

# WhyFLOSS Conference

Madrid, Julio 2007

Organiza: **neuroWork™**

Colaboran:  **CINDETEC innova**  **UNED**  **Sun microsystems**

Media sponsors: **LINUX+** **LINUX**  **Paceta Tecnológica**  
**Todo LINUX** **MUNDO LINUX**

## Alta disponibilidad con GNU/Linux

Jesus Espino, Presidente del Grupo de Usuarios de Linux de la Universidad Carlos III de Madrid y Vocal de HispaLinux

<http://www.whyfloss.com/es/conference/madrid07/>



# Alta Disponibilidad con Linux HA

Jesús Espino García  
jespinog@gmail.com

**WhyFLOSS**  
**Conference**

Madrid, Julio 2007

# ¿Qué clustering?

- Conjunto de maquinas que trabajan en equipo para conseguir un objetivo.
- Los casos mas habituales son:
  - Cluster computacional.
  - Alta Disponibilidad.

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:
  - Caro

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundancia interna, detección de errores).

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundancia interna, detección de errores).
  - Pérdidas de servicio muy breves, entre decimas de segundo y unos pocos segundos.

# Alta Disponibilidad VS Toleraciona a Fallos

- Toleracia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundacia interna, deteccion de errores).
  - Perdidas de servicio muy breves, entre decimas de segundo y unos pocos segundos.
- Alta disponibilidad:

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundancia interna, detección de errores).
  - Pérdidas de servicio muy breves, entre decimas de segundo y unos pocos segundos.
- Alta disponibilidad:
  - Barato

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundancia interna, detección de errores).
  - Pérdidas de servicio muy breves, entre decimas de segundo y unos pocos segundos.
- Alta disponibilidad:
  - Barato
  - Redundancia de hardware

# Alta Disponibilidad VS Toleraciona a Fallos

- Toleracia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundacia interna, deteccion de errores).
  - Perdidas de servicio muy breves, entre decimas de segundo y unos pocos segundos.
- Alta disponibilidad:
  - Barato
  - Redundancia de hardware
  - Perdida de servicios entre algunos segundos hasta unos pocos minutos

# Alta Disponibilidad VS Tolerancia a Fallos

- Tolerancia a fallos:
  - Caro
  - Hardware tolerante a fallos (redundancia interna, detección de errores).
  - Pérdidas de servicio muy breves, entre decimas de segundo y unos pocos segundos.
- Alta disponibilidad:
  - Barato
  - Redundancia de hardware
  - Pérdida de servicios entre algunos segundos hasta unos pocos minutos
  - Hace uso en subsistemas tolerantes a fallos.

# ¿Qué es Linux HA?

- Un sistema de clustering de Alta disponibilidad
- Libre
- Maduro: En el mercado desde 1998
- Extendido: Mas de 30.000 cluster en produccion.
- Una comunidad de desarrollo abierta liderada por IBM y Novell.
- Distribuido con la mayoria de las distribuciones.
- Sin requerimientos de hardware especial.
- Independiente del kernel (todo en espacio de usuario).

# ¿Qué nos permite hacer?

- Elaborar sistemas resistentes a un unico punto de fallo.
- Los tiempos de caida son muy cortos
- Aproximadamente añade un "9.<sup>a</sup> tu disponibilidad.
- Hasta 16 nodos.
- Recuperacion en caso de fallo de nodo o de servicio.
- Recuperacion en caso de perdida de conectividad.
- Clusters activo-pasivo y activo-activo
- Monitorizacion de recursos
- Interfaz grafico de configuracion y monitorizacion.

