RAS Framework Prototype

Real-time Data Reduction of Monitoring Data

Swen Böhm

- Reliability of HPC Systems
- Proactive Fault Tolerance
- The Monitoring System
- Test and Evaluation
- Future Work

Outline

Reliability is the ability to perform and maintain the function in routine and unexpected circumstances.

Software reliability is not aspect of this work.

Hardware reliability is increasing, but not as fast as other hardware aspects.

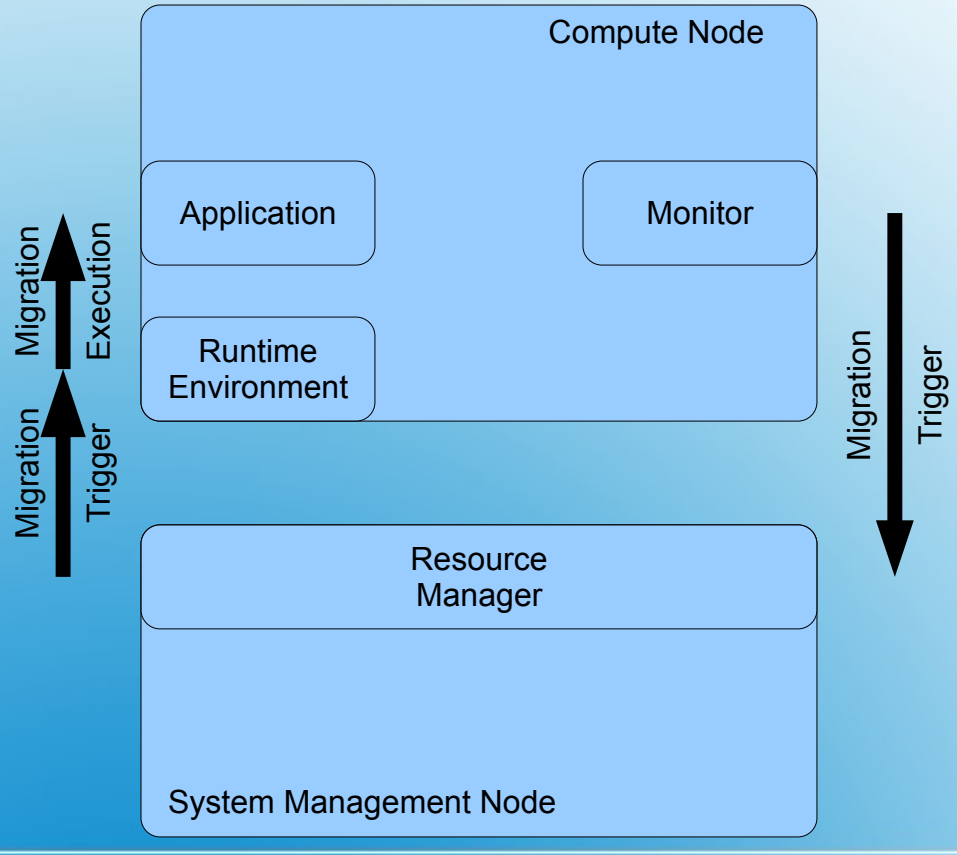
HPC systems are growing in scale. The amount of components that could fail is growing faster then the reliability of the components.

