# Modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

## Tesis de grado

Daniel Wladdimiro Cottet
Profesor guía: Dr. Nicolás Hidalgo C.
Profesora co-guía: Dra. Erika Rosas O.

Departamento de Ingeniería Informática
Universidad de Santiago de Chile

Santiago - Chile
2015

# Contenidos

La interacción entre los distintos usuarios generó una gran cantidad de datos, la cual se puede procesar y obtener información relevante

- Sistemas de procesamiento capaces de lidiar con restricciones de temporalidad
- Manejo de grandes flujos de datos en tiempo real
- Debido a la necesidad de respuestas rápidas y actualizadas
- Apoyo en la toma de decisiones
- Por ejemplo:
  - Predicciones del comportamiento en la bolsa de valores
  - Recopilación de información en caso de emergencia
  - Seguridad en redes

- Tipos de sistemas de procesamiento de *stream*
  - S4
  - Storm
  - Samza
- *Problemas*
  - Poca adaptación del sistema en tiempo de ejecución
  - Posibles problemas de distribución de carga
  - Baja en el rendimiento
  - Pérdida de recursos e información

Dado el carácter estático del grafo de procesamiento en tiempo de ejecución y el carácter altamente dinámico del tráfico, pueden surgir problemas de balance de carga entre los operadores de la topología, sobrecargando alguno de estos y comprometiendo el rendimiento del sistema.

# Contenidos

- Perspectivas al problema de balance de carga en procesamiento de *stream*
  - Recursos físicos
  - Grafo de operadores
- Para la optimización del sistema, se presentan dos enfoques [Dong and Akl, 2006]
  - Estático
  - Dinámico

# Balance de carga

## Enfoque dinámico

- En el estado del sistema
- Las variables y estados de cada uno de sus atributos
- Cambios en el sistema ante una anomalía
- Tipos de modelo para las soluciones:
  - Reactivo [Gulisano et al., 2012]
  - Predictivo [Nguyen et al., 2013]

- Existes distintas técnicas que son utilizadas en ambos modelos
- Por ejemplo:
  - Planificación determinista [Xu et al., 2014, Dong et al., 2007]
  - Descarte de eventos [Sheu and Chi, 2009]
  - Migración [Xing et al., 2005]
  - Fisión [Gulisano et al., 2012, Ishii and Suzumura, 2011, Gedik et al., 2014, Fernandez et al., 2013]

# Contenidos

### Objetivo general

Diseño, construcción y evaluación de un modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

### Objetivo general

Diseño, construcción y evaluación de un modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

1. Diseñar e implementar un algoritmo reactivo que permita analizar en el momento la carga de los operadores

## Objetivo general

Diseño, construcción y evaluación de un modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

1. Diseñar e implementar un algoritmo reactivo que permita analizar en el momento la carga de los operadores
2. Diseñar e implementar un algoritmo de predicción que permita estimar la carga de los operadores

### Objetivo general

Diseño, construcción y evaluación de un modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

1. Diseñar e implementar un algoritmo reactivo que permita analizar en el momento la carga de los operadores
2. Diseñar e implementar un algoritmo de predicción que permita estimar la carga de los operadores
3. Diseñar e implementar un algoritmo que permita la administración del número de operadores del grafo de procesamiento de forma elástica

### Objetivo general

Diseño, construcción y evaluación de un modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

1. Diseñar e implementar un algoritmo reactivo que permita analizar en el momento la carga de los operadores
2. Diseñar e implementar un algoritmo de predicción que permita estimar la carga de los operadores
3. Diseñar e implementar un algoritmo que permita la administración del número de operadores del grafo de procesamiento de forma elástica
4. Diseñar y construir experimentos que permitan validar la hipótesis formulada

### Objetivo general

Diseño, construcción y evaluación de un modelo elástico de replicación de operadores para un sistema de procesamiento de *stream* en tiempo real

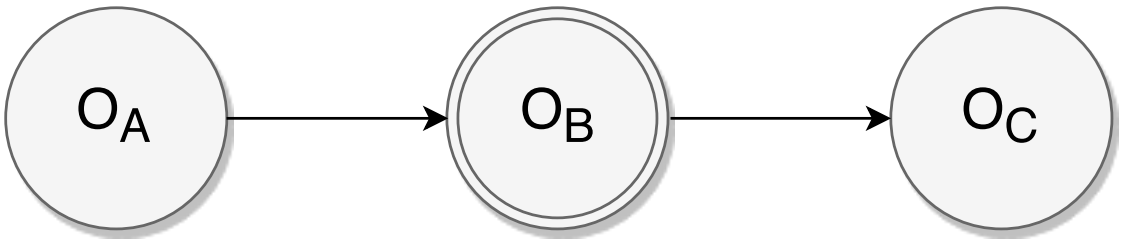
1. Diseñar e implementar un algoritmo reactivo que permita analizar en el momento la carga de los operadores
2. Diseñar e implementar un algoritmo de predicción que permita estimar la carga de los operadores
3. Diseñar e implementar un algoritmo que permita la administración del número de operadores del grafo de procesamiento de forma elástica
4. Diseñar y construir experimentos que permitan validar la hipótesis formulada
5. Evaluar y analizar el rendimiento del modelo a través de aplicaciones generadas sobre un sistema de procesamiento de *stream*

Los alcances del proyecto se encuentran:

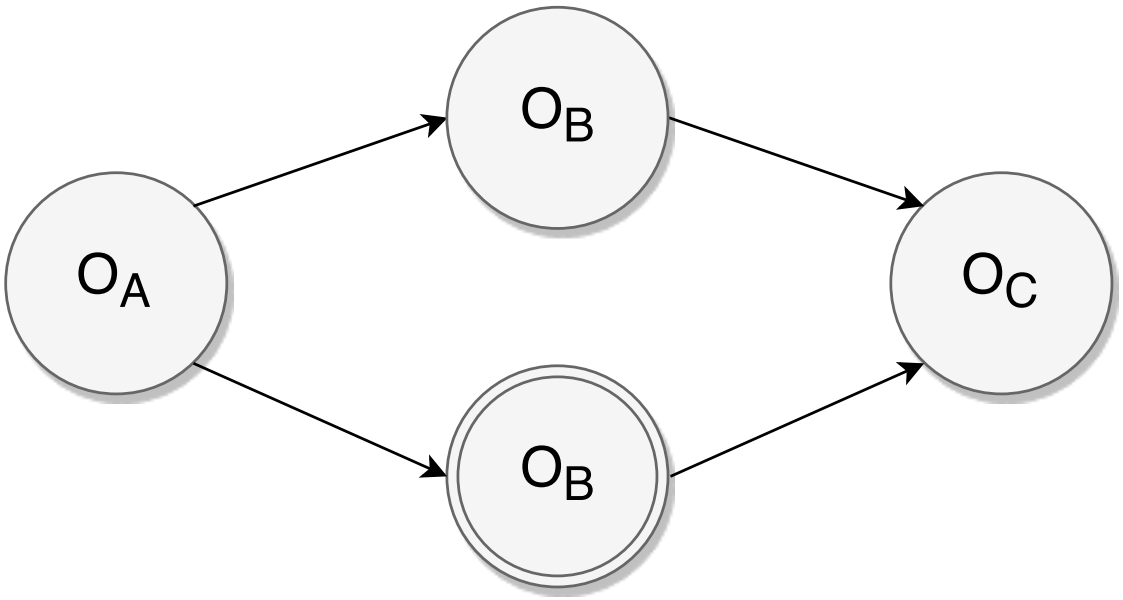
- Evaluación sobre un sólo sistema de procesamiento de *stream*
- Datos emitidos de la fuente de datos son homogéneos
- Distribución de carga a nivel de operadores y no de máquinas
- No garantiza completo procesamiento de los datos
- Costo de comunicación de manera igualitaria para todos los operadores

