

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



1. Introducción
2. Balance de carga
3. Solución propuesta
4. Diseño del modelo elástico
5. Experimentos y evaluación
6. Conclusiones

Introducción



Antecedentes y motivación

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



Introducción

Antecedentes y motivación



- 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



1. Introducción
2. Balance de carga
3. Solución propuesta
4. Diseño del modelo elástico
5. Experimentos y evaluación
6. Conclusiones



- 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



- 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]



- Existe 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]



1. Introducción
2. Balance de carga
3. Solución propuesta
4. Diseño del modelo elástico
5. Experimentos y evaluación
6. Conclusiones



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*

Solución propuesta

Objetivos y alcance del proyecto



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



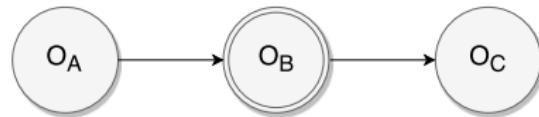
1. Introducción
2. Balance de carga
3. Solución propuesta
4. Diseño del modelo elástico
5. Experimentos y evaluación
6. Conclusiones



Diseño del modelo elástico

Análisis del modelo elástico

- Recursos lógicos del sistema según el enfoque dinámico
- Bajo overhead → Escalable
- Técnica de fisión
 - Elasticidad → Añadir/Reducir la cantidad de réplicas del operador
 - Solucionando:
 - Sobrecarga
 - Utilización de recursos

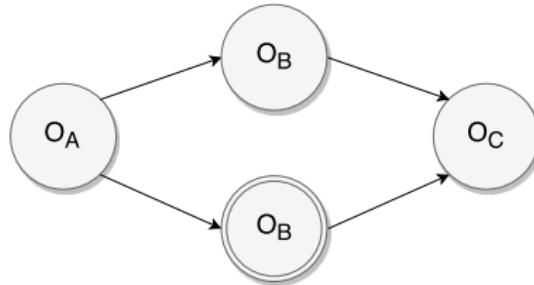




Diseño del modelo elástico

Análisis del modelo elástico

- Recursos lógicos del sistema según el enfoque dinámico
- Bajo overhead → Escalable
- Técnica de fisión
 - Elasticidad → Añadir/Reducir la cantidad de réplicas del operador
 - Solucionando:
 - Sobrecarga
 - Utilización de recursos

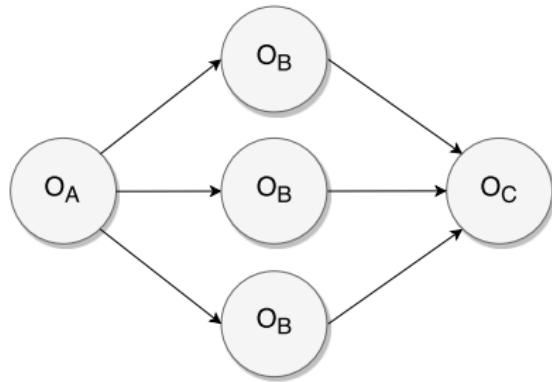




Diseño del modelo elástico

Análisis del modelo elástico

- Recursos lógicos del sistema según el enfoque dinámico
- Bajo overhead → Escalable
- Técnica de fisión
 - Elasticidad → Añadir/Reducir la cantidad de réplicas del operador
 - Solucionando:
 - Sobrecarga
 - Utilización de recursos

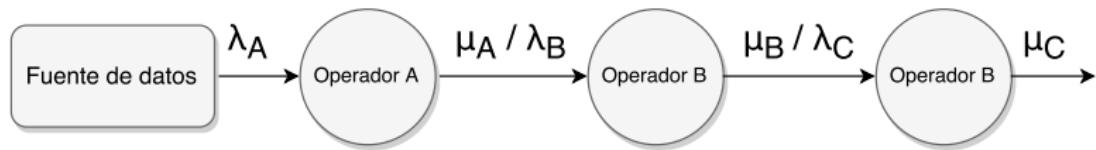


Diseño del modelo elástico

Análisis del modelo elástico



- El modelo elástico está basado en la teoría de colas
 - Tasa de rendimiento ρ
 - $\rho = \frac{\lambda}{s\mu}$

