High Availability

高可用性软件的调研

背景知识

- Heartbeat的出现：使用Heartbeat守护进程使多节点间能够协同工作，并彼此通信，获悉对方是否还处于运行状态。
- CRM，LRM：虽然集群中的节点已经可以相互通信，但是我们需要统一管理这个集群中每个节点所提供给的服务，这个时候就需要“resource agents”来根据Heartbeat传递的当前节点状态调度resource脚本。在这里，LRM负责本地节点根据CRM计算出来的步骤完成节点状态的切换。
- Pacemaker：Heartbeat项目将自己CRM的功能单独开始了一个新的项目，即Pacemaker。
- Corosync：完成集群节点间通信的另一个工具（可替代Heartbeat）。
- *Red Hat Cluster Suite*：在RED HAT环境中，他们开发了自己的高可用软件套件（*Corosysnc + OpenAIS + Cman【crm】 + RGManager【lrm】*）

1. The Messaging & Membership：实现Heartbeat通信协议（节点是否存活的通信），ClusterConsensusManager(CCM，节点成员加入与退出的通信)

2. ResourceAgent and LocalResourceManager(LRM)：定义了一系列脚本，管理节点上某些服务的开启，关闭，运行设置，并根据状态进行转换.

3. ClusterResourceManager（CRM）：包括许多子组件 ClusterInformationBase(CIB), ClusterResourceManagerDaemon (CRMD), Designated Coordinator(DC), PolicyEngine(PE), Transitioner；这些组件收集集群中节点状态信息，计算出使集群中节点一致的同某一个状态转换到另一个状态的行为，并保证消息顺序的将这些行为发送给LRMd或者其他节点的CRMd.【1】

