CIM Based Distributed Monitoring

Viktor Mihajlovski
Adrian Schuur

Preface
This document lays out how an advanced distributed monitoring concept can be devised and implemented using CIM design principles and technology.

Summary

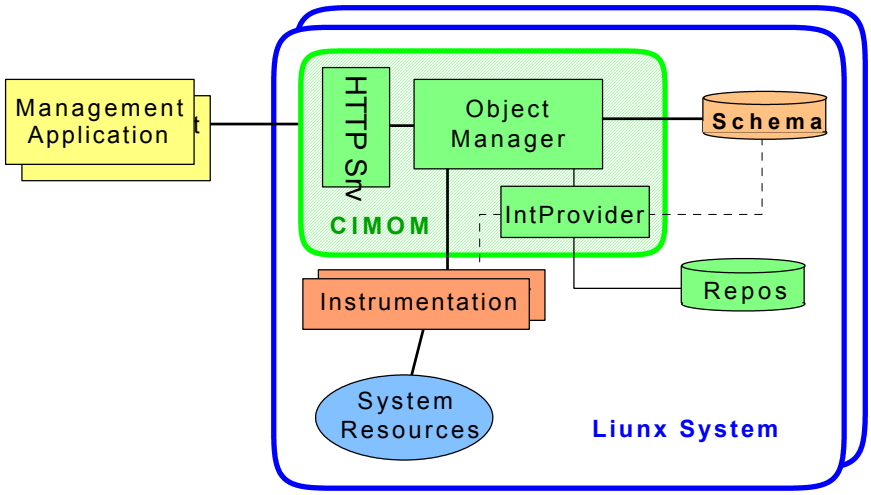
The proposal described here spans a wide variety of areas we can talk about. They go from specific instrumentation down in the kernel, up to complete management applications but also sub topics like modeling management and/or recovery policies. The management domains on the other hand could also be as diverse as simple alerting up to workload balancing or disaster recovery.

CIM Basics
This paragraph is not going into the technical details of CIM. The reader is kindly referred to the appropriate places, like the DMTF websites for technical information.

For the purpose of our discussion it will be sufficient to know about some key concepts of CIM which is the acronym for Common Information Model. At it's core it is a data modeling concept plus a set of data models suited for systems management purposes. The data models within CIM (usually referred to as CIM Schemas) provide a rather resource-centric view onto systems management entities like systems, logical and physical devices, software, configurations etc.

These schemas are not necessarily complete but a strength of CIM is it's extensibility (due to the object oriented modeling approach that is being used). That makes it possible to create extensions to the CIM Schemas (the set of all DMTF-originated CIM Schemas is called CIM Common Schema) addressing the respective aspects of systems management.

Besides the model-related specifications there are also some standards allowing to build inter operable implementations, the so called WBEM standards, which stands for Web Based Enterprise Management. They consist of the CIM data model, operation and data encoding in XML and a transport protocol based on HTTP.