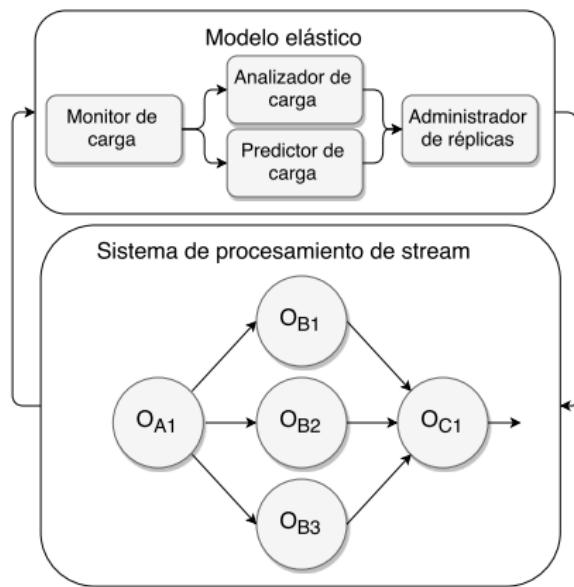




Diseño del modelo elástico

Análisis del modelo elástico

- Se propone un modelo elástico compuesto por 4 módulos
- Basado en el modelo MAPE (Monitoreo, Análisis, Procesamiento y Ejecución)





El monitor de carga es el encargado de recolectar los datos del grafo de operadores

- $\rho = \frac{\lambda}{\mu s}$
- Algoritmo reactivo
- Algoritmo predictivo → Historial
 - n muestras



Analizador de carga

Este módulo está enfocado en analizar el comportamiento del operador en el momento



Analizador de carga

Este módulo está enfocado en analizar el comportamiento del operador en el momento

Umbrales basados en el rendimiento del operador

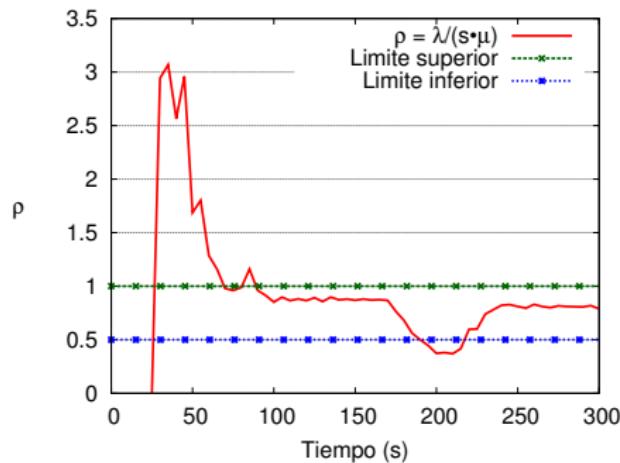
Diseño del modelo elástico

Algoritmo reactivo



- Análisis del estado del operador → Ventana de tiempo T_r
- Tasa de rendimiento ρ

$\rho > 1$	Inestable
$1 \geq \rho \geq 0.5$	Estable
$\rho < 0.5$	Ocioso





Predictor de carga

Este módulo está enfocado en predecir el estado del operador basado en el comportamiento de su historia según las muestras obtenidas por el monitor



Predictor de carga

Este módulo está enfocado en predecir el estado del operador basado en el comportamiento de su historia según las muestras obtenidas por el monitor

Cadenas de Markov



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

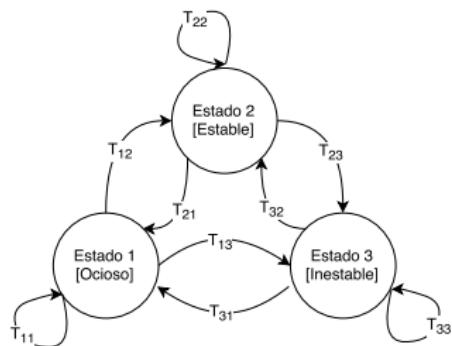
Diseño del modelo elástico

Algoritmo predictivo



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

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





- Si la cadena es irreducible y sus estados son aperiódicos y positivos recurrentes
- Ecuación de Chapman-Kolmogórov → Distribución Estacionaria
 - La probabilidad que la cadena pueda estar en un estado i en un futuro lejano

$$[\Pi_1 \quad \Pi_2 \quad \Pi_3]_{(t+1)}$$

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



- Administración de la cantidad de réplicas requeridas de cada operador
- Recursos disponibles de la máquina
- Según la ventana de tiempo realiza un tipo análisis
 - $T_r \rightarrow$ Algoritmo reactivo
 - β alertas consecutivas \rightarrow Modificar la cantidad de réplicas ω
 - $T_p \rightarrow$ Algoritmo predictivo
 - Menor frecuencia
 - Mayor cómputo
 - Modifica una mayor cantidad de réplicas $\rightarrow \theta$



