



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA



## Especialización en Cómputo de Altas Prestaciones – Modalidad a distancia

### Tolerancia a Fallos en Sistemas de Cómputo de Altas Prestaciones

Año 2021

Duración: 70 hs. Totales.

Cantidad de horas presenciales/VC: 25 hs.

Cantidad de horas de actividades en línea y de trabajo final: 45 hs.

#### OBJETIVOS GENERALES:

En Tolerancia a Fallos: el término FIT (fallo en el tiempo) se define como una tasa de fallos de 1 por mil millones de horas. Un chip que tiene una tasa de fallos de 1000 FIT es equivalente a tener un MTBF de 1 millón de horas y falla una vez cada 114 años, pero un sistema con miles de chips similares fallará cada pocos minutos.

Por esto se espera que los futuros sistemas Exascale exhiban tasas de fallos mucho más altas que los sistemas actuales, por razones relacionadas tanto con el hardware como con el software.

Los objetivos de este curso se centran en tres temas fundamentales de los sistemas Exascale:

- La ejecución eficiente
- La escalabilidad
- La disponibilidad

A partir de estos requisitos principales que deben considerarse en el diseño y uso de estos sistemas a gran escala se propone el aprendizaje de métodos y tecnologías para:

1. La ejecución eficiente de las aplicaciones que proporciona un uso justo de los recursos computacionales, minimizando la energía gastada y preservando el medio ambiente.
2. La función de escalabilidad permite que el sistema crezca proporcionalmente cuando aumenta la demanda del servicio.
3. La disponibilidad significa que el sistema continúa brindando sus servicios a pesar de fallos de hardware y software. El objetivo en HPC a gran escala es lograr disponibilidad y escalabilidad, mientras se mantiene bajo estrictas restricciones de rendimiento (por ejemplo, tiempo de procesamiento) y métricas de costos (por ejemplo, consumo de energía).



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA



## COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA

C.1- Analizar problemas del mundo real que por su complejidad y/o volumen de datos requieran cómputo paralelo y diseñar soluciones desde el punto de vista del hardware necesario, lo que requiere un conocimiento de las arquitecturas paralelas actuales.

C.4- Tener capacidad de configurar arquitecturas y desarrollar programación en la nube (Cloud Computing).

C.6- Conocer y poder aplicar técnicas de detección y corrección de fallas en sistemas paralelos.

### CONTENIDOS MINIMOS:

- Conceptos de Computación Exascale y la importancia del tratamiento de fallas al crecer las arquitecturas de cómputo de altas prestaciones.
- Análisis de diferentes tipos de fallas en sistemas de cómputo de altas prestaciones.
- Mecanismos para la detección y corrección de fallas, en la capa de aplicación y en la capa de sistema operativo.
- Herramientas específicas para monitorizar una arquitectura paralela y lograr la recuperación automática de fallas.
- Inyección de fallos y estudio de la respuesta de sistemas con diferente redundancia para soportar los mismos.

## PROGRAMA

### Introducción

¿Por qué Exascale computing? El desafío Exascale. Computación de altas prestaciones (HPC). La ejecución eficiente, la escalabilidad y la disponibilidad.

### Análisis del rendimiento de las aplicaciones.

#### 1. Evaluación de las prestaciones:

- Modelos, métricas y herramientas para la evaluación de prestaciones.
- Benchmarks.
- Predicción de prestaciones: objetivo y usos.
- Fallas que afectan las prestaciones.
- Fallas de hardware y software.