# Contenidos

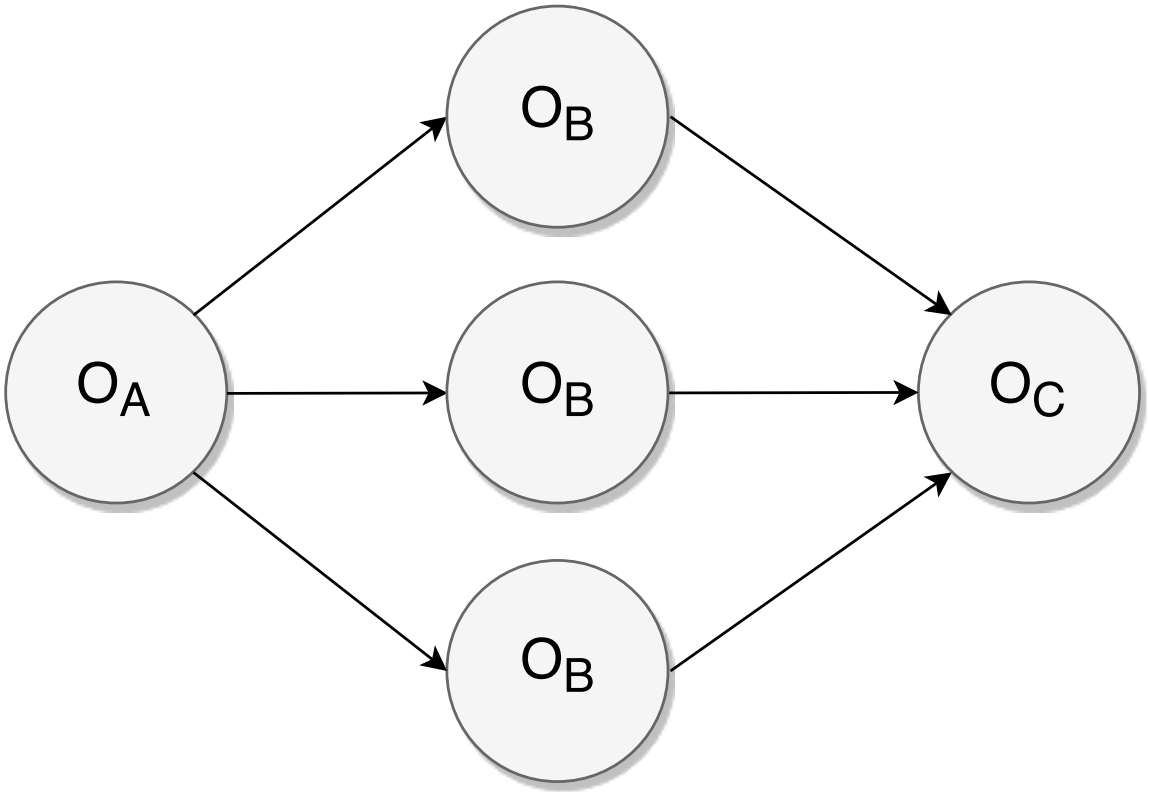
- Recursos lógicos del sistema según el enfoque dinámico
- Bajo overhead $\rightarrow$ Escalable
- Técnica de fisión



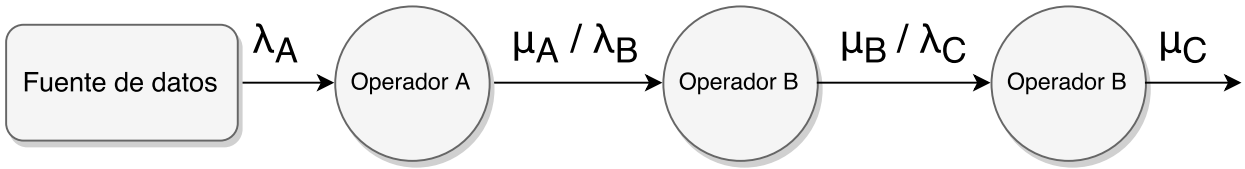
- Recursos lógicos del sistema según el enfoque dinámico
- Bajo overhead $\rightarrow$ Escalable
- Técnica de fisión



- Recursos lógicos del sistema según el enfoque dinámico
- Bajo overhead $\rightarrow$ Escalable
- Técnica de fisión



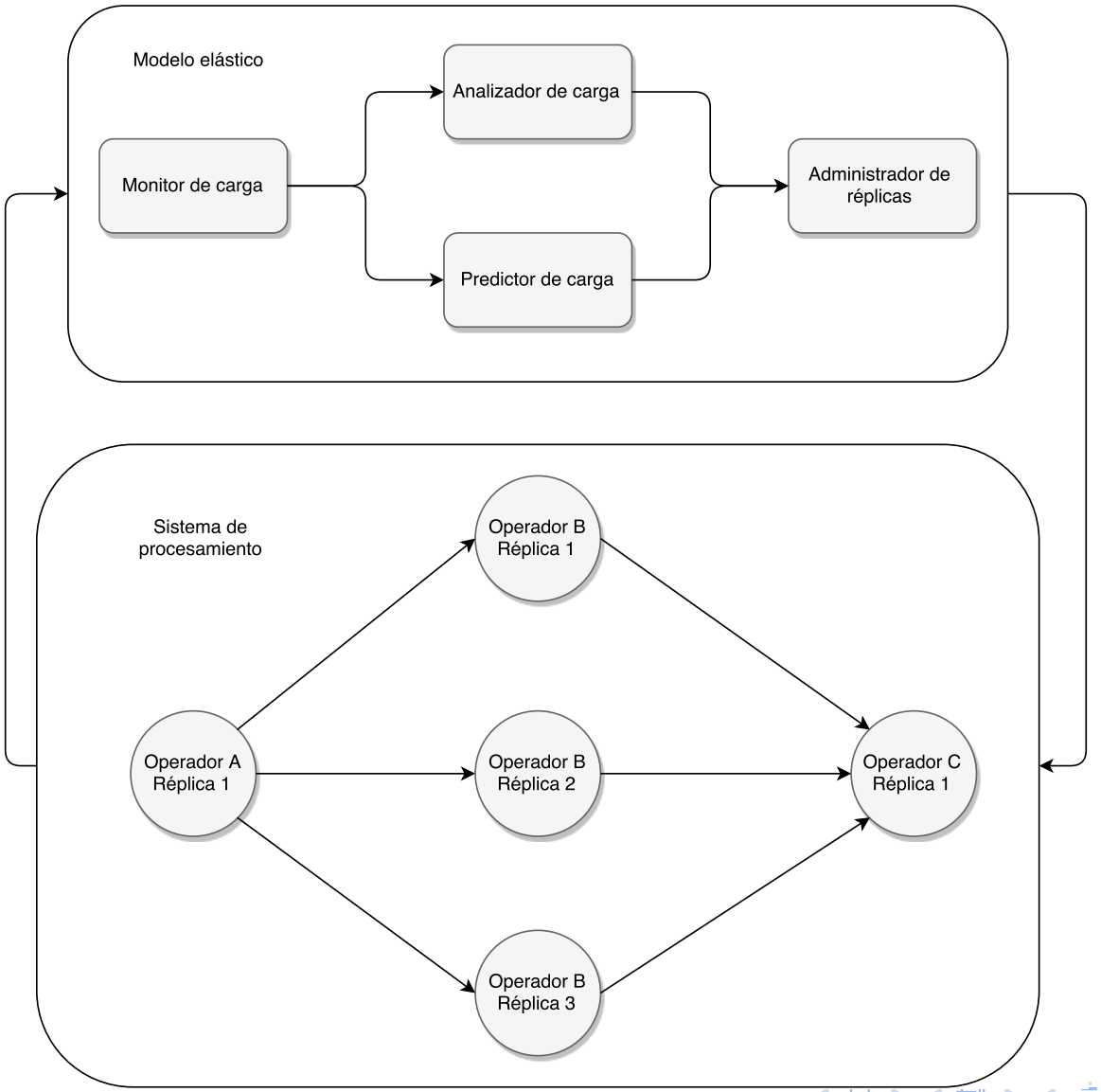
- Umbrales $\rightarrow$ Tasa de rendimiento $\rho$
  - $\rho = \frac{\lambda}{s\mu}$




- Enfoque dinámico y elasticidad
  - Estados: ocioso, estable e inestable
- Dos tipos de algoritmos: reactivo y predictivo
- Recolector de datos y administrador de réplicas

