



# Tolerancia a fallos en Sistemas de Cómputo de Altas Prestaciones

Dolores Rexachs

High Performance Computing for Efficient  
Applications and Simulation (HPC4EAS)

The logo for CAPAP-H, with the letters 'CAPAP-H' in a white, stylized font. The 'A's are replaced by triangles.

"VIII Seminario de Invierno Red CAPAP-H.  
30-31 de Enero 2017, Elche"

Red de Computación de Altas Prestaciones sobre Arquitecturas Paralelas Heterogéneas

# Disponibilidad de los recursos de CAP

## Fiabilidad y disponibilidad en computadores paralelos

### Tolerancia a Fallos permanentes de nodos: Arquitectura RADIC

RADIC configurable para dominios específicos de aplicaciones (clúster multicore)

RADIC un middleware del sistema transparente a las aplicaciones

RADIC: herramientas de soporte al usuario para gestionar la tolerancia a fallos

Middleware configurable (Integrado en la librería de paso de mensajes)

Entorno virtual tolerante a fallos (API en la capa de sockets)

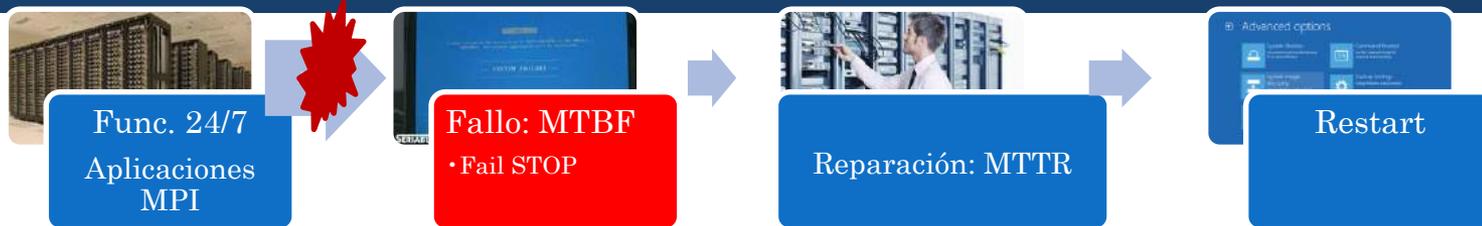
Soporte para crear un sistemas automático de tolerancia a fallos

### Fiabilidad frente a Fallos transitorios

Análisis de Robustez

Evitar la corrupción silenciosa de datos en clúster multicore

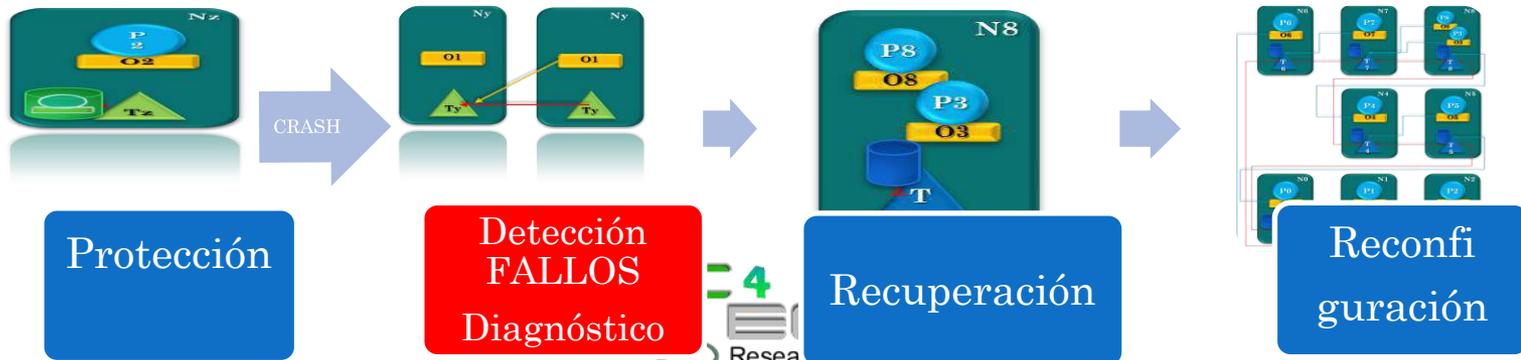
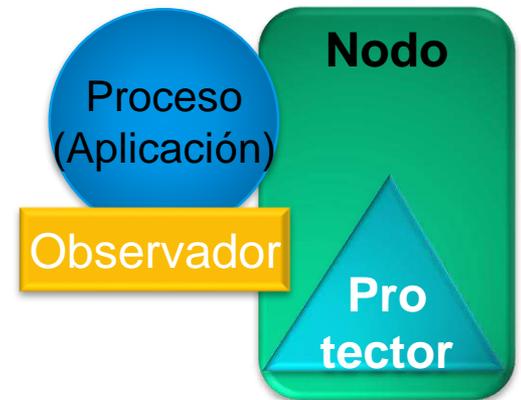
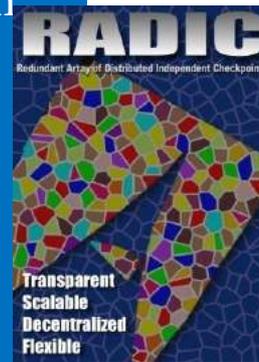
# Tolerancia a fallos: RADIC