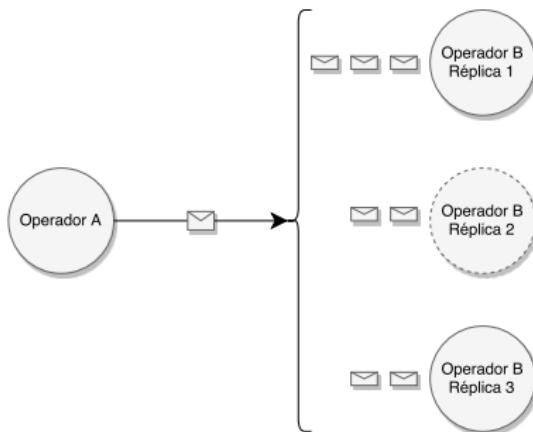
1. Introducción
2. Balance de carga
3. Solución propuesta
4. Diseño del modelo elástico
5. Experiments y evaluación
6. Conclusiones

Experimentos y evaluación

Implementación del sistema



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

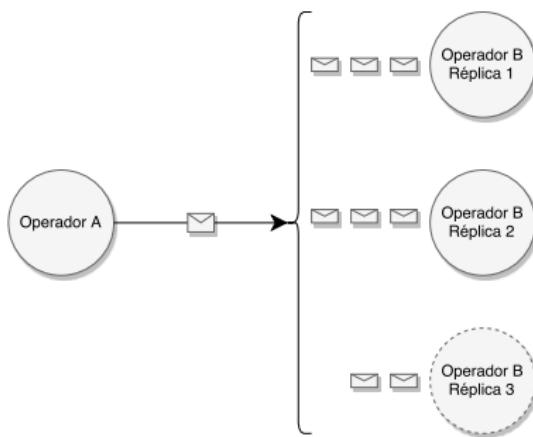


Experimentos y evaluación

Implementación del sistema



- SPS S4 → 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

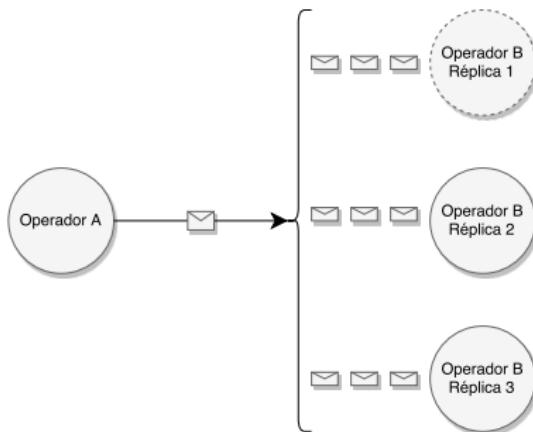


Experimentos y evaluación

Implementación del sistema



- SPS S4 → 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



Experimentos y evaluación

Diseño de los experimentos



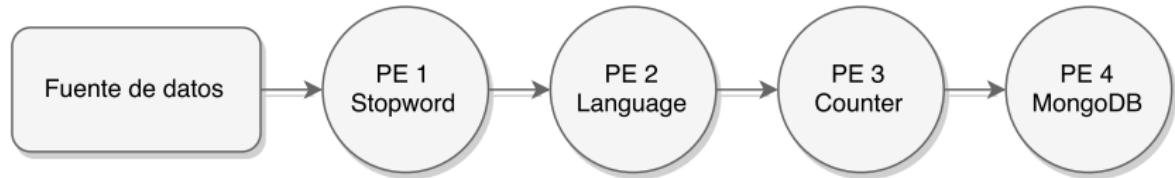
- Dos tipo de aplicaciones
 - Aplicación funcional
 - Aplicación sintética
- Generación de stream
 - 4.5 millones de tweets
 - 27-28 de Febrero y 1-2 de Marzo de 2010
 - Inglés, español y portugués
 - Interacción entre usuarios durante el terremoto del 27 de Febrero en Chile

Experimentos y evaluación

Diseño de los experimentos



- Aplicación funcional: Análisis de *tweets* en escenarios de desastres naturales
 - Validación del modelo dado un escenario aplicado