# Diseño del modelo elástico

## Análisis del modelo elástico



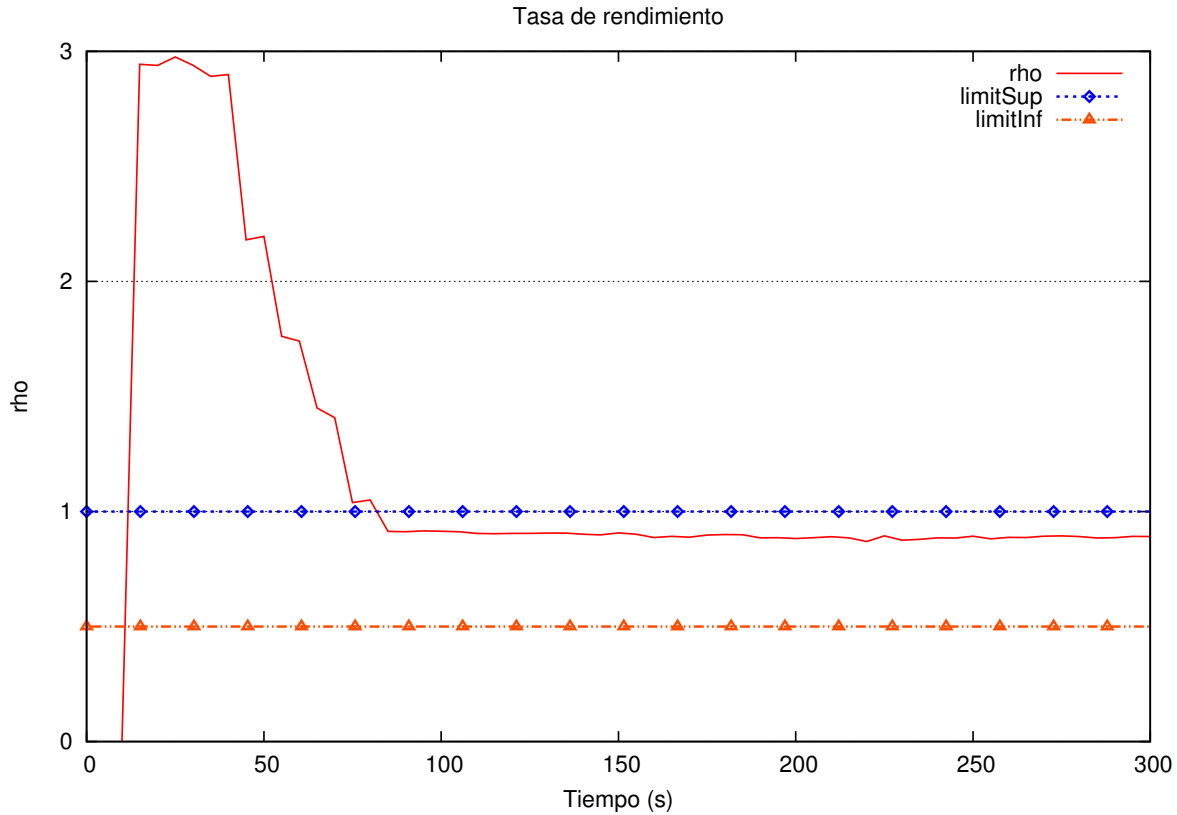
El monitor de carga es el encargado de recolectar los datos

- Algoritmo reactivo $\rightarrow$ Tasa de procesamiento $\rho$
  - Tasa de rendimiento $\mu$ es homogénea
  - Ventana de tiempo $T_r$
- Algoritmo predictivo $\rightarrow$ Historial
  - Ventanas de tiempo de 1 segundo
  - $n$ muestras
  - Ventana de tiempo $T_p$

- Análisis del estado del operador $\rightarrow$ Período de tiempo
- Tasa de rendimiento $\rho$

$$\begin{array}{cc} \rho > 1 & \text{Inestable} \\ 1 \geqslant \rho \geqslant 0.5 & \text{Estable} \\ \rho < 0.5 & \text{Ocioso} \end{array}$$

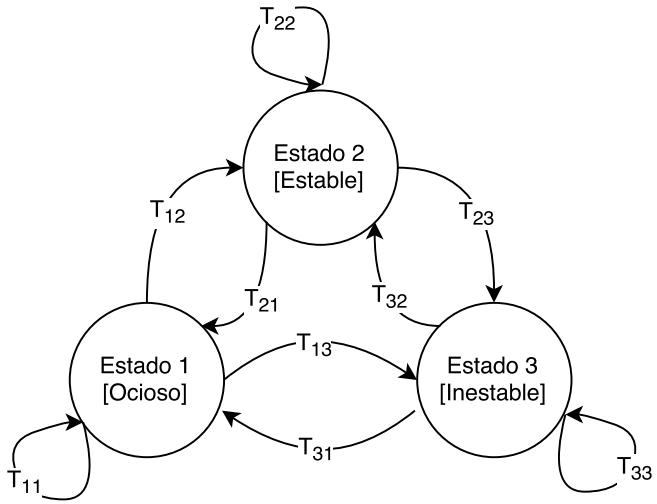
- Comportamiento de la tasa de rendimiento



# Diseño del modelo elástico

## Algoritmo predictivo

- Definir muestras en tiempos discretos, las cuales cambian con el tiempo según un proceso estocástico
- Determinar los estados finitos que se utilizan para la conformación de la cadena
- Obtener una cantidad representativa de muestras para la construcción de la cadena de Markov en el período analizado



# Diseño del modelo elástico

## Algoritmo predictivo

- Construcción de la matriz de transición
  - La transición de estados de un período a otro

$$P = \begin{bmatrix} T_{1,1} & T_{1,2} & T_{1,3} \\ T_{2,1} & T_{2,2} & T_{2,3} \\ T_{3,1} & T_{3,2} & T_{3,3} \end{bmatrix}$$

- Ecuación de Chapman-Kolmogórov $\rightarrow$ Distribución Estacionaria
  - Comportamiento a futuro de la Cadena de Markov

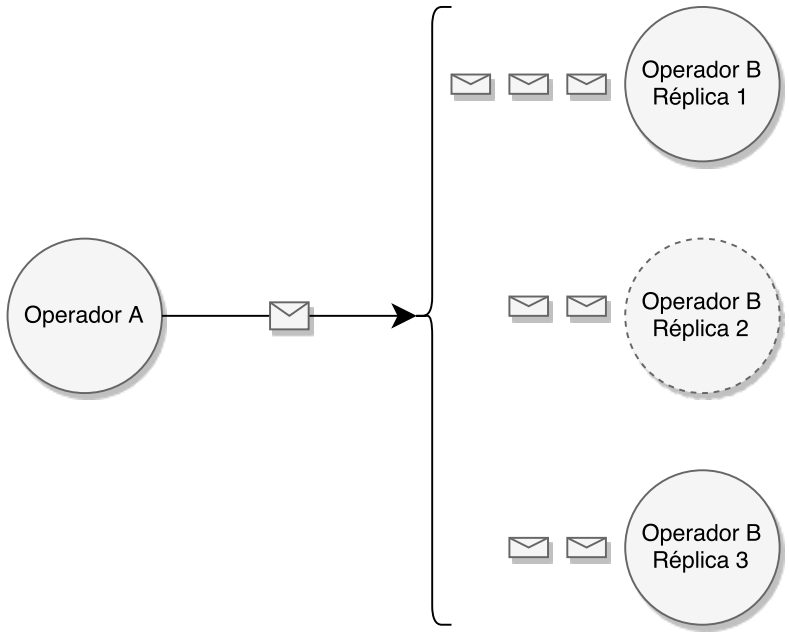
$$\begin{bmatrix} \Pi_1 & \Pi_2 & \Pi_3 \end{bmatrix}_{(t+1)}$$

- $\sigma(\Pi_1, \Pi_2, \Pi_3) > 0.25$
  - No posee incertidumbre
  - En caso contrario, no es un comportamiento determinante