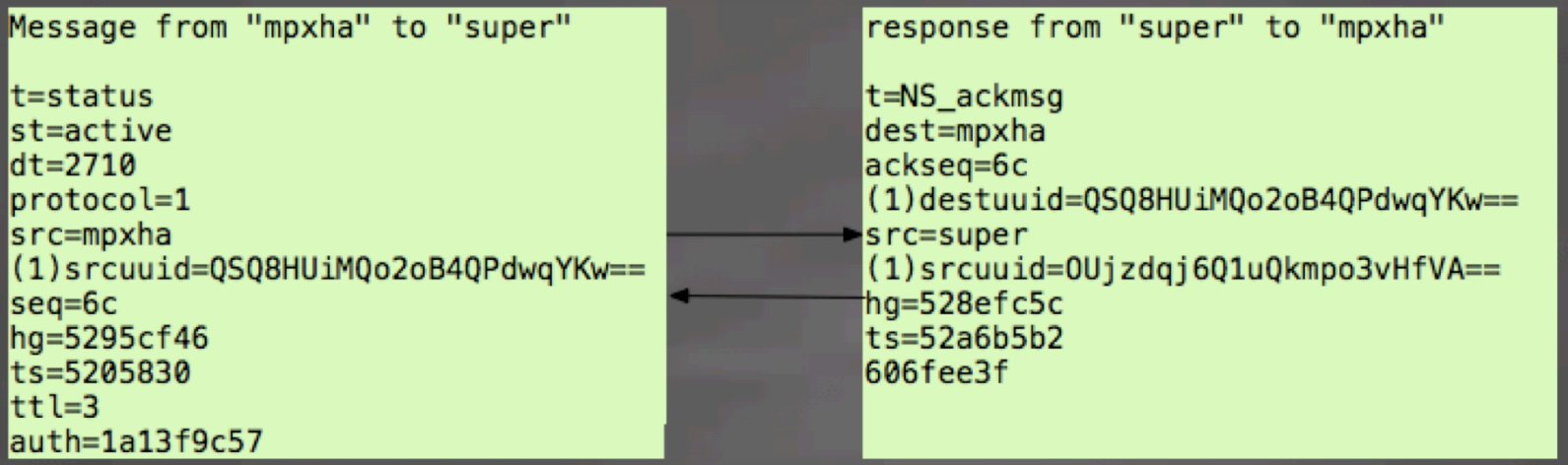
Contents

- Heartbeat
- Corosync
- Pacemaker
- reference

Heartbeat

- 用于在两个机器之间按照一定的时间间隔探测对方是否还存活的一种技术。
- Heartbeat协议：进行资源的监控和协商（对floating IP的协商）。在Heartbeat能监控的节点间通过竞选或者预先设置确定哪个节点可以提供资源（对该Floating IP请求进行响应）。
- 提供可靠的通信：保证不会因为节点的假死造成错误的资源切换（Fail-over技术，Stonith技术）；保证当真正的断裂出现时，又能及时检测（IP-FAIL，ldirector）。

Heartbeat protocol



- 上图为wireshark抓取到heartbeat通信包
- 几种Heartbeat协议【2】
  - The binary heartbeat protocol
  - The static heartbeat protocol
  - The expanding heartbeat protocol
  - The dynamic heartbeat protocol

The Binary heartbeat protocol

```
process p[0]
const tmin, tmax : integer                          {0 < tmin ≤ tmax}
var active : boolean,                                {initially true}
    rcvd : boolean,                                  {initially true}
    t : 0..tmax                                      {initially tmax}
begin
  active → if true → skip
           [] true → active := false
           fi
 [] timeout active ∧
    {a time period of at least t units has passed
    without sending a beat message} →
      if rcvd → t := tmax
       [] ¬rcvd → t := t/2
      fi;
      if t < tmin → active := false
       [] t ≥ tmin → send beat to p[1];
                     rcvd := false
      fi
 [] rcv beat from p[1] → if active → rcvd := true
                         [] ¬active → skip
                         fi
end
```