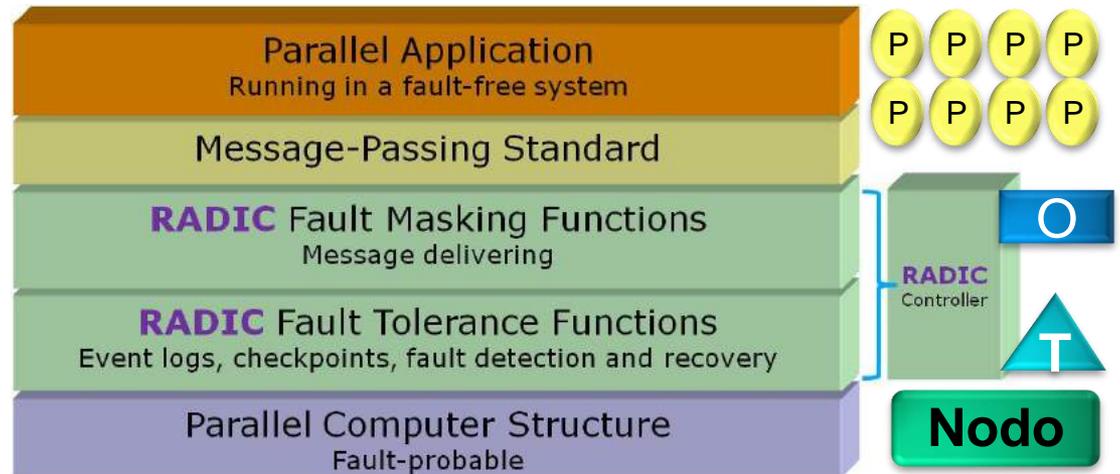
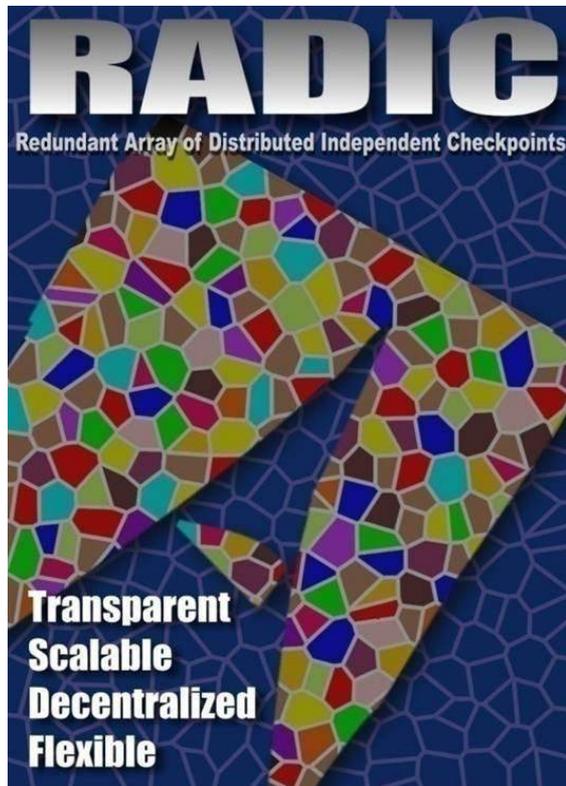
## Propuesta