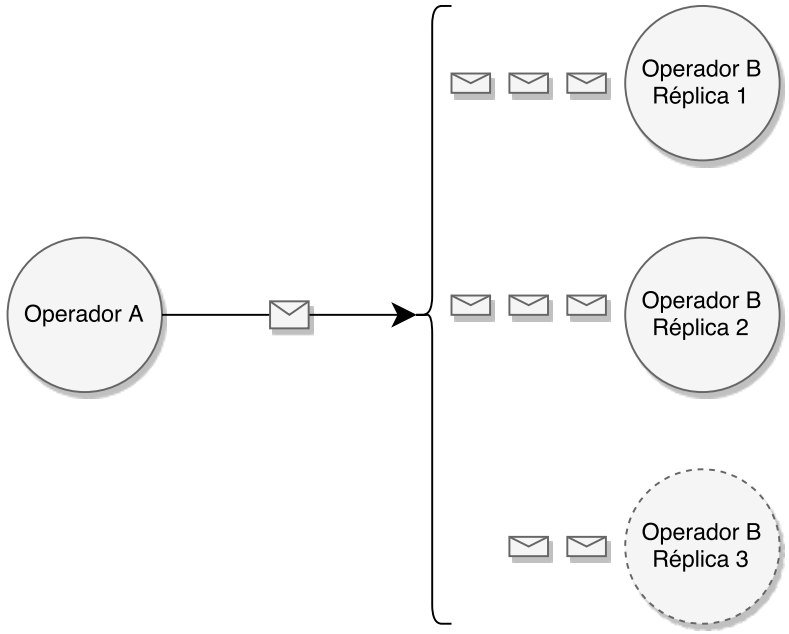
- Administración de réplicas de un operador
- Recursos disponibles de la máquina
- Según el período se ejecuta
  - $T_p \rightarrow$ Algoritmo predictivo
    - Menor frecuencia
    - Mayor cómputo
    - Modifica mayor cantidad de réplicas
  - $T_r \rightarrow$ Algoritmo reactivo
    - Dos alertas $\rightarrow$ Modifica

# Contenidos

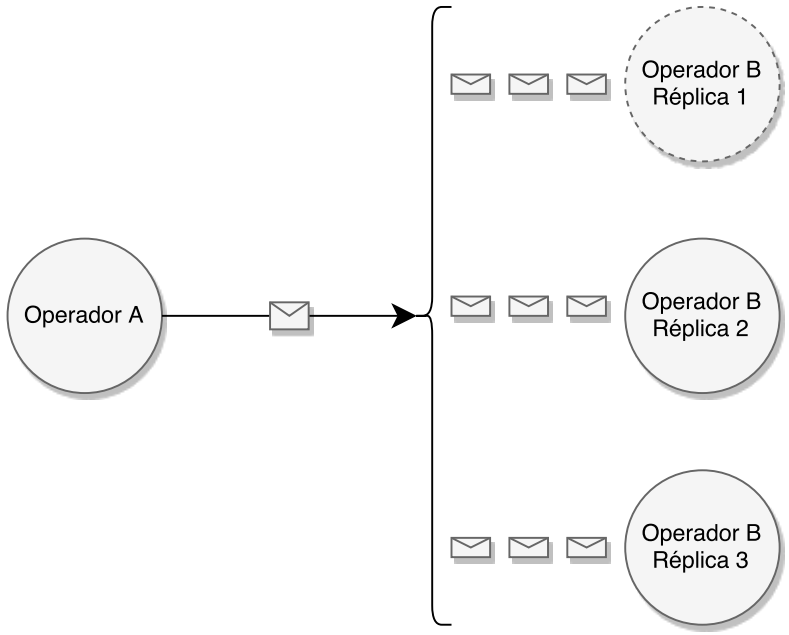
- SPS S4 $\rightarrow$ Modifica el código fuente
  - Cantidad de eventos entrantes y salientes en cada PE
- Distribución de la carga según la cola
  - Política según el largo de la cola



- SPS S4 $\rightarrow$ Modifica el código fuente
  - Cantidad de eventos entrantes y salientes en cada PE
- Distribución de la carga según la cola
  - Política según el largo de la cola

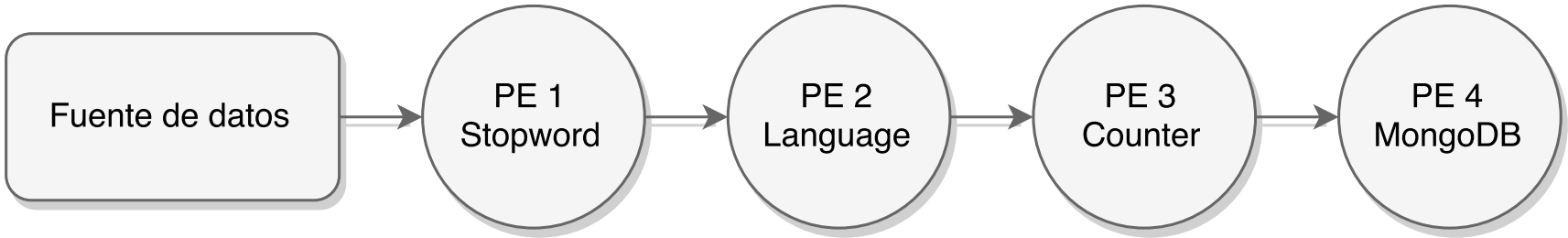


- SPS S4 $\rightarrow$ Modifica el código fuente
  - Cantidad de eventos entrantes y salientes en cada PE
- Distribución de la carga según la cola
  - Política según el largo de la cola

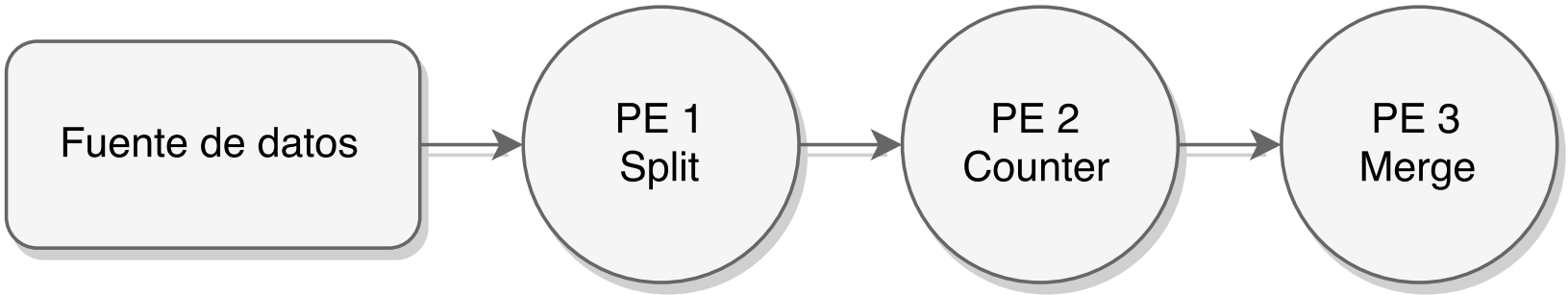


- Tres tipo de aplicaciones
  - Aplicación usando operadores con estado
  - Aplicación usando operadores sin estado
  - Aplicación sintética
- Generación de stream
  - 4.5 millones de tweets
  - 27-28 de Febrero y 1-2 de Marzo de 2010
  - Inglés, español y portugués
  - Interacción entre usuarios durante el terremoto del 27 de Febrero en Chile

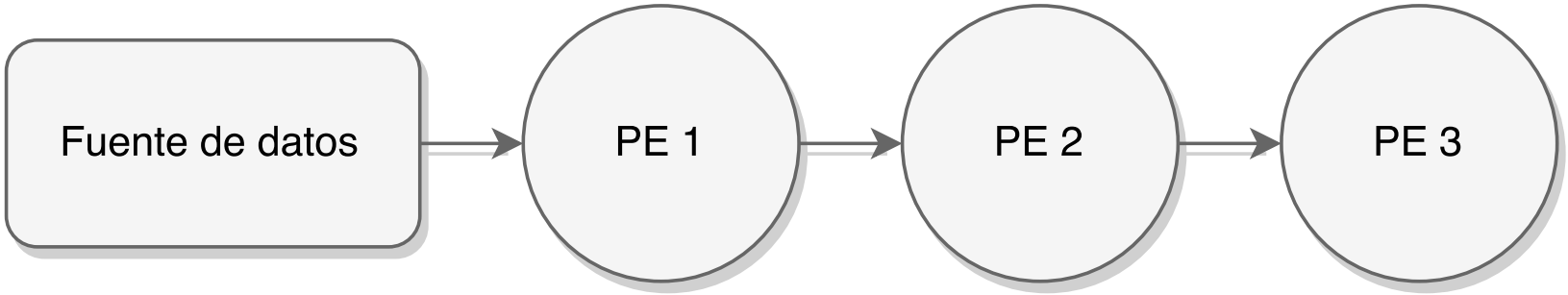
- Aplicación 1: Análisis de *tweets* en escenarios de desastres naturales