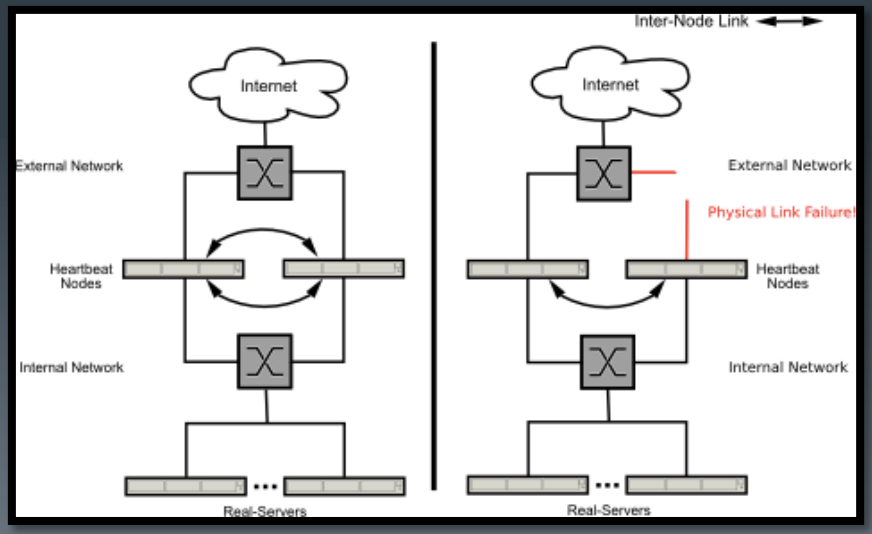
Heartbeat Processes

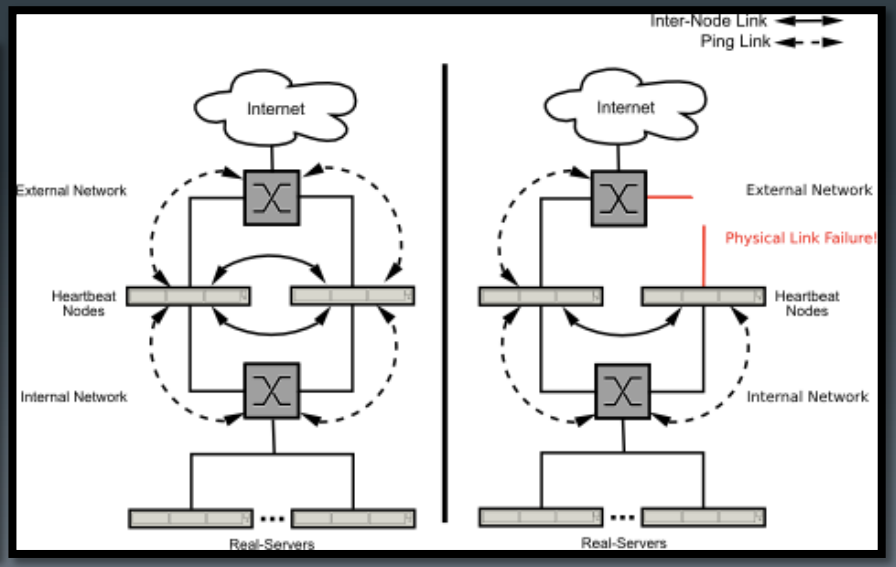
- 当Heartbeat服务起来后，系统中会有2+2*n个关于Heartbeat的进程
- Master进程，FIFO进程，已经>=一对的read/write进程
- 其他进程用于完成heartbeat的一些附加组件功能以及CRM功能

```
# ps -AHfww | grep heartbeat
root      2772  1639  0 14:27 pts/0    00:00:00          grep heartbeat
root      4175     1  0 Nov08 ?        00:37:57   heartbeat: master control process
root      4224  4175  0 Nov08 ?        00:01:13     heartbeat: FIFO reader
root      4227  4175  0 Nov08 ?        00:01:28     heartbeat: write: bcast eth2
root      4228  4175  0 Nov08 ?        00:01:29     heartbeat: read: bcast eth2
root      4229  4175  0 Nov08 ?        00:01:35     heartbeat: write: mcast bond0
root      4230  4175  0 Nov08 ?        00:01:32     heartbeat: read: mcast bond0
102       4233  4175  0 Nov08 ?        00:03:37     /usr/lib/heartbeat/ccm
102       4234  4175  0 Nov08 ?        00:15:02     /usr/lib/heartbeat/cib
root      4235  4175  0 Nov08 ?        00:17:14     /usr/lib/heartbeat/lrmd -r
root      4236  4175  0 Nov08 ?        00:02:48     /usr/lib/heartbeat/stonithd
102       4237  4175  0 Nov08 ?        00:00:54     /usr/lib/heartbeat/attrd
102       4238  4175  0 Nov08 ?        00:08:32     /usr/lib/heartbeat/crmd
102       5724  4238  0 Nov08 ?        00:04:47       /usr/lib/heartbeat/pengine
```