2. **PAS2P:** una herramienta para analizar y predecir la ejecución de aplicaciones paralelas.
  - Descripción y modelos.
  - Validación (experimentación).
  - Aplicación al caso de fallas.
3. **Análisis de la escalabilidad:**
  - Tipos de escalabilidad.
  - P3S Predicción la escalabilidad de las aplicaciones paralelas.
  - Escalabilidad y fallas.
4. **Mejorando las prestaciones:**
  - Selección del mapping y recursos de sistema para mejorar las prestaciones de las aplicaciones científicas.
  - Tolerancia a Fallos
5. **Estrategias para aumentar la fiabilidad y disponibilidad de un computador paralelo:**
  - Tipos de fallos. Técnicas de Redundancia.
  - Protocolos de “rollbackrecovery” basados en Checkpoint.
  - Protocolos de “rollback-recovery” basados en Log.
  - Métodos para la evaluación de la tolerancia a fallos.
  - Inyección de fallos
6. **Tolerancia a fallos en computadores paralelos:**
  - Estrategias para la tolerancia a fallos permanentes.
  - Estrategias para la tolerancia a fallos transitorios.
  - Evaluación del overhead.
  - Impacto del fallo.
  - Configuración de la tolerancia a fallos.
  - Ejemplo de una arquitectura Paralela Tolerante a fallos: RADIC.
7. **Middleware para tolerancia a fallos a nivel de la aplicación.**
  - Estrategias de incorporación de soft para la tolerancia a fallos permanentes.
  - Estrategias de incorporación de soft para la tolerancia a fallos transitorios.

## **ACTIVIDADES EXPERIMENTALES y DE INVESTIGACION**

### **• Herramienta PAS2P.**

- Análisis de la aplicación.
- Extracción de la firma. Análisis del efecto Mapping.

### **• RADIC FT.**

- Operaciones de Checkpoint / restart
- Análisis del efecto del Mapping



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA



## **ACTIVIDADES EXPERIMENTALES y DE INVESTIGACION**

### **Tareas en Laboratorio (presencial o remoto)**

Tal como se explica en el ítem relacionado con la metodología, ésta se basa en clases sincrónicas (presenciales o remotas) combinadas con actividades demostrativas en el laboratorio para aplicar los conceptos teóricos y que así el alumno adquiera las competencias y habilidades sobre cada uno de los temas que forman parte del contenido de la asignatura.

Además el alumno debe analizar un proyecto/desarrollo relacionado con los temas dictados en la teoría, cuyo despliegue práctico se realiza sobre una infraestructura virtualizada que los alumnos pueden acceder en forma remota (en el Laboratorio dedicado a Paralelismo en el Postgrado).

### **Investigación/ Estudios adicionales:**

Los alumnos analizarán papers relacionados con los problemas de Fallas en función del crecimiento de las arquitecturas de procesamiento paralelo y el enfoque de las soluciones a nivel de la capa de sistema operativo y de la capa de aplicación.

Se les propondrán temas de I+D orientados al estudio comparativo de soluciones de sistemas paralelos, de modo de detectar y corregir/administrar condiciones de falla.

## **METODOLOGIA Y MODALIDAD DE EVALUACION**

La metodología se basa en clases sincrónicas a través del sistema de videoconferencias adoptado por el Postgrado de Informática combinadas con sesiones en el laboratorio remoto para aplicar los conceptos teóricos y que así el alumno adquiera las competencias y habilidades sobre cada uno de los temas que forman parte del contenido de la asignatura.

Se requiere un 80% de asistencia a los encuentros sincrónicos, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

El trabajo se complementa con un proyecto experimental que debe desarrollar el alumno para cumplimentar las horas asignadas con soporte tutorizado por el profesor (*on-line*) y seguimiento a través del Entorno Virtual IDEAS contemplado en el SIED de la Facultad de Informática de la UNLP.

El despliegue práctico se realizará sobre una infraestructura virtualizada accesible al alumno en la que se puede “inyectar” diferentes tipos de fallas para estudiar soluciones mediante las herramientas explicadas en el curso.

La evaluación se realizará mediante un examen escrito al final de las sesiones sincrónicas.



## RECURSOS Y MATERIALES DE ESTUDIO

Como materiales de estudio, se dispone de:

- Presentaciones multimedia desarrolladas ad-hoc para introducir cada uno de los diferentes ejes temáticos.
- Ejemplos donde se aplican los conceptos teóricos
- Ejercicios prácticos que son desarrollados en clase
- Material de lectura para estudiar y profundizar conceptos abordados en las clases
- Enlaces a artículos de actualidad de repositorios reconocidos en el área
- Acceso a equipamiento remoto situado en la Facultad de Informática de la UNLP.
- Software específico para determinadas actividades de laboratorio que se detallan en este programa.