- Aplicación 2: Contador de palabras en muestras de textos



- Aplicación 3: Aplicación sintética

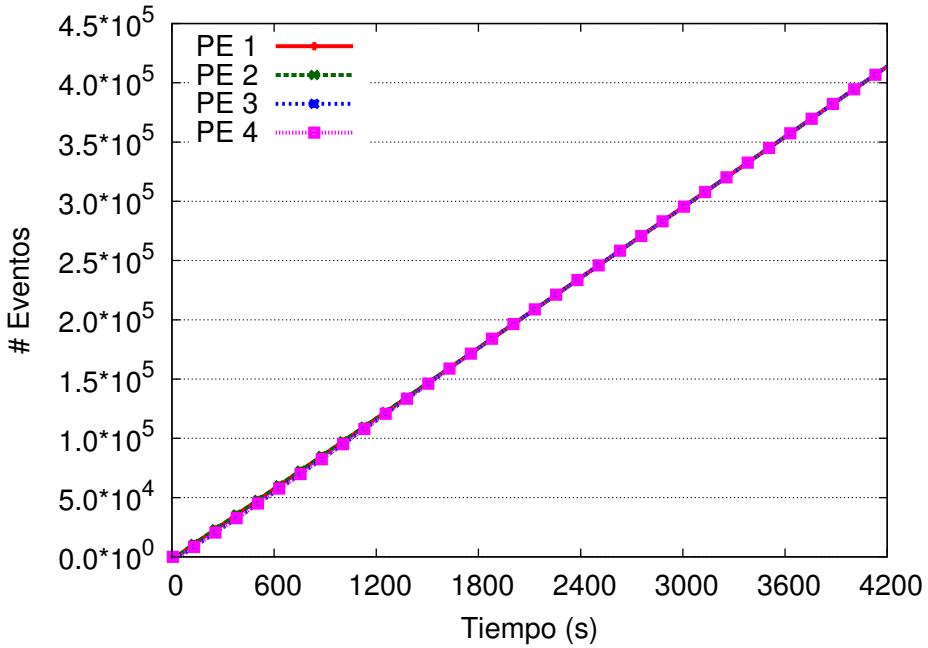


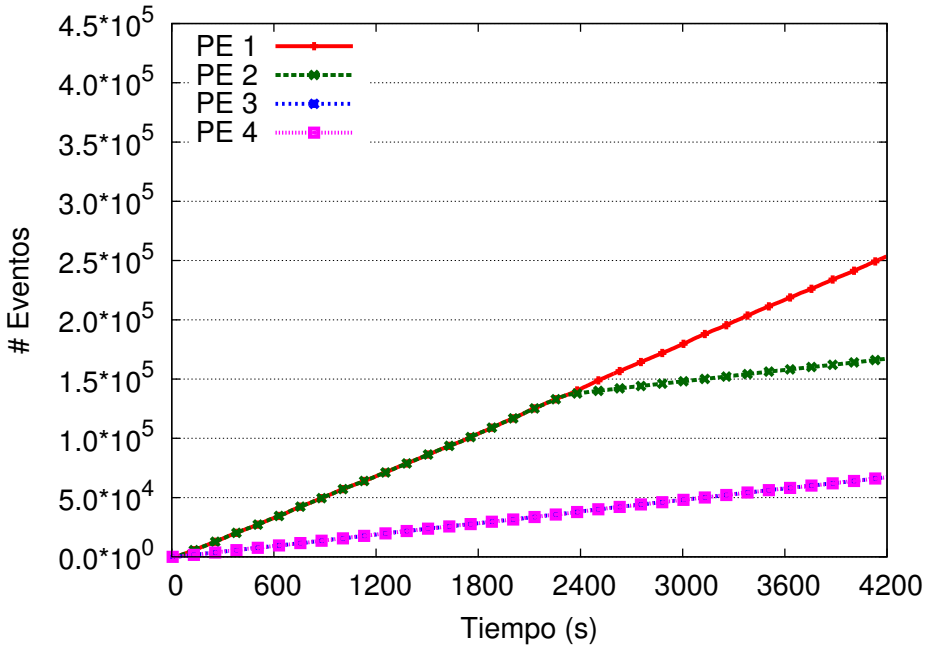

- Período de tiempo que duerme la hebra asignada al PE

| PE | Tiempo (ms) |
|---|---|
| 1 | 20 |
| 2 | 30 |
| 3 | 15 |

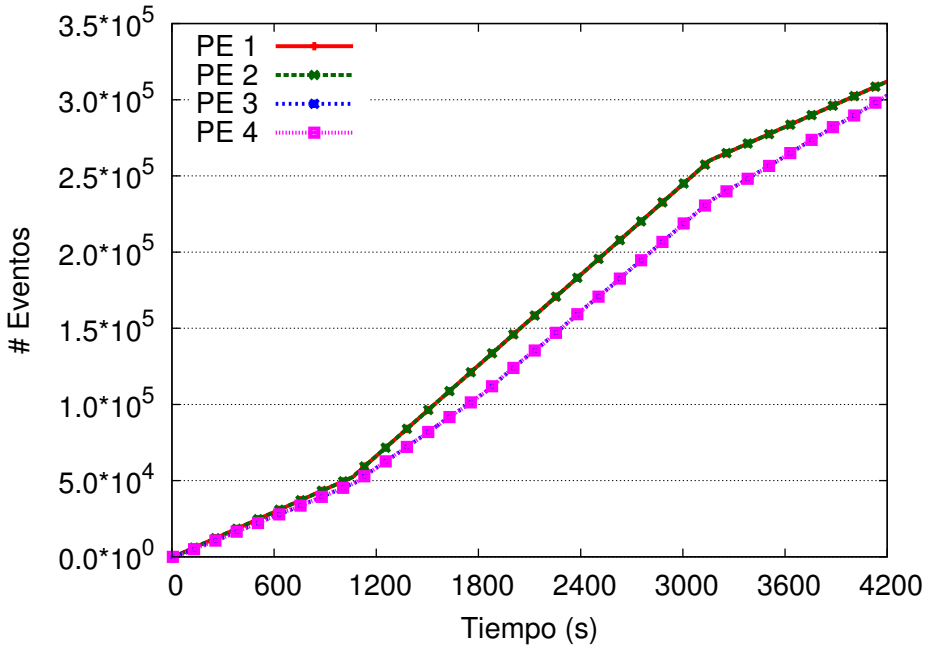
- Para la ejecución de todos los experimentos se ha utilizado una máquina con un Intel Xeon CPU E5-2650 v2 de 2.60 GHz, 32 GB de RAM y SO Ubuntu 14.04.2 LTS
- Para la evaluación de la primera y segunda aplicación se realiza dos tipos de experimentos
  - Envío constante de 100 eventos/s
  - Envío variable de 50 eventos/s en primer tercio, 100 eventos/s en segundo tercio, y 50 eventos/s en último tercio
  - Ambos en un período de 70 minutos
- Para la evaluación de la tercera aplicación se realiza un experimento
  - Envío constante de 100 eventos/s
  - Período de 15 minutos
- Cada uno de los experimentos se realiza con y sin modelo elástico