Diseñar una arquitectura de tolerancia a fallos para sistemas de cómputo paralelos basados en paso de mensajes que sea

- Controlador distribuido
- Decisiones no centralizadas
- Protección y recuperación automática
- Flexible
- Escalable



# RADIC



No requiere recursos dedicados (Redundancia intrínseca de componentes)

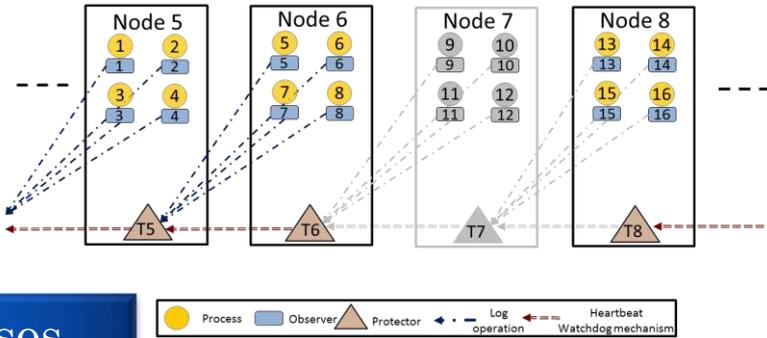
Redundancia temporal  
Rollback-recovery



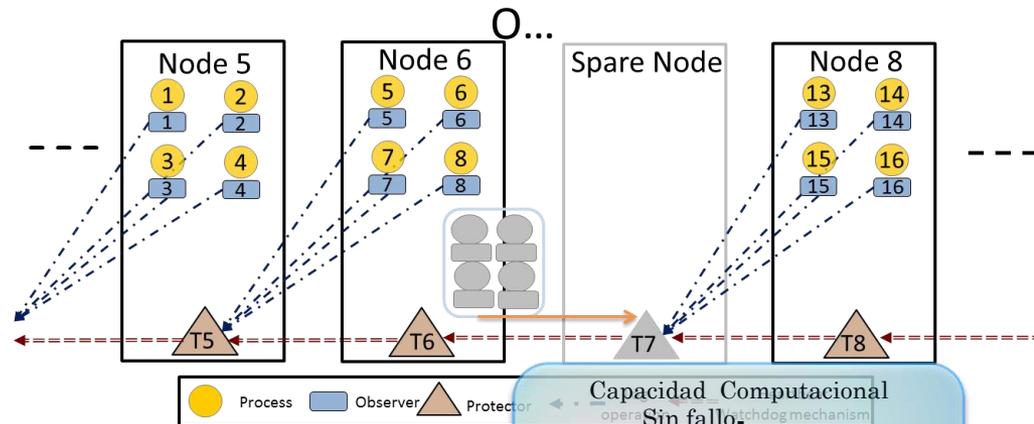
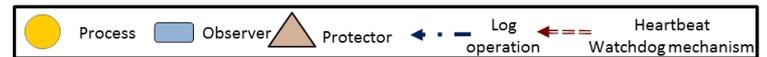
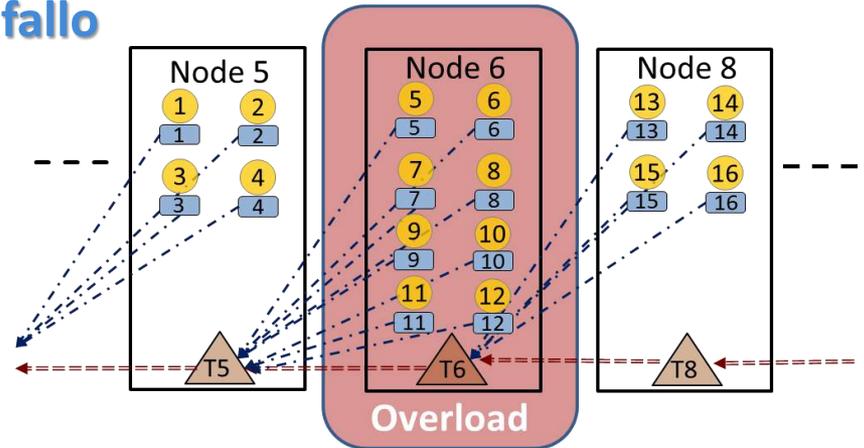
Aumentar la disponibilidad y Reducir el coste de mantenimiento