IPFAIL

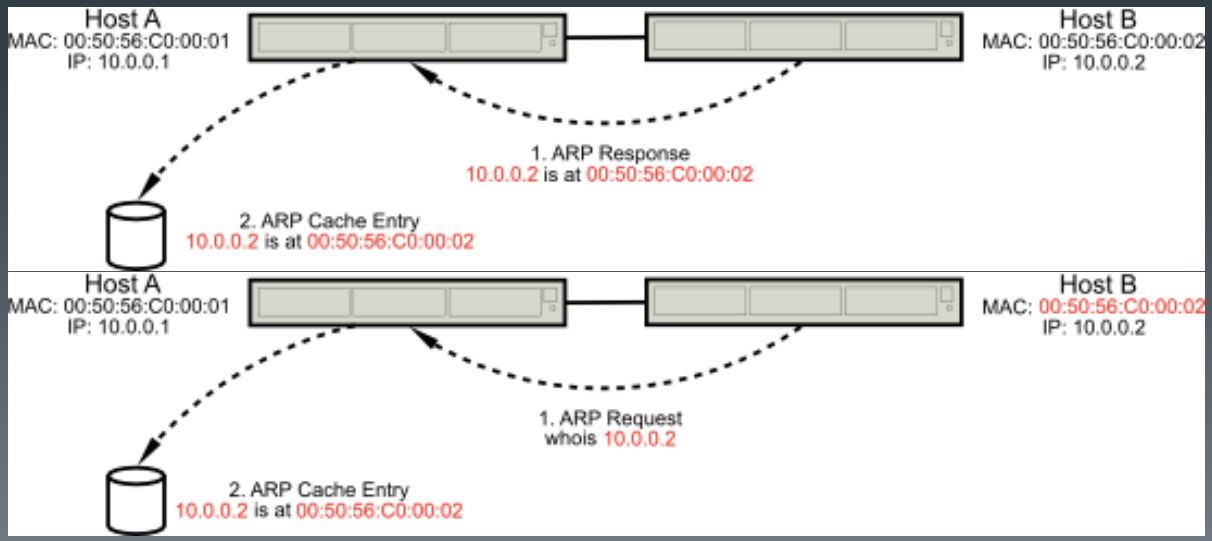
- 到目前为止，heartbeat只是用来检测急群中节点是否存活的工具，但是有时候，节点存活不代表其上的某条ip链路依然可用。
- IPFAIL PLUGIN就是完成这个功能，当ping节点不可连接时，就会启动fail-over机制。
- 如果在ha.cf中开启auto_failback功能的画，当主节点重新恢复可用，资源又回变为由主节点提供。





Heartbeat Virtual IP

- Heartbeat集群为了能够使其中的某个节点来提供某个特定的服务，往往使用virtual ip技术使交换机可以将请求转发至目前资源可用的节点，这就是Gratuitous ARP在做的事情。
- 有两种实现Gratuitious ARP的方式
  - 需要接管此虚拟IP地址的节点直接广播ARP reply，使在这个局域网内的节点刷新自己的ARP缓存。
  - 需要接管此虚拟IP地址的节点发送ARP request自己，然后再回复，从而刷新局域网内的所有节点的ARP缓存



ldirector

- Ldirector可以理解为层级更高的检测节点上的服务是否还在运行，并决定由谁来提供资源的机制。
- 例如，向HTTP server通过请求已知的URL，查看返回的数据是否含有希望的字段，称为negotiate check.
- 目前，Ldirectord提供的negotiate check有HTTP, HTTPS, FTP, IMAP, POP, SMTP, LDAP, NNTP, 以及MySQL.

STONITH

- 注意的点：使用Heartbeat的方式进行节点监控容易出现多个分区问题，这样每个分区都有可能提供资源，即集群中有多个节点提供资源，这是不允许的。（使用Stonith来解决。）

- STONITH以及Fencing的概念在这里引入，当一个节点的状态不确定的时候，fencing机制能够保证这个节点没有在提供重要的资源（服务）。

- Fencing的分类和实现：
  - 资源层面的fencing：保证对未知节点某种服务的不可用（例如，使用STONITH）
  - 节点层面的fencing：保证该节点上没有提供任何服务（例如，直接关机）