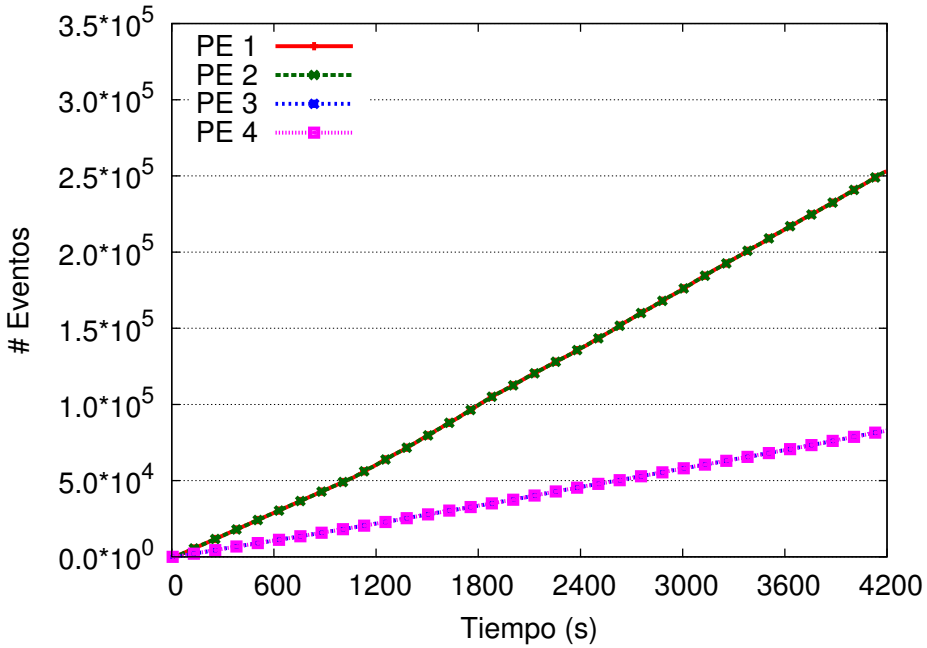
- 401.618 eventos procesados con uso del modelo *vs* 67.141 eventos procesados sin uso del modelo
- Mejora de 6 veces la cantidad de eventos procesados



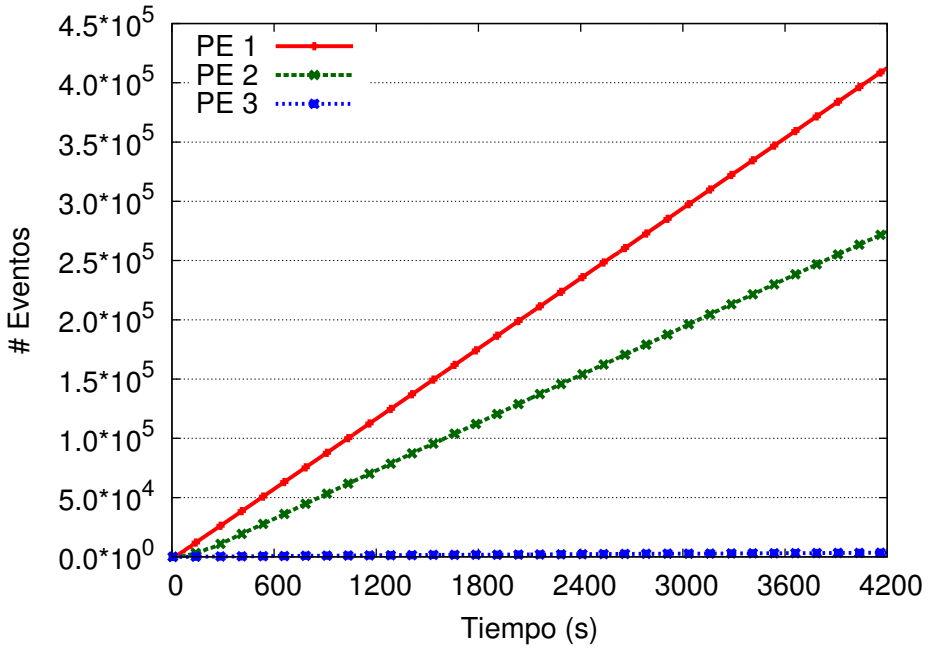



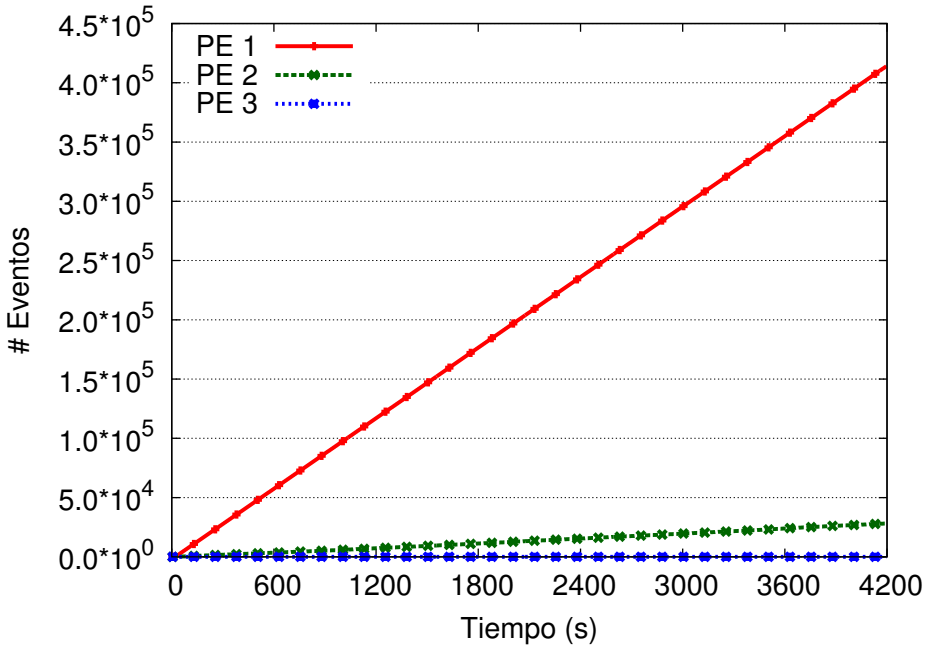
- 303.156 eventos procesados con uso del modelo *vs* 82.770 eventos procesados sin uso del modelo
- Mejora de 3 veces la cantidad de eventos procesados