Reliability of HPC Systems

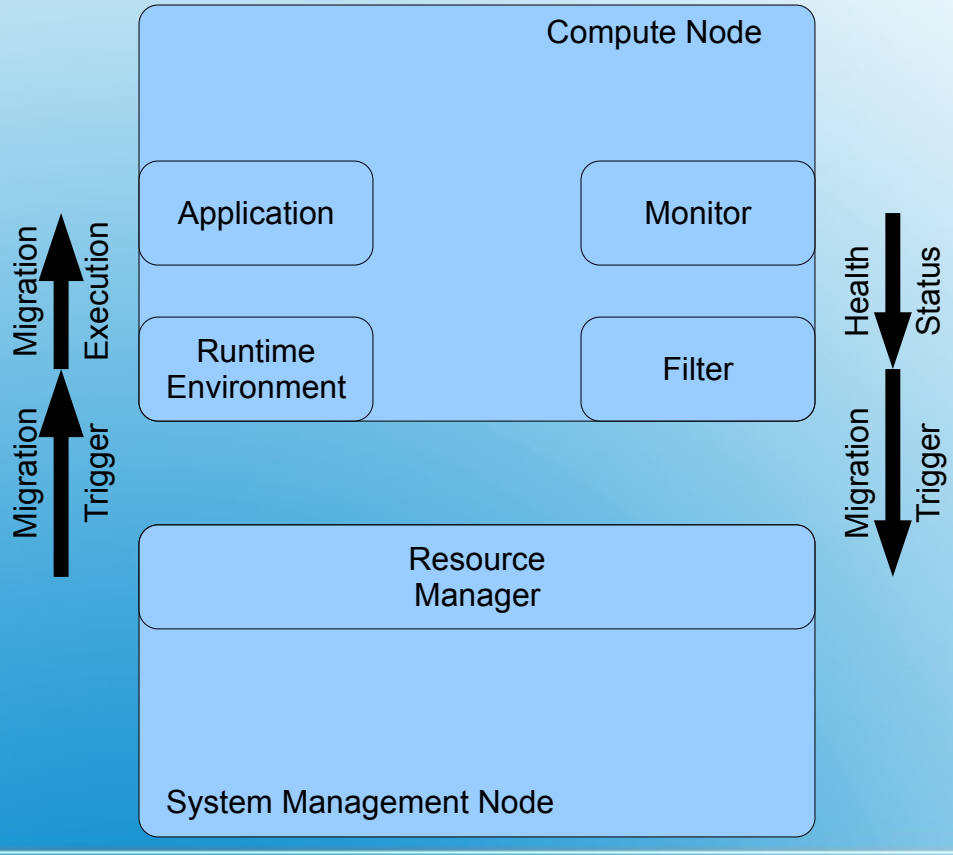
How can Fault Tolerance be achieved?

- FT trough check-pointing and restart
  - reach its limits
  - costly
- Redundant Hardware and computations
  - very cost intensive
  - does not circumvent bottlenecks
  - wasting resources?
- Proactive Fault Tolerance (PFT)