Heartbeat Config（ha.cf）【3】【4】

- 心跳方式：ucast,mcast,serial
- debugfile /var/log/HA-debug:该文件保存 heartbeat 的调试信息
- logfile /var/log/HA-log:heartbeat 的日志文件
- keepalive 2:心跳的时间间隔,默认时间单位为秒
- deadtime 30:超出该时间间隔未收到对方节点的心跳,则认为对方已经死亡。
- warntime 10:超出该时间间隔未收到对方节点的心跳,则发出警告并记录到日志中。
- initdead 120:在某些系统上,系统启动或重启之后需要经过一段时间网络才能正常工作,该选项用于解决这种情况产生的时间间隔。取值至少为 deadtime 的两倍。
- udpport 694:设置广播通信使用的端口,694 为默认使用的端口号。
- baud 19200:设置串行通信的波特率。
- serial /dev/ttyS0:选择串行通信设备,用于双机使用串口线连接的情况。如果双机使用以太网连接,则应该关闭该选项。
- bcast eth0:设置广播通信所使用的网络接口卡。
- auto_failback on:heartbeat 的两台主机分别为主节点和从节点。主节点在正常情况下占用资源并运行所有的服务,遇到故障时把资源交给从节点并由从节点运行服务。在该选项设为 on 的情况下,一旦主节点恢复运行,则自动获取资源并取代从节点,否则不取代从节点。
- ping ping-node1 ping-node2:指定ping node, 并不构成双机节点，他们仅仅用来测试网络连接。
- respawn hacluster /path/ipfail: 指定与heartbeat一同启动和关闭的进程，该进程被自动监视，遇到故障则重新启动。最常用的进程是ipfial，该进程用于检测和处理网络故障，需要配合ping语句指定Ping Node来检测网络连接。

Corosync [5]

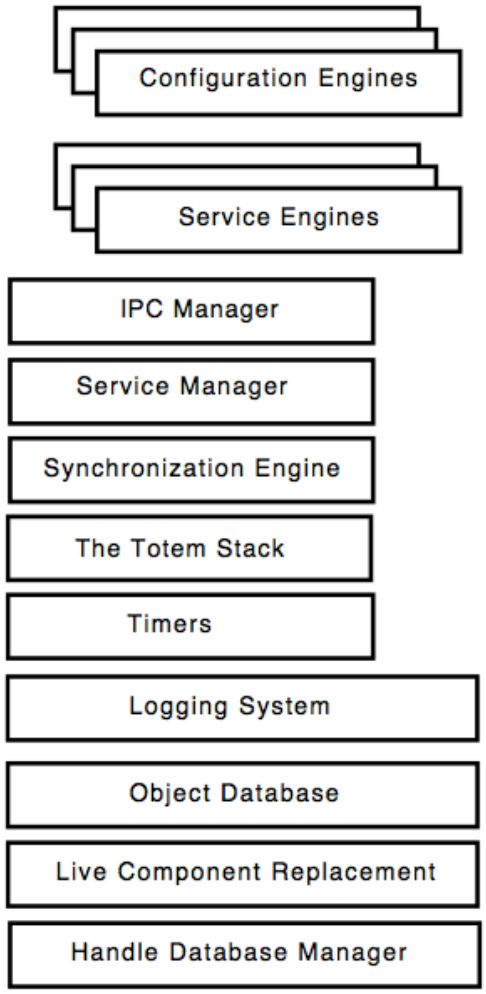


Figure 1: Corosync Cluster Engine Architecture

Corosync：另一种提供集群间节点管理和通信的机制，逐渐代替Heartbeat。

The Totem Stack：提供底层的集群节点管理和通信，保证节点能够收到顺序一致并且同步的消息指令，同时有效管理节点成员的加入退出等事件。

IPC Manager：服务IPC请求的接收和传输，将需要传递的消息通过Service Manager的路由，最终传递给适合的Service Engine.

Service Engine: 例如pacemaker, CMAN
Service Manager: 用于加载Service Engine，在所有Service Engine之间做请求路由，发送节点成员更改信息，同步多个Service Engine的行为。
Synchronization Engine：完成Service Engine之间的同步。

