A Widely Distributed De-Facto Standard Cluster Infrastructure for Application High Availability

Features and Benefits

- Architecture field proven by thousands of deployments for over five years.
- Supported by open source developers and packaged and supported by most Linux distributions.
- Secure communication supported by encryption and authentication.
- High performance on 10G Infiniband and Ethernet networks.
- Available on Linux, Solaris, BSD, Darwin.
- Works with virtualization technology.

Pure Open Source

Corosync is developed entirely in open source under Revised BSD license by a developer community that approaches problem solving by iterating designs through continuous improvement.

The Corosync software is not pay to play, pay to contribute, or pay on revenue. Roadmap influence is entirely based upon merit contributions rather then payment of association fees.

Today's software vendors operate in a highly competitive market where service high availability is a critical feature of any software product.

Enabling high availability in software services requires a high availability infrastructure. Developing high availability infrastructure is costly and complex.

The Corosync Cluster Engine open source community responded to this challenge by providing software vendors with a reusable field proven widely distributed cluster infrastructure that improves availability, minimizes time-to-market, reduces development costs, and eliminates barriers to entry.

Corosync designers use a minimalist approach to API design. This approach leads to minimal cost evaluation, rapid prototyping and agile development without significant investment in developer time.

Corosync maintains a small code footprint. While proprietary high availability cluster infrastructure may contain millions of lines of code coupled with tens to hundreds of other software package dependencies,

Corosync is compact at 60k lines of code and can be compiled with no other dependency then the C Library and Kernel. This allows system designers to design with confidence that using Corosync as a dependency won't expose them to thousands of bugs in millions of lines of code.

De-Facto Standard

Coroysnc's technology is based upon 20 years of research in distributed computing. Coupled with a large and growing community of software vendors, end users, and developers, Corosync provides the best of breed well supported high availability infrastructure available on nearly all Linux distributions by default.

Corosync is adopted by the reference open source availability manager Pacemaker, the leading open source AMQP implementation Qpid, the first and leading open source AIS implementation openais, and open source file systems GFS2 and OCFS2.

Choose confidently the cluster infrastructure selected most often by Linux distributions and open source projects. Choose Corosync.

High Availability

Availability is a mathematical model that provides a percentage that a software system remains in an operable condition. It is given by the equation:

$$A = \frac{MTTF}{MTTF + MTTR}$$

Two methods exist to improve availability. The mean time to failure (MTTF) can be increased to large values. This method generally is not used in COTS hardware.

The more common method to improving availability is to reduce the mean time to repair (MTTR) to very small values. This is most often achieved by using redundant software components running on redundant hardware.

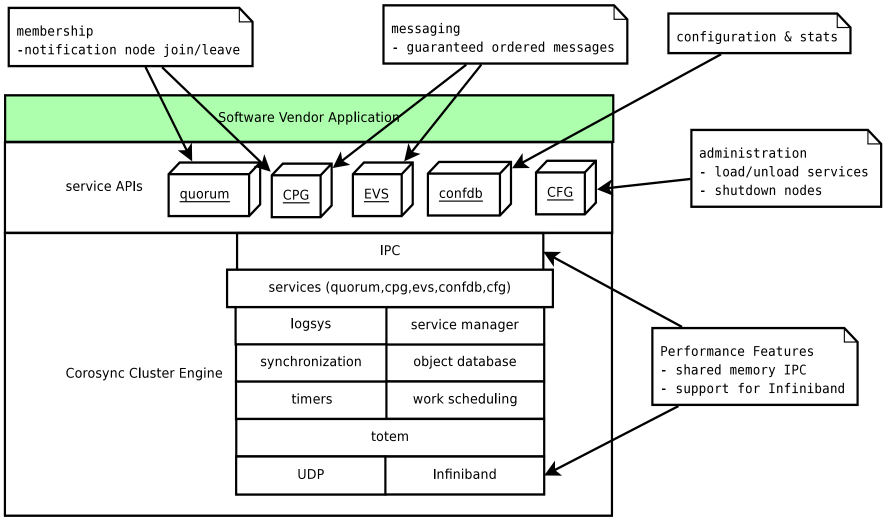
The Corosync Apporach

Corosync approaches high availability by ensuring every redundant server in the system maintains an redundant copy of information used to make decisions for the application. This approach, called a distributed state machine, is simple to use.

In a typical state machine, software designers call functions which change the state of the application. Using Corosync, software designers send messages instead of call functions. When these messages are delivered, the state machine on all nodes changes its state in an ordered and consistent fashion.

Performance

Corosync is highly tuned and designed for performance. Special consideration has been taken to minimize memory copies and context switching.



Closed Process Groups

Closed process groups provide a membership within applications. When an application joins the process group, All applications in that process group are sent a membership change with the process ID of the application and the node id of the application. This membership information can be used to make application decisions.

Once an application is joined to a process group, it may send and receive messages. A sent message is delivered to all members of the process group, which then change their internal distributed state machine.

Quorum

The quorum service engine provides information if the cluster is quorate which may be useful for some applications. It is optional.

EVS

The Extended Virtual Synchrony interface is a passthrough interface to the lower layer messaging system.

CONFDB

The confdb provides access to configuration and statistics information.

CFG

The configuration API provides mechanisms to control the operational state of Corosync.