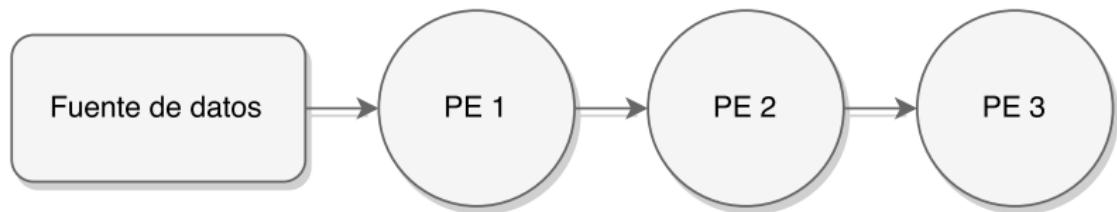


Experimentos y evaluación

Diseño de los experimentos



- Aplicación sintética
 - Costos del uso del monitor



- Período de tiempo que duerme la hebra asignada al PE
 - Basadas del promedio de ejecución de operadores

PE	Tiempo (ms)
1	20
2	30
3	15

Experimentos y evaluación

Evaluación



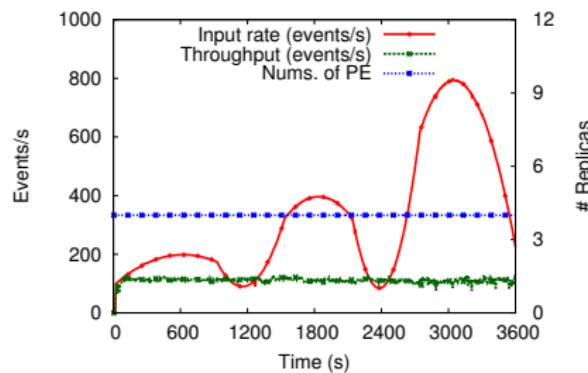
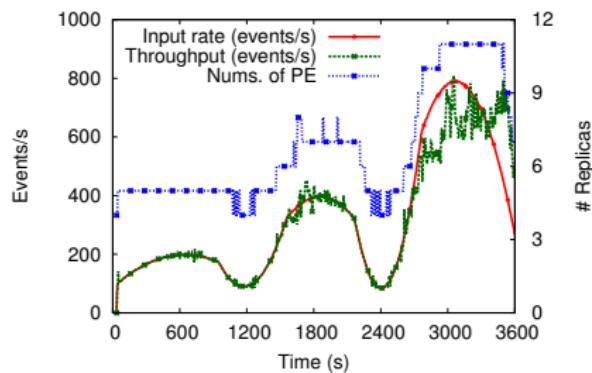
- Para la ejecución de todos los experimentos se ha utilizado una máquina con un Intel Xeon CPU E5-2650 v2 de 2.60 GHz, 32 GB de RAM y SO Ubuntu 14.04.2 LTS
- Para la evaluación de la primera aplicación se han realizado dos experimentos con distintos tiempos de ejecución:
 - En el primer experimento es 60 minutos
 - Y en el segundo experimento es 10 minutos
- Para la evaluación de la segunda aplicación se ha realizado un experimento
 - Envío constante de 100 eventos/s
 - Tiempo de ejecución de 15 minutos
- Cada uno de los experimentos se prueba con y sin modelo elástico

Experimentos y evaluación

Aplicación funcional - Experimento 1 - Rendimiento y cantidad de réplicas



- 96 eventos/segundo con uso del modelo vs 16 eventos/segundo sin uso del modelo
- Mejora de 5 veces más eventos/segundo

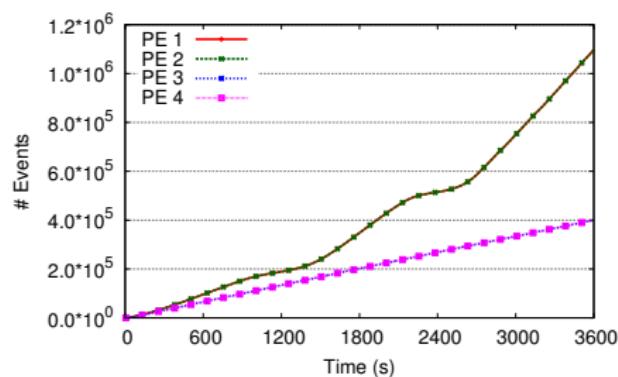