# ¿Qué no puede hacer?

- Dar el 100 % de disponibilidad (nadie puede)

# Algunos numeros

- 99.9999 % -¿ 30sec
- 99.999 % -¿ 5min
- 99.99 % -¿ 52min
- 99.9 % -¿ 9h
- 99 % -¿ 3.5dias

- Nodo: Maquina (real o virtual) que forma parte del cluster.
- Recurso: Algo que manejar (Servicio, IP, Unidad de disco...)
- Agente de recurso: Un (init) script que controla un recurso.
- DC: Designated Coordinator, el nodo maestro del cluster.
- Heartbeat: Pulso de comunicacion que indica a un nodo el estado de los demas.
- Failover: Toma de control por parte de un nodo de los servicios de otro.
- Split Brain: Falla la comunicacion "Heartbeat" entre dos grupos diferentes de nodos
- Fencing
- Quorum
- SPOFs (Single Point of Failure)

- Division del cluster por culpa de fallos
- Normalmente de comunicacion del Heartbeat
- Cada nodo crea ser el nodo activo.
- OJO!! Peligro de corrupcion de datos.

- Es un intento de evitar la mayor parte de los fallos de split brain
- Trata de asegurarse que solo uno de las divisiones este activa.
- Uno de los sistemas mas comunes de quorum es el voto. solo la division con mas nodos puede ejecutar el cluster.
  - Esto no funciona muy bien para 2 nodos.
- Otro sistema de quorum es el acceso a un recurso exclusivo, el que llegue despues, desiste y cede los servicios.

- Excluye a los nodos que esten dando fallos para que no accedan a los recursos del cluster.
- Linux HA soporta:
  - STONITH (Shoot The Other Node In The Head) para node Fencing
  - Sistemas Self-Fencing (Sistemas que aseguran acceso exclusivo a un recurso (ServerRAID de IBM)) para resource fencing

- Elemento del sistema que en caso de fallo puede dejar todo el sistema inutilizado.
- Un buen diseño elimina todos los SPOFs.
- Existe un limite, normalmente impuesto por el coste, en la eliminacion de SPOFs.
- Mucho cuidado con los SPOFs no evidentes:
  - Unidades raid con una sola controladora.
  - Enlaces redundantes usando el mismo cable.

- Importante las tres R de la alta disponibilidad:

- Importante las tres R de la alta disponibilidad:
  - Redundancia

- Importante las tres R de la alta disponibilidad:
  - Redundancia
  - Redundancia

- Importante las tres R de la alta disponibilidad:
  - Redundancia
  - Redundancia
  - Redundancia

- Importante las tres R de la alta disponibilidad:
  - Redundancia
  - Redundancia
  - Redundancia
- La mayoría de los SPOFs se eliminan con redundancia.

- Importante las tres R de la alta disponibilidad:
  - Redundancia
  - Redundancia
  - Redundancia
- La mayoría de los SPOFs se eliminan con redundancia.
- Siempre existen algún punto de fallo simple, hay que saber donde parar.

# Compartiendo datos: Mediante unidad compartida

- El modo normal.
- Eleva el precio de cluster.
- Da muy buen rendimiento.
- Solucion de bajo coste (AoE).

# Compartiendo datos: Sin datos que compartir

- Es raro, pero existen casos.
- Firewalls
- Balanceadores de carga.
- Servidores Proxy.
- Servidores Web estaticos (replicacion inicial de los contenidos).

# Compartiendo datos: Replicación de los datos

- El modo barato.
- No necesitas unidad compartida.
- El rendimiento es menor.
- Se basa en los sistemas de replicación de los propios servicios y en servicios de replicación de ficheros (DRBD)
- Una relación calidad/precio excelente

# ¿Que son?

Son extensiones de Joomla que nos permiten mostrar informacion en diferentes posiciones de la web.

# Sistemas de ficheros de Cluster

- Preparados para accesos concurrentes desde varios nodos.
- Pensados para mantener la consistencia de los datos.
- Los sistemas de ficheros para cluster libres mas conocidos son:
  - GFS: Global File System (Comprado por RedHat y licenciado como GPL).
  - OCFS2: Oracle Cluster File System (Desarrollado por Oracle, GPL).