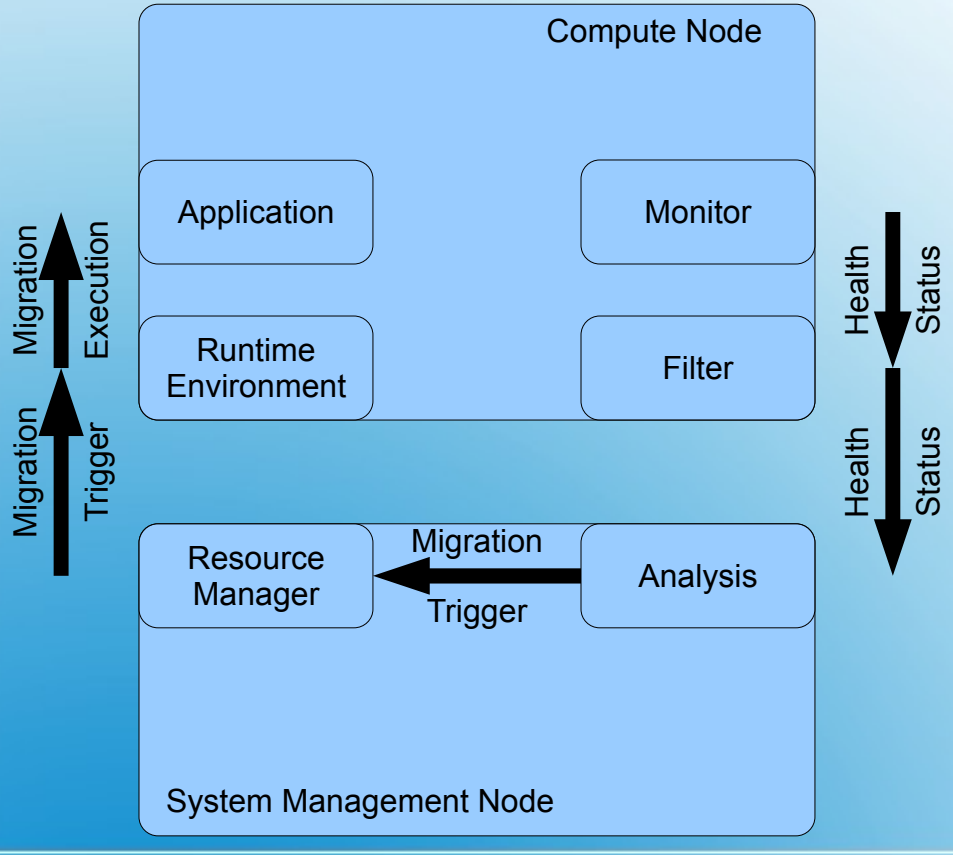
PFT – Class I


Compute Node
Application
Monitor
Runtime Environment
Migration
Execution
Migration
Trigger
Migration
Trigger
Resource Manager
System Management Node

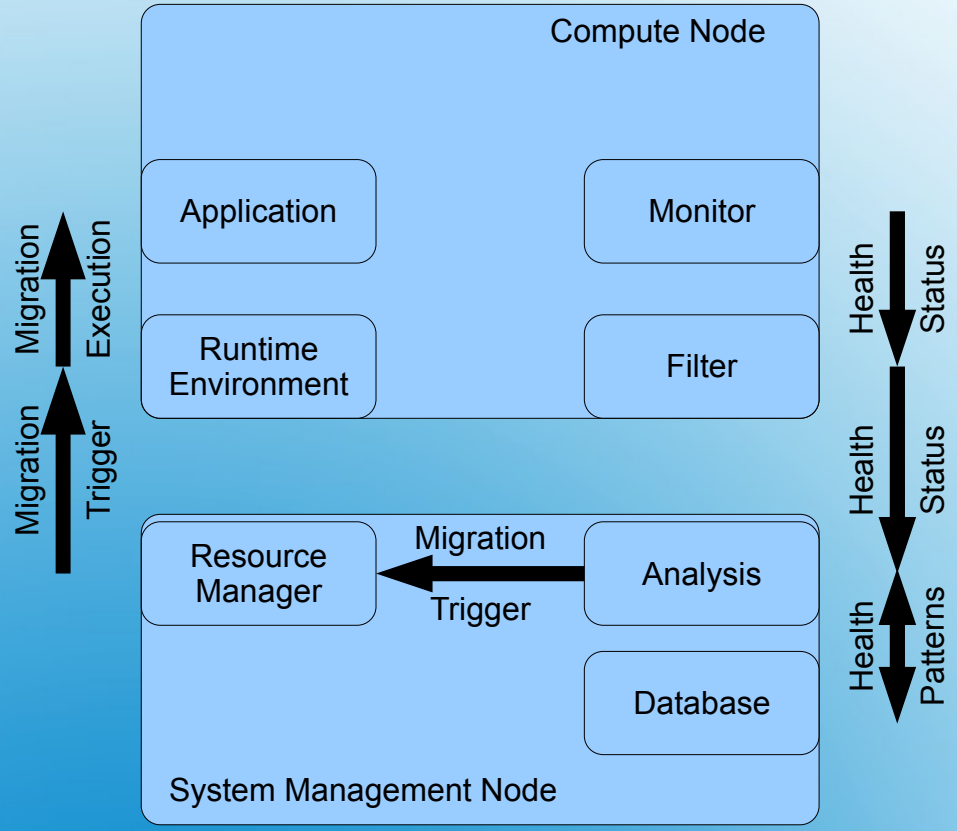
PFT – Class II


Compute Node
Application
Monitor
Runtime Environment
Filter
Resource Manager
System Management Node
Migration
Execution
Migration
Trigger
Health
Status
Migration
Trigger

PFT – Class III



PFT – Class IV


Compute Node
Application
Monitor
Runtime Environment
Filter
Migration
Execution
Migration
Trigger
Health
Status
Health
Status
Health
Patterns
Resource Manager
Migration
Trigger
Analysis
Database
System Management Node

Proactive Fault Tolerance (PFT)

- control loop
- continuously monitoring health state
- performs reliability analysis