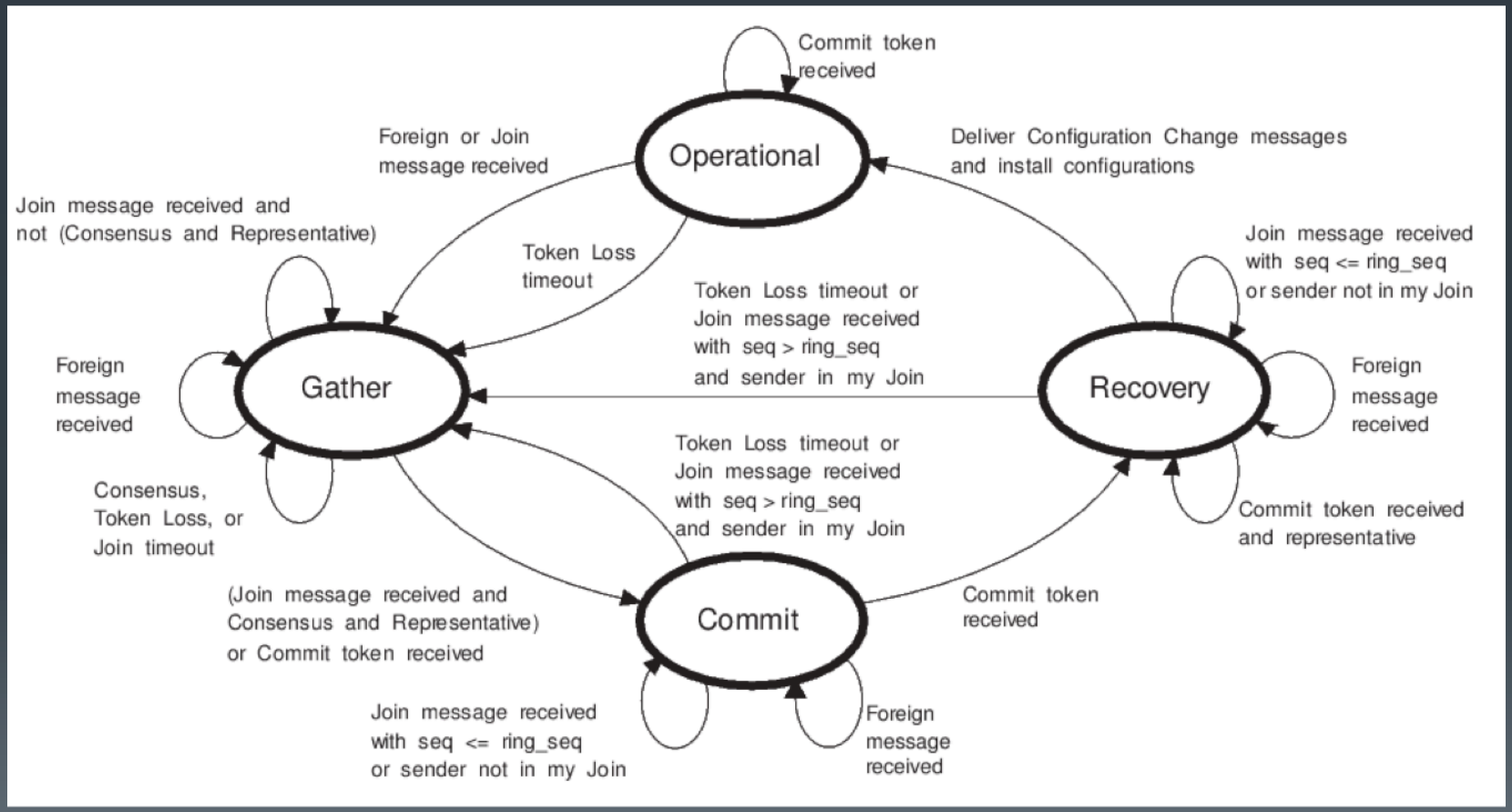
The Totem Stack based【9】

- Totem Single Ring Ordering and Membership Protocol(SRP)【6】【7】
  - 该协议用于保证分布式节点之间消息传递的一致顺序性，并管理集群成员节点的加入离开。
- Totem Redundant Ring Protocol(RRP)【8】
  - 将SRP应用到冗余的网络上，保证SRP协议本身的高可用性

