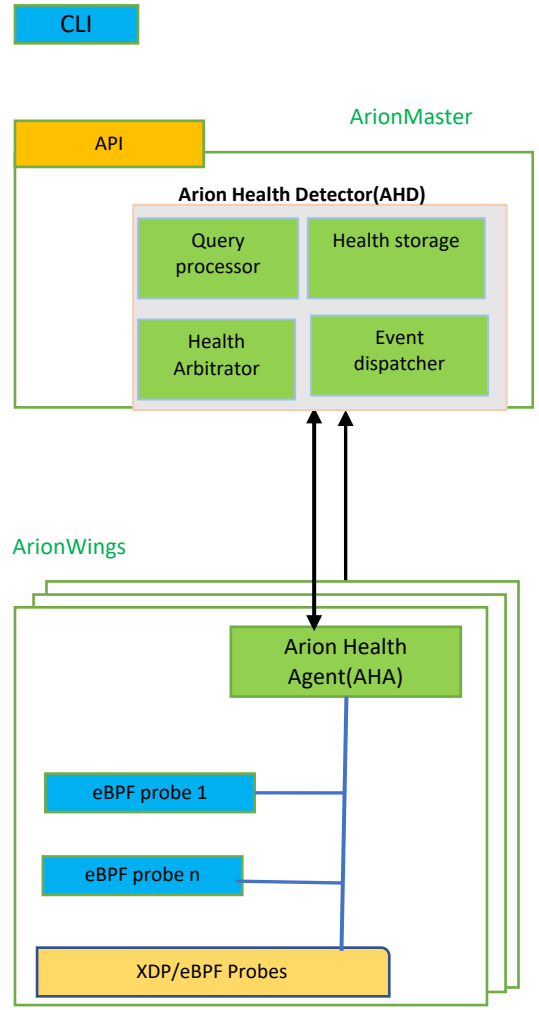
Arion Health Detector(AHD): installed per cluster. It consists of query processor; event dispatcher, event collector, health table:

- **Query processor:** analysis query to form into various events for event dispatcher; or response with health items as requested;
- **Event dispatcher:** dispatch events to AHA;
- **Health Arbitrator:** collect triggered events from AHA and stored in Health storage; analyze triggered events
- **Health storage:** store current and histogram of cluster health info.

APIs : A set of APIs for health check definition and queries for CLI to use. We can construct **DSL** for defining events.

Key goals:

1. Able to define health monitoring events and deploy them;
2. Able to install eBPF probes and collect health data and trigger events at AHA;
3. Able to collect triggered events and store in Health storage;
4. Able to query health info via CLI for other components in cluster.



Arion Health/Anomaly check framework

By: Wei Yue
wyue@futurewei.com

What are the differences?

1. Azure Anomaly Detector embed time-series anomaly detection capabilities into your apps to help users identify problems quickly, its granularity is at most at per minute level;

2. The cloud/edge application becomes increasingly sensitive to performance anomaly at micro-second granularity:
   Traditional analytic technics can't meet new demands!

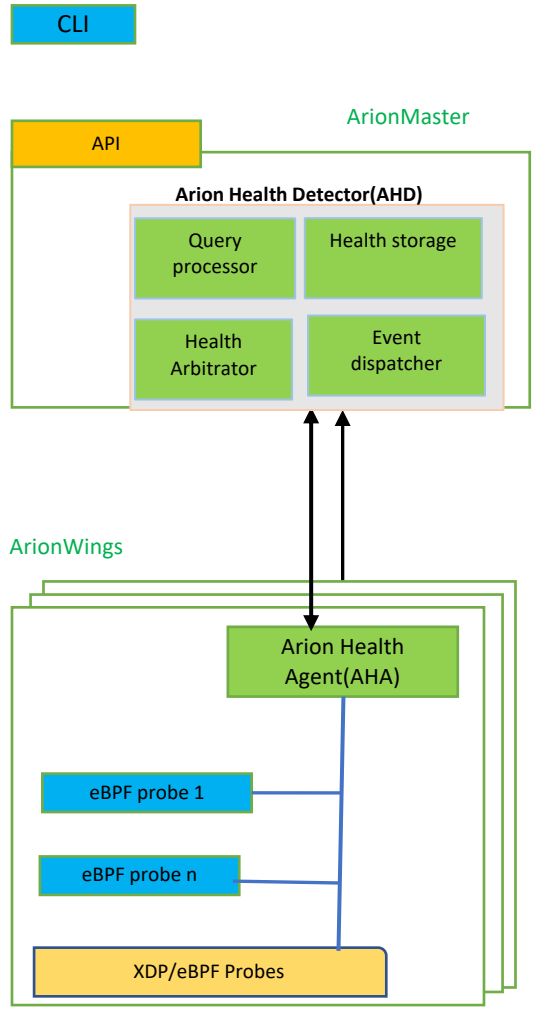
3. Arion Health framework is with finer monitoring granularity at the edge with faster response and can potentially detect otherwise undetectable symptoms:
   Towards micro-seconds anomaly detection granularity

4. flexible extendibility:
   - dynamic event creation and injection;
   - allow exploring various AI algorithms into the equation in the future.

5. Far less resource consumption.

What monitor metrics to start?

- network telemetry;
- generic node health info.



Arion Health/Anomaly check framework

By: Wei Yue
wyue@futurewei.com

Background knowledge

eBPF programs are event-driven and are run when the kernel or an application passes a certain *hook point*.

Pre-defined hooks

- system calls;
- function entry/exit;
- kernel tracepoints;
- network events;
- others.

If a ***predefined hook*** does not exist for a particular need, it is possible to create a kernel probe (*kprobe*) or user probe (*uprobe*) to attach eBPF programs almost anywhere in kernel or user applications.

Potential monitoring event examples

- Packet drops
- eBPF map access
- Protocol statistics
- Block-I/O Latencies
- File system Latency
- CPU Scheduling Latency
- ……