NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

ClusterCockpit Metric Collector

Thomas Gruber (NHR@FAU)
Holger Obermeier & Mehmet Soysal (NHR@KIT)

What is it?

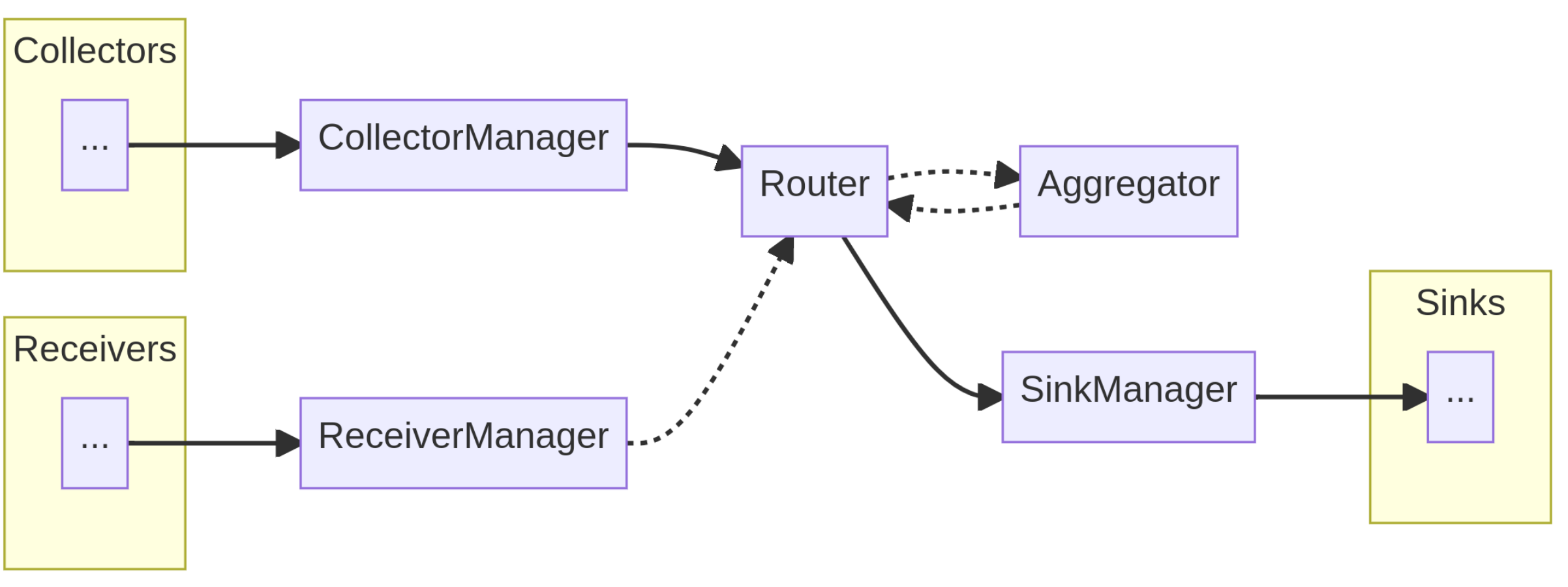
Common:

- A node agent collecting local metrics
- Sends data to various data backends (data bases, pub/sub services, …)

Specialties:

- Receive also data from remote nodes/services for processing
  → Set up tree like structures of collectors
- Process metrics while they are flowing through the system
  - Renaming metrics
  - Adding information (tags or meta information)
  - Manipulate metrics value (adjust units, derive differences, …)

Internal structure and connections



Internal structure and connections

- Internal connections are Golang channels (buffered, asynchronous)
- Same I/O format for all components
  Format derived from InfluxDB line-protocol
  - Metric name
  - Set of **`key=value`** tags
  - Set of **`key=value`** meta information (not published by sinks)
  - Metric value
  - Timestamp
- Central time ticker tells components about new measurement interval
- For all components templates exist