SRP

- 关键技术
  - The Total Ordering Protocol: 在广播网络中保证传递到各个节点的消息具有相同的顺序，并且是安全的（在某个消息到达之前，其之前的所有消息都已经到达），但是在这个协议下，假设网络上不会出现错误，即tokem和消息均不丢失，也没有某个动作失败。
  - The Membership Protocol:检测进程失败，网络出现分区，token丢失的事件，以及新的成员节点加入退出，重新创建/修复整个RING。
  - The Recovery Protocol: 在经过Membership Protocol后，使原RING上所有节点的状态，以及准备传递给原RING节点的消息传递给新RING上的所有节点，同样保证消息是有序和安全（达到Extended Virtual Synchrony要求）。
  - The Flow Control Mechanism:使用消息Buffer，通过窗口控制，保证在广播网络下SRP依然可以具有高的吞吐量，又不会由于其中部分节点能力原因，造成消息丢失。

SRP中节点的4个状态

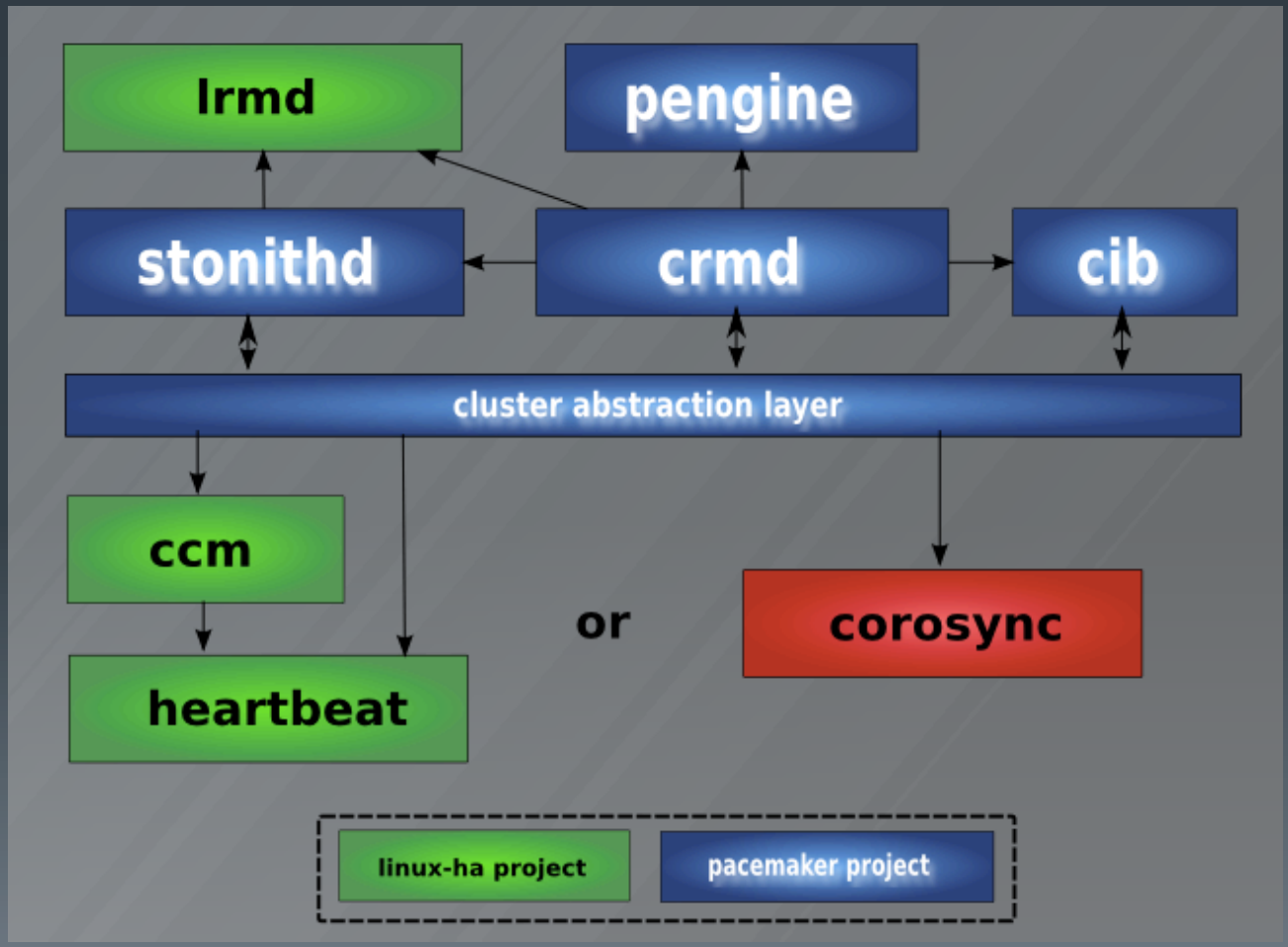


RRP

- 三种网络副本方式
  - Active Replication：所有的消息和token会同时在每个网络上传输。也就是说每个节点会收到来自于多个网络的某个消息的多个副本，这时，为了保证这个节点的应用实际只收到该消息一次，就需要使用Filter机制。
  - Passive Replication: 消息只在冗余网络的其中一个网络中传递。这样，消息的最大吞吐量就可以达到多个网络带宽的总和。
  - Active-passive Replication: 所有消息和Token只在K个网络上传递（1 < K < N）

Pacemaker【10】【11】

Pacemaker最早在Heartbeat的2.0.0版本中（2005.7）被独立出来



Pacemaker

- ClusterInfomationBase,CIB: 用于存储所有集群配置的定义信息，包括节点，资源，状态间的转换，并同步到所有节点上。

```
cib crm(live) # cib new drbd
INFO: drbd shadow CIB created
crm(drbd) #
```

```
crm(drbd) # configure primitive WebData ocf:linbit:drbd params drbd_resource=wwwdata \
    op monitor interval=60s
crm(drbd) # configure ms WebDataClone WebData meta master-max=1 master-node-max=1 \
    clone-max=2 clone-node-max=1 notify=true
```

- LocalResourceManagerDaemon,LRMd：上图中ocf:linbit:drbd就是Local Resource 脚本，由LRMd调用。【12】【13】

```