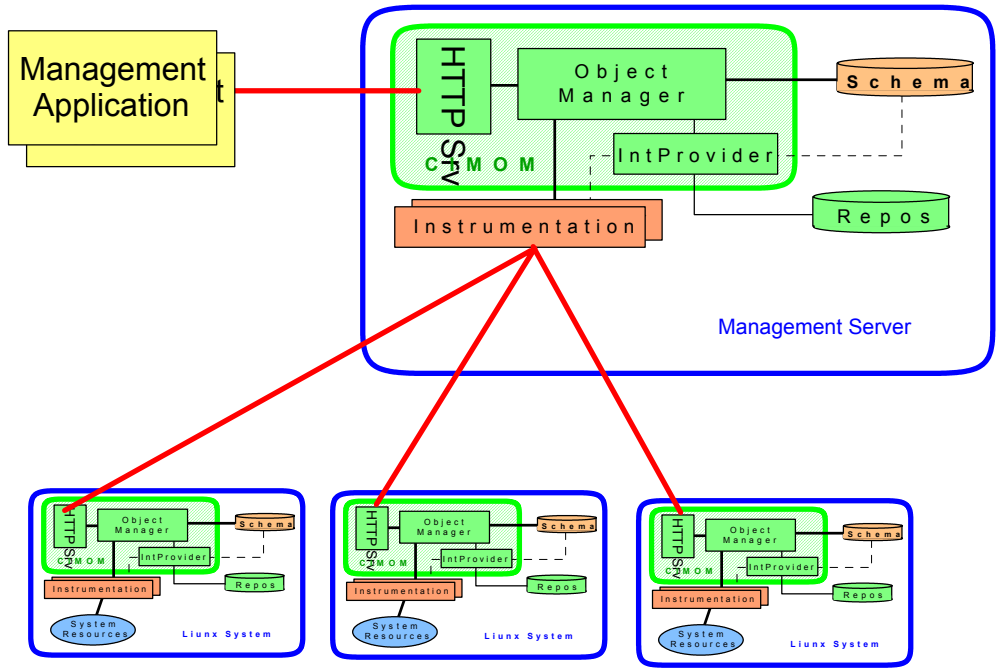


b

As has been mentioned, the CIM modeling approach is primarily resource-centric which by itself is OK as all of those resources must be manageable to a certain extent. If we however think of enterprises with hundreds or thousands of systems and much more associated systems management objects of different nature, it becomes evident that some more advanced concepts are needed to avoid systems management induced complexity and performance problems.

An approach that appears to be well-suited to tackle this problem is a multi-tiered systems management architecture. By introducing so called systems management servers that concentrate and consolidate information from systems and/or resources to be manged, it is possible to build scalable systems management infrastructures (or better: solutions). The systems beeing managed in this manner themselves can be indiviual systems or HA clusters.

An example would be the calculation of the health state of an HTTP server farm which could take into account CPU utilization, TCP/IP stack utilization, paging rate, etc. Instead of having one central application constantly monitoring and evaluating this information, a systems management server could subscribe to threshold violations of a set of systems/resources of the individual servers (CIM also supports event subscription through so called indications), compute and consolidate the health state and report only critical events to a central management application like a console, a WAP handy or a pager. Of course it is possible to implement even more sophisticated scenarios that include policy based recovery actions initiated by the systems management server.



The following paragraphs provide an in-depth discussion of the required piece parts for distributed monitoring in a CIM enabled enterprise.

Basic Monitoring Instrumentation

Although basic instrumentation (giving CIM access to the managed resources) maybe not sufficient for advanced systems management scenarios, it is still essential as it must deliver the necessary building blocks (or atoms if you wish). Providing basic instrumentation for distributed monitoring means developing a data model (the what) and writing providers (the how) for resource access.

Data Modeling

A data model for distributed monitoring has to consider various aspects from the CIM Common Schema including system state, performance data and configuration. Parts of this (at least the modeling) can be expected to be covered by the upcoming metrics schema, but there will be no out-of-the-box schema that just needs implementation.
In order to come up with a sound and usable (also meaning inter operable) model it will be necessary to acquire a good understanding of the CIM Schema and participate in the "community", i.e. WBEMsource initiative, DMTF work groups.

Instrumentation

Especially when considering system state and performance information, we will find that there are multiple places to look for this data. One approach could be to inspect the system log files for this purpose. While this is fairly easy to implement, it will surely not satisfy the requirements for high-volume, real time data acquisition. This makes it necessary to investigate whether the current structures to be found in the Linux kernel are adequate for this. In the - likely - case that

they are not, proposals for the necessary modifications have to be made and implemented at least as a prototype.

Enterprise Monitoring Servers

The basic idea behind the management servers (or, in this case: monitoring servers) is that they are CIMOMs that are not dedicated to a certain system, but maintain (consolidated) systems management information from many single systems.

While in theory there's no difference to the single-system CIMOMs (need to have data models and providers), the objects living in the systems management servers have to be much more flexible and configurable than those dealing primarily with "plain" resources. In fact the providers in questions will expose application-like behavior as they will subscribe to and control resources on the managed systems.

Again, there will be more than one approach to achieve the goal. With narrowly focused systems management objects and providers, although being easier to implement, we expect shortcomings in the ability to adapt the management/monitoring servers to different environments. Ideally the behavior of the server would be controlled by policies.
Once more, a prototype should help to find the right balance between a special purpose focused and a more flexible, configurable approach.

Monitoring Applications

Monitoring applications can be of any kind. They may be straight forward traditional (alert) consoles, but more sophisticated wireless devices like WAP, PDA or simple pagers. The application can be more complex enabling sophisticated policy based recovery scenarios. Multiple concurrent management applications are supported as well each subscribing to events they have an interest in.