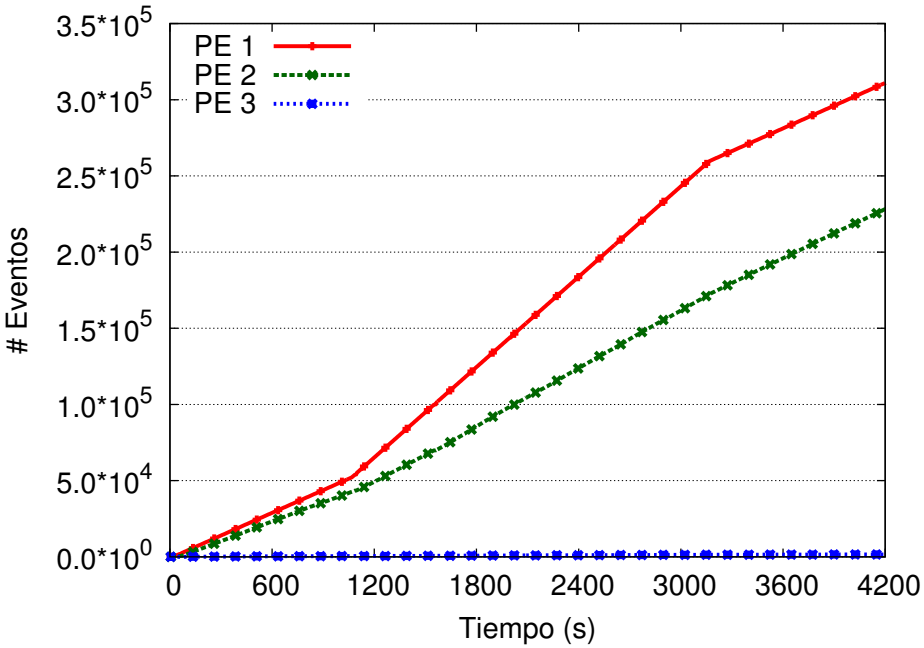



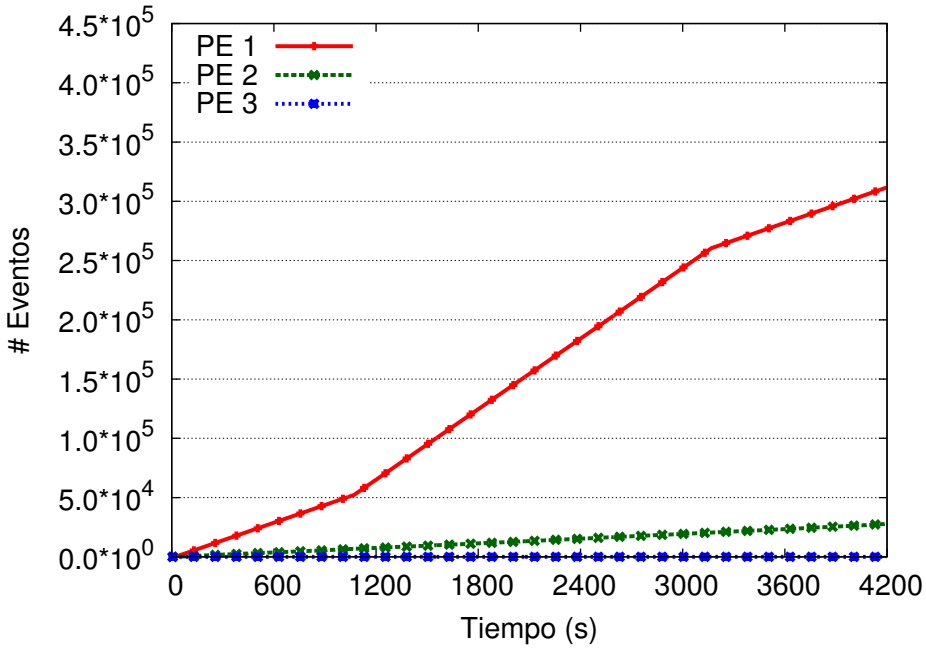
- 275.290 eventos procesados con uso del modelo *vs* 28.152 eventos procesados sin uso del modelo
- Mejora de 9 veces la cantidad de eventos procesados



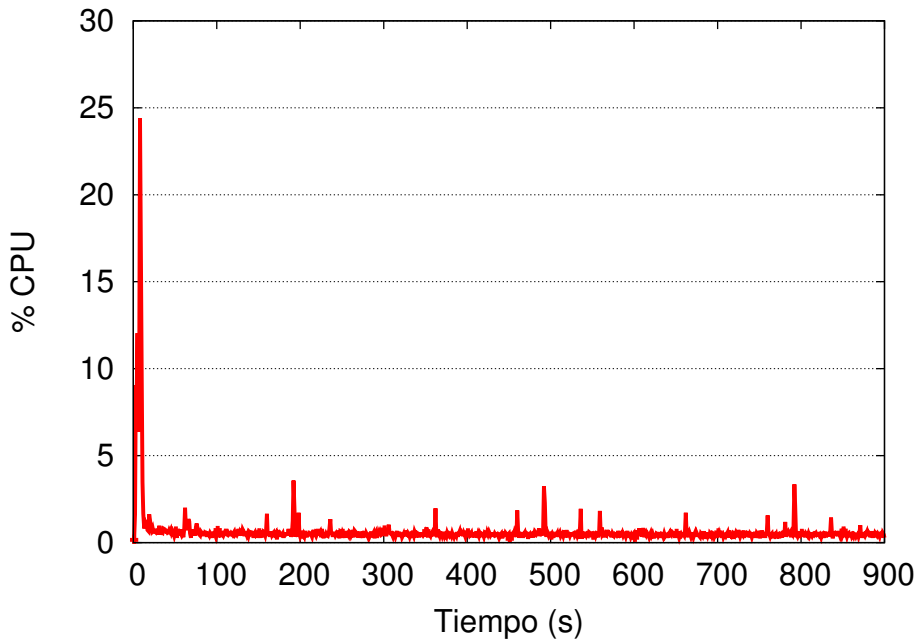



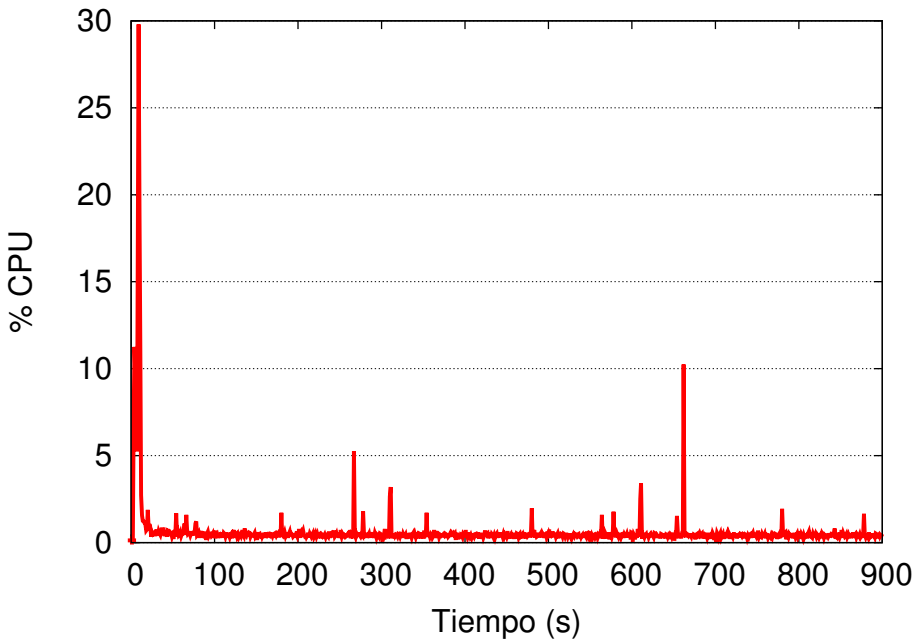
- 228.942 eventos procesados con uso del modelo *vs* 27.751 eventos procesados sin uso del modelo
- Mejora de 8 veces la cantidad de eventos procesados



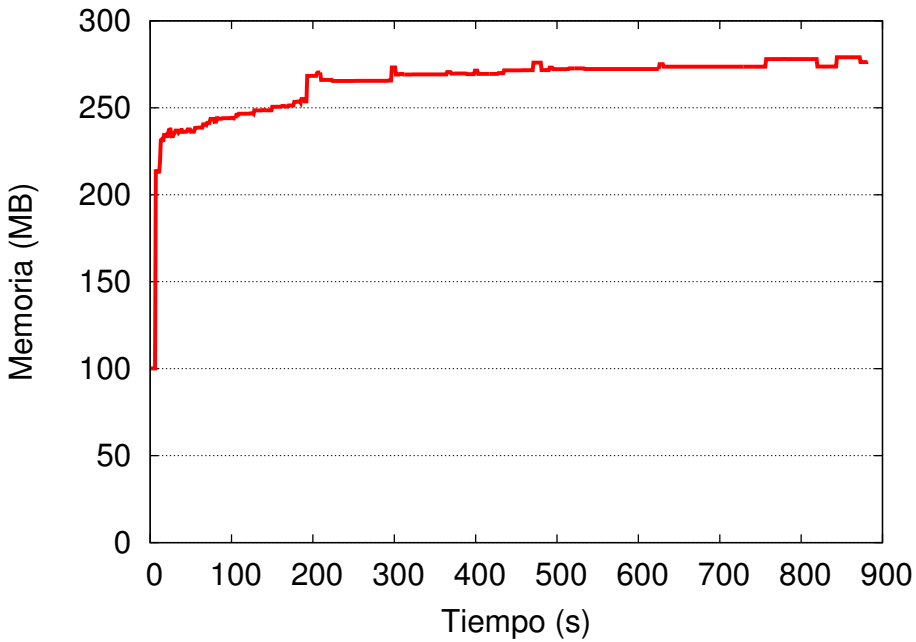



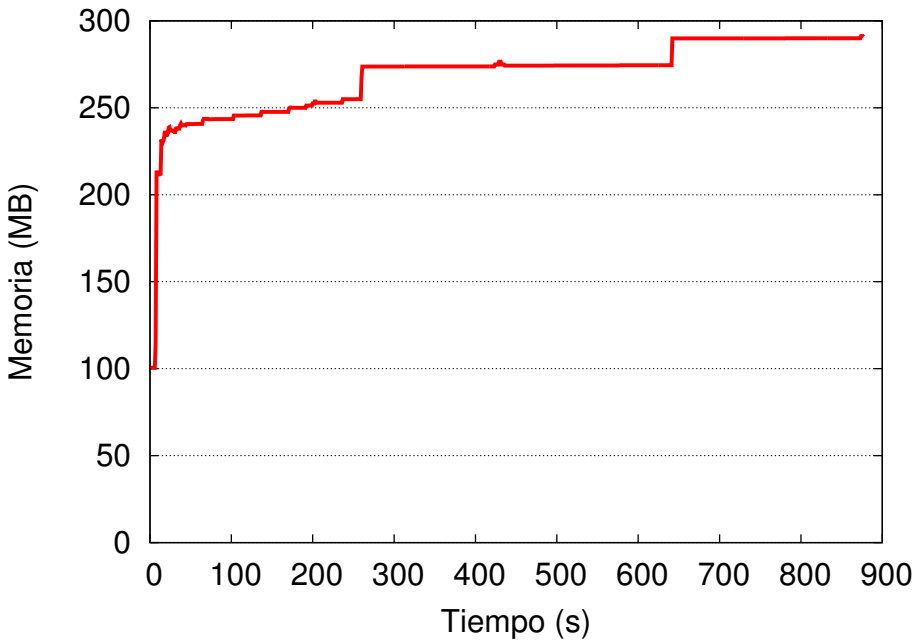
- $0,62\%$ con uso del modelo *vs* $0,61\%$ sin uso del modelo
- Aumento de un $0,01\%$ de utilización promedio de CPU