# crm ra list ocf pacemaker
ClusterMon      Dummy           HealthCPU       HealthSMART     Stateful        SysInfo
SystemHealth    controld        o2cb            ping            pingd
# crm ra list ocf heartbeat
AoEtarget               AudibleAlarm            CTDB                    ClusterMon
Delay                   Dummy                   EvmsSCC                 Evmsd
Filesystem              ICP                     IPaddr                  IPaddr2
IPsrcaddr               IPv6addr                LVM                     LinuxSCSI
```

Pacemaker

- PolicyEngine,PEngine:计算模块，计算出从节点从当前状态转换到下一个状态所要完成的一系列动作以及关联关系的转化图。
- TransitionerEngine,Tengine:将PolicyEngine计算出来的转化图递交给LRMd。
- ClusterResourceManagerDaemon,CRMd:在集群上的所有节点上运行，PE,TE,LRMd的消息代理，所有节点中的其中一个作为主CRMd进程，其他则在主CRMd失败的时候，通过竞选成为主CRMd。

Reference

- [1] A new Cluster Resource Manager for heartbeat
- [2] Accelerated Heartbeat Protocols
- [3] ha.cf配置例子1
- [4] Linux manual ha.cf
- [5] The Corosync Cluster Engine
- [6] The Totem Single Ring Ordering and Membership Protocol
- [7] totem SRP ppt
- [8] The Totem Redundant Ring Protocol
- [9] Totem SRP & RRP ppt(中文版)
- [10] Clusters from Scratch
- [11] Pacemaker Configuration Explained
- [12] The OCF Resource Agent Developer's Guide
- [13] Resource agent manual pages

Thanks

http://xuechendi.github.io/blog/2013/12/09/heartbeat-and-drbd-internal/

xuechendi@gmail.com