# RADIC

## Fallo de un nodo en un Sistema con tolerancia a fallos basado en RADIC



## Recuperación y enmascaramiento del fallo



Capacidad Computacional Sin fallo

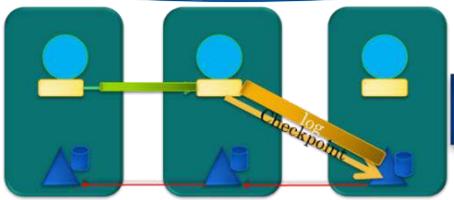
$$\sum_{i=1}^n C_i$$

después fallo

$$\sum_{i=1}^n C_i$$

### Recursos

Almacenamiento Redundancia (Checkpoint + Log)



### Comunicación

Capacidad Computacional Sin fallo

$$\sum_{i=1}^n C_i$$

Ejecución:  $T_i, O_i$

# Selección y configuración de la estrategia de tolerancia a fallos

Configuración

Req. usuarios

Aplicación

Entorno de ejecución

Análisis de requerimientos

Configuración de la tolerancia a fallos

Recursos

Protocolos: Rollback Recovery

Intervalo de checkpoint para no coordinados

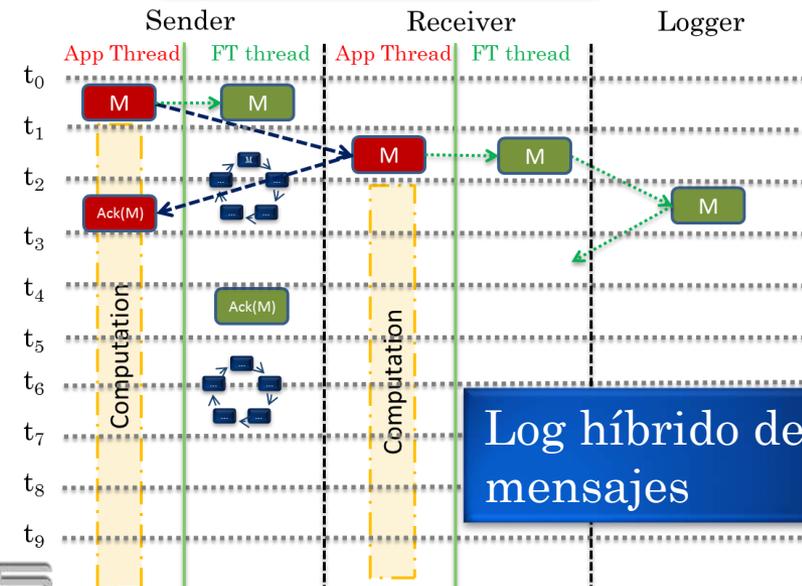
Fialho

$$\frac{\sqrt{\phi t_c (2\alpha - t_c - 2\Delta_{tr})}}{\phi} - t_c$$

Factor de interdependencia entre procesos

$$\phi = \frac{P(n)}{N}$$

Log de mensajes



Synchronous sender logging with asynchronous receiver logging

# Efectos de los fallos transitorios

## DUE

(Detected Unrecoverable Error)

Excepción de instrucción inválida

Error de paridad

Violación en el acceso a la memoria

Puede causar una condición de “fail-stop”

## SDC

(Silent Data Corruption)

Cambio en una instrucción

Cambio en un dato

Cambio en una dirección

Cambiar el resultado de una operación

**¡Más peligroso!**

Puede generar resultados impredecibles

# Tolerancia a fallos transitorios en clúster multicore

- **Fiabilidad**
  - Evitar la propagación de la corrupción de datos en clúster multicore
- **Detectar fallos transitorios en clúster multicore**
  - **SMCV: a Methodology for Detecting Transient Faults in Multicore Clusters**
- **Proteger**
  - Adaptar la arquitectura RADIC para tolerar este tipo de fallos



# Tolerancia a fallos en Sistemas de Cómputo de Altas Prestaciones

Dolores Rexachs

High Performance Computing for Efficient  
Applications and Simulation (HPC4EAS)

<http://grupsderecerca.uab.cat/hpc4eas/>