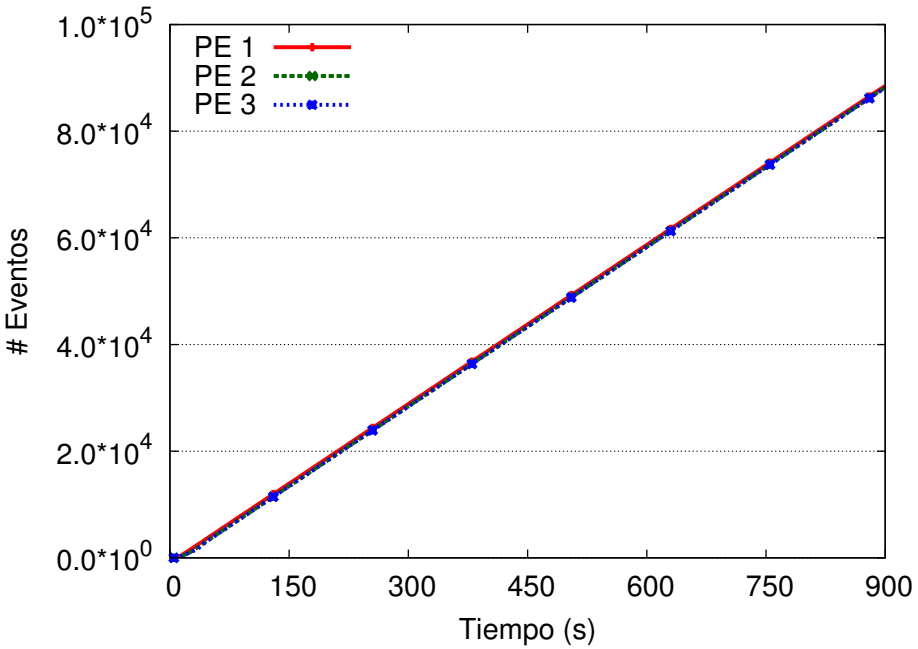



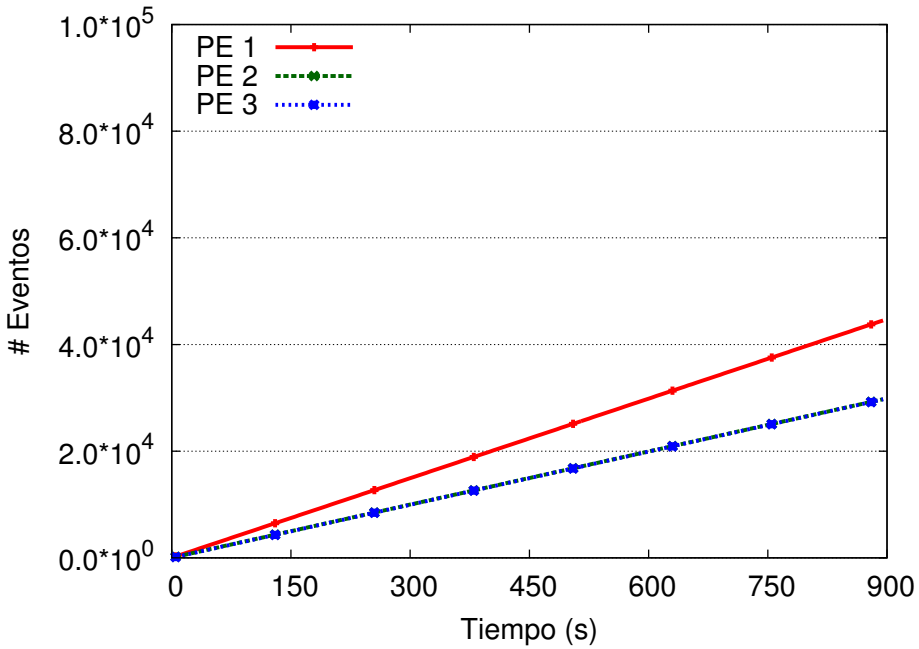
- 264MB con uso del modelo *vs* 268MB sin uso del modelo
- Disminución de $1,5\%$ de consumo de memoria RAM



- 88.169 eventos procesados con uso del modelo *vs* 28.714 eventos procesados sin uso del modelo
- Mejora de 3 veces la cantidad de eventos procesados



- Diseño e implementación de un modelo elástico
  - Algoritmo reactivo
  - Algoritmo predictivo
  - Módul de administración de réplicas
- Construcción de tres escenarios $\rightarrow$ Validación del modelo diseñado
- Evaluado y analizado el rendimiento del sistema con y sin uso del modelo elástico

# Contenidos

- Implementación en el SPS S4
  - Procesamiento de los eventos
  - Buffer
- Homogeneidad de la tasa de procesamiento
- No se realiza un análisis de los recursos físicos
- No es capaz de detectar patrones estacionarios
- Modelo diseñado
  - Bajo cómputo
  - Rápido análisis de los operadores
  - Elasticidad

- Algoritmo predictivo
  - Adaptabilidad del número de réplicas según el historial
  - Machine Learning
- Implementación en otro SPS

# Preguntas

# Bibliografía I


Dong, F. and Akl, S. G. (2006).
Scheduling algorithms for grid computing: State of the art and open problems.

Dong, M., Tong, L., and Sadler, B. M. (2007).
Information retrieval and processing in sensor networks: Deterministic scheduling versus random access.
*IEEE Transactions on Signal Processing*, 55(12):5806–5820.

Fernandez, R. C., Migliavacca, M., Kalyvianaki, E., and Pietzuch, P. (2013).
Integrating scale out and fault tolerance in stream processing using operator state management.

# Bibliografía II

In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 725–736.

Gedik, B., Schneider, S., Hirzel, M., and Wu, K. (2014).
Elastic scaling for data stream processing.
*IEEE Trans. Parallel Distrib. Syst.*, 25(6):1447–1463.

Gulisano, V., Jiménez-Peris, R., Patiño-Martínez, M., Soriente, C., and Valduriez, P. (2012).
Streamcloud: An elastic and scalable data streaming system.
*IEEE Trans. Parallel Distrib. Syst.*, 23(12):2351–2365.

# Bibliografía III



Ishii, A. and Suzumura, T. (2011).
Elastic stream computing with clouds.
In *IEEE International Conference on Cloud Computing, CLOUD 2011, Washington, DC, USA, 4-9 July, 2011*, pages 195–202.


Nguyen, H., Shen, Z., Gu, X., Subbiah, S., and Wilkes, J. (2013).
AGILE: elastic distributed resource scaling for infrastructure-as-a-service.
In *10th International Conference on Autonomic Computing, ICAC'13, San Jose, CA, USA, June 26-28, 2013*, pages 69–82.


Sheu, T. and Chi, Y. (2009).
Intelligent stale-frame discards for real-time video streaming over wireless ad hoc networks.
*EURASIP J. Wireless Comm. and Networking*, 2009.

# Bibliografía IV



Xing, Y., Zdonik, S. B., and Hwang, J. (2005).
Dynamic load distribution in the borealis stream processor.
In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 791–802.


Xu, J., Chen, Z., Tang, J., and Su, S. (2014).
T-storm: Traffic-aware online scheduling in storm.
In *IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014, Madrid, Spain, June 30 - July 3, 2014*, pages 535–544.

# Dudas