## ACTIVIDADES EXPERIMENTALES Y APROPIACIÓN DE SABERES

Los trabajos experimentales pueden desarrollarse en cada clase o continuarse en más de una clase. Parten de una especificación/consigna del docente (explicada en la clase) y un trabajo individual o en grupos que interactúan en el que los alumnos resuelven un problema experimental concreto relacionado con la temática.

Los trabajos podrán ser individuales o grupales. Para esto último se configura el entorno virtual para que los alumnos del mismo grupo se encuentren en un espacio virtual diferente del resto. Durante el desarrollo del trabajo, el docente estará conectado respondiendo dudas y consultas.

Estos trabajos pretenden desarrollar y/o fortalecer las aptitudes de opinión crítica en los temas relativos del curso. Los alumnos deberán sintetizar su comprensión de los temas, al realizar correctamente la tarea experimental propuesta.

También se pretende desarrollar la capacidad de poder comunicar y transmitir los resultados, en presentaciones pautadas a lo largo del curso.

En general, finalizada una actividad, hay una sesión de discusión conjunta donde los participantes comunicarán sus opiniones e intercambiarán los distintos puntos de vista.



## BIBLIOGRAFÍA BASICA

### **Fault-Tolerant Message-Passing Distributed Systems: An Algorithmic Approach**

Michel Raynal, (2018) Springer.

### **Fault Tolerant System**

I Koren, C, Mani Krishna (2007). Morgan Kaufmann. Elsevier

### **Fault tolerance at system level based on RADIC architecture**

M. Castro-Leon, H. Meyer, D. Rexachs, E. Luque (2015), J. Parallel Distrib. Comput. Volume 86, December 2015, Pages 98-111, ISSN 0743-7315, (2015),  
DOI: <http://dx.doi.org/10.1016/j.jpdc.2015.08.005>

### **High availability for parallel computers**

D Rexachs, E Luque. Journal of Computer Science and Technology (JCS&T) 10 (3), 110-116

## BIBLIOGRAFÍA COMPLEMENTARIA

### **Predicting robustness against transient faults of MPI based programs**

João Gramacho, Alvaro Wong, Dolores Rexachs, Emilio Luque (2016) “”. IJCSE 12(2/3): 155-165 ISSN: 1742-7185. DOI: <http://dx.doi.org/10.1504/IJCSE.2016.076218>

### **Hybrid Message Logging. Combining advantages of Sender-based and Receiver-based Approaches.**

Hugo Meyer, Dolores Rexachs, Emilio Luque (2014) ICCS 2014: 2380-2390.

### **A tool for detecting transient faults in execution of parallel scientific applications on multicore clusters**

D. Montezanti, E. Rucci, D Rexachs, E Luque, M. Naiouf, A. de Giusti. (2014).. Journal of Computer Science & Technology (JCS&T). Vol. 14 - No. 1. pp 32-38.

### **SMCV: a Methodology for Detecting Transient Faults in Multicore Clusters**

Diego Montezanti, Fernando Emmanuel Frati, Dolores Rexachs, Emilio Luque, Marcelo R. Naiouf, Armando De Giusti (2012). CLEI Electron. J. 15(3)

### **A Survey of Rollback-recovery Protocols in Message-passing Systems**

Elnozahy, E.N.; Alvisi, L.; Wang, Y. & Johnson, D.B. (2002) ACM Computing Surveys, ACM Press, 34, 375-408.



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA



### **What is Missing in Current Checkpoint Interval Models?**

L. Fialho, D. Rexachs and E. Luque, Distributed Computing Systems (ICDCS), 2011 31st International Conference on, Minneapolis, MN, 2011, pp. 322-332.

doi: 10.1109/ICDCS.2011.12

### **Providing Non-stop Service for Message-Passing Based Parallel Applications with RADIC.**

Guna Santos, Angelo Duarte, Dolores Rexachs, Emilio Luque (2008). Euro-Par 2008: 58-67.

### **Fault Tolerant Master-Worker over a Multi-Cluster Architecture.**

J. Rodrigues de Souza, Eduardo Argollo, Angelo Duarte, Dolores Rexachs, Emilio Luque (2005) PARCO 2005: 465-472.