CMX – A GENERIC SOLUTION TO EXPOSE MONITORING METRICS IN C AND C++ APPLICATIONS

Felix Ehm, Yves Fischer, Georgia-Maria Gorgogianni, Steen Jensen, Peter Jurcso
CERN, Geneva, Switzerland

The knowledge of the internal state of processes is essential for problem diagnosis as well as for constant monitoring for pre-failure recognition. The CMX library provides monitoring capabilities for C/C++ similar to the Java Management Extensions (JMX). It allows registering and exposing runtime information as floating point numbers and character data. This can be subsequently used by diagnostics tools for checking thresholds, sending alerts and trending. CMX uses shared-memory for low latency read/update actions, which is an important requirement in real-time processes.

Integration
- CMX has no external dependencies.
- Lightweight: <100kB code, only 3100 SLOC.
- Simple and intuitive C API.
- Object-oriented C++ API available.
- Portable by POSIX conformance.

Features
- Low latency operations.
- Supported data types: numerical values and strings.
- Flexible memory model: number of metrics can be specified.
- Components: metric groups created on demand during runtime.
- Automatic timestamp for updates.

Future
- Integration of CMX into the majority of CERN’s accelerator control system components.
- Use of CMX to monitor about 2000 server processes on 1000 computers of CERN’s accelerator control system.
- Experience collected in production environment may lead to further extensions.
- Elaborate other usage domains and scenarios.
- CMX is a public project: http://cern.ch/cmx

Example Use Cases
- Current number of threads.
- Current memory consumption.
- Current number of threads.
- Communication errors.

Example: expose build-time information

Example: memory and CPU usage.

Conclusions
With the new CMX library, a software developer has a simple and intuitive API which offers a time-saving way to expose internal information on (real-time) C/C++ processes. For the first time, it is possible to inspect these programs – without using debugging tools – during their execution. CMX is fully integrated into DIAMON, and thus, allows inspecting information remotely in the same way as it is now for Java processes using one central interface.

Pre-failure recognition and detailed diagnostics, which are essential for running complex infrastructures, are now possible and the first experiences within CERN’s accelerator controls group show that it enhances the monitoring and diagnostic capabilities of C/C++ programs.

CMX - Gain control over your C/C++ processes!

Architecture

Features
- Pre-failure detection.
- Monitoring.
- Trending.
- Maintenance.
- Diagnostics.
- Debugging.

Integration
- CMX has no external dependencies.
- Lightweight: <100kB code, only 3100 SLOC.
- Simple and intuitive C API.
- Object-oriented C++ API available.
- Portable by POSIX conformance.

Example Use Cases
- Current number of threads.
- Current memory consumption.
- Current number of threads.
- Communication errors.

Example: expose build-time information.

Example: memory and CPU usage.

Future
- Integration of CMX into the majority of CERN’s accelerator control system components.
- Use of CMX to monitor about 2000 server processes on 1000 computers of CERN’s accelerator control system.
- Experience collected in production environment may lead to further extensions.
- Elaborate other usage domains and scenarios.
- CMX is a public project: http://cern.ch/cmx

Conclusions
With the new CMX library, a software developer has a simple and intuitive API which offers a time-saving way to expose internal information on (real-time) C/C++ processes. For the first time, it is possible to inspect these programs – without using debugging tools – during their execution. CMX is fully integrated into DIAMON, and thus, allows inspecting information remotely in the same way as it is now for Java processes using one central interface.

Pre-failure recognition and detailed diagnostics, which are essential for running complex infrastructures, are now possible and the first experiences within CERN’s accelerator controls group show that it enhances the monitoring and diagnostic capabilities of C/C++ programs.