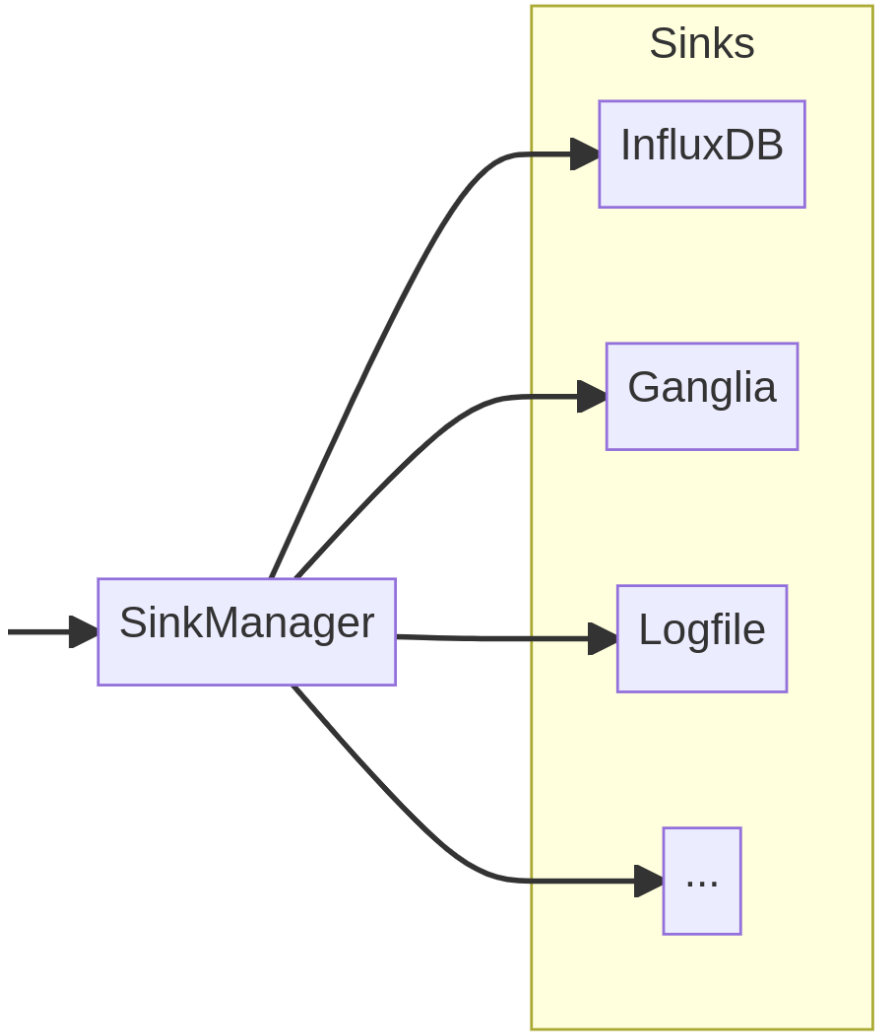
Available collectors

- Big set of already available HPC-relevant collectors
  - Shared file systems: BeeGFS, Lustre, NFS, IBM Spectrum Scale (GPFS)
  - System: Compute, Memory, NUMA, Performance Counter (LIKWID)
  - Network: InfiniBand, OmniPath, Ethernet
  - Accelerators: Nvidia GPUs (incl MiG) and AMD GPUs
  - Possibility to run own commands or read own files

- Simple implementation per collector (see template)
  - **`Init()`**: check usablility of collector & initialize data structures
  - **`Read()`**: read metrics and send them in the pipeline
  - **`Close()`**: perform some final work at shutdown

Available sinks

- In the end, the collector publishes the metrics to various sinks
  - Ganglia Monitoring System
  - InfluxDB
  - Stdout, Stderr, Logfile
    (→ Can be adjusted for own data sink)
  - Prometheus (HTTP-Server)
  - NATS (Pub/Sub)
  - HTTP POST Requests
    (used by CC Metric Store)



Available receivers

- Enables proxy configurations / tree-like structure by stacking CC Collectors

- Other receivers:
  - Crawl Prometheus clients
  - Subscribe to NATS Subjects
  - Metriken als HTTP POST Requests
  - Redfish to get IPMI-like data from remote systems

- Differentiation between collector and receiver:
  The collectors run periodically to get data from the local system, receivers accept data at all times from remote systems

Central component: Router & Aggregator

- Router: central dispatcher between collectors/receivers and sinks

- Aggregator: flexible metric manipulator
  - Renaming
  - Adding, deleting, manipulating tags and meta information
  - Adjust unit prefixes (kByte → GByte) and normalize units (kB → kByte)
  - Combine metrics to new metric
  - Uses powerful expression system:
    ```
    Name(metric) == 'mem_used' && Tag(metric, 'type') == 'accelerator'
    ```
  - *Disclaimer*: I'm currently working on it

Installation

- We provide RPM packages (Deb in preparation)
- Single binary
- No Golang runtime required
- Systemd integration
- External libraries are dynamically loaded (Nvidia, AMD, LIKWID, Ganglia)
- One JSON config per component
  - Single metrics can be deactivated
  - Devices can be excluded

Thanks for your attention

https://github.com/ClusterCockpit/cc-metric-collector