



Datenerfassung und Knotenagenten

Thomas Röhl

FEPA Workshop, 21.07.2017, Erlangen

Agenda

- Datenerfassung
 - Auf Hardwareebene
 - Auf Systemebene
 - Auf Applikationsebene
 - Von notwendigen Komponenten
- Knotenagenten
 - Eigene Skripte
 - Downloadables:
 - Diamond
 - Collectd
 - Telegraf
 - ...

Datenerfassung auf Hardwareebene

- Hardware Performance Monitoring:
 - Events auf Hardwareebene zählen
 - Außer Management kein weiterer Overhead
 - Möglichkeiten: perf_event, LIKWID, tacc_stats*, PAPI
- Peripherie
 - Metriken/Events von Peripheriegeräten (InfiniBand Adapter, GPUs, ...)
 - Je nachdem wie gut die Abfragen implementiert sind
 - Wird Kommunikation mit Peripherie benötigt?
 - Oder sind die Daten bereits im System zugreifbar?
- Ggfs. abgeleitete Metriken erzeugen (Bandbreiten, Ratios, ...)

Datenerfassung auf Systemebene

- Meistens auslesen von /proc und /sys (loadavg, stat, meminfo, cpu speed, ...)
- Kleinere Programme ausführen und Ausgaben parsen
 - Nvidia-Tool für GPUs
 - Logfile-Analyse
 - ipmitool
 - lm_sensors
 - [...]
- Einige Programme benötigen längere Abfragezeiten:
z.B. LIKWID/ping benötigt Zeit bevor Werte geliefert werden

Datenerfassung auf Applikationsebene

- Overload Bibliotheken zum Erfassen von Funktionsaufrufen
 - Tooling Bibliotheken ebenfalls nutzbar (OMPT, PMPI, ...)
- Instrumentierung in der Applikation
 - User muss Mehrnutzen dafür erkennen
 - Instrumentierung muss „sicher“ sein
- Erfassen der Systemumgebung:
 - module System, Job Properties
- Erfassen aller Shell-Kommandos
 - Kann viel Information sein, filtern notwendig
 - Ausführliche Analyse möglich (Binary-Analyse, File check, ...)
- Ausführlichere Werkzeuge wie REMORA, Vampir, Scalasca, ...

Kontenagenten

Eigene Skripte

- Oft einrichtungsspezifische Skripte
- Alle machen quasi das selbe
 - Sprache unterscheidet sich
 - Jeder parst Ausgaben anders
 - Eigene Berechnung von abgeleiteten Metriken
- Kontrolle wann und wie Metriken erfasst werden
- Kontrolle wann/wohin/wie Daten versendet werden

Knotenagenten

Downloadables (Diamond, tacc_stats, collectd, ...)

- Vielzahl von Knotenagenten existieren
- Push oder Pull Mechanismen für Metriken
- Verschiedene Anzahl von Metrik-Plugins und Ausgabemodule
- Meist multi-threaded
 - Ist ja eigentlich gut
 - Aber will man das für alle Metrik-Plugins? -> HPM misst anderes Plugin und nicht App
- ABER: Mehr Software auf den Compute-Nodes benötigt
- Hauptsächlicher Unterschied ist die Programmiersprache der Agenten
 - Plugins werden häufig lokal abgeändert (andere Metrikenamen, andere Einheiten, ...)
 - Ausgabemodule werden auch gern angepasst
 - Beispiel: Diamond InfluxDB Modul erzeugt keine Tags (Indizes), alles sind Fields