Challenges

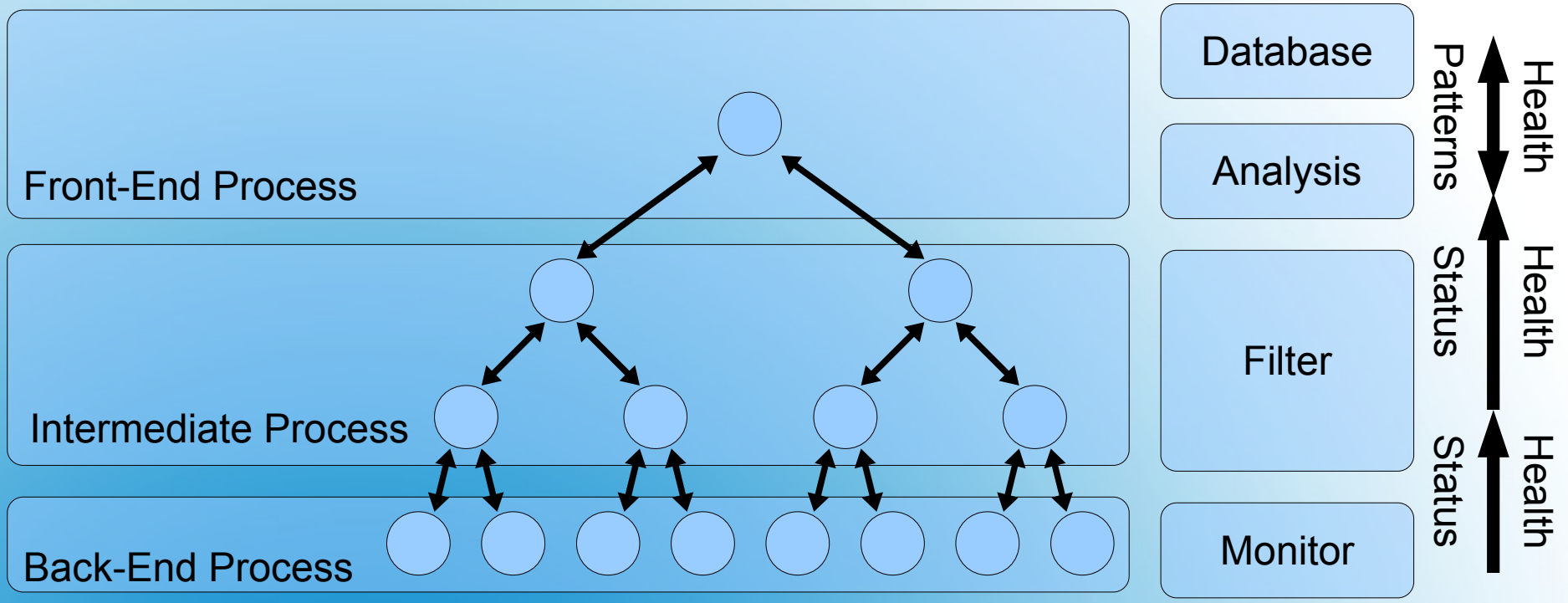
- monitoring produces vast amount of data
- storing and processing can exceed capabilities
- reaction time is crucial

Requirements for a Monitoring System for PFT

- small impact on system performance
- metrics have to be configurable
- capture metrics in small intervals
- Reduction of monitoring data
- portable and modular
- has to be fault tolerant itself

The Monitoring System

Uses an Tree Based Overlay Network



The Front-End

- Configures the Back-Ends
- Stores Monitoring Data to Database

The Back-End

- Modular Metric Capturer
- Classifies the Values

The Filter

- Merges the Packets from the Back-Ends

Test

Test Systems
- On local host
- XTORC cluster (32-nodes)

Test Scenario
- Running long term monitoring
- Kill processes to evaluate FT

Evaluation

Data rate
- Produces ~ 300 kB / h Monitoring data

FT
- Monitoring not affected by Back-End & Intermediate Child failures

The Future

Improvements for the monitoring system

- Reintegration of nodes
- Different Constellations of Nodes
- Time Adjustments

RAS Framework

- Analysis of the Monitoring data
- Pattern matcher

Thank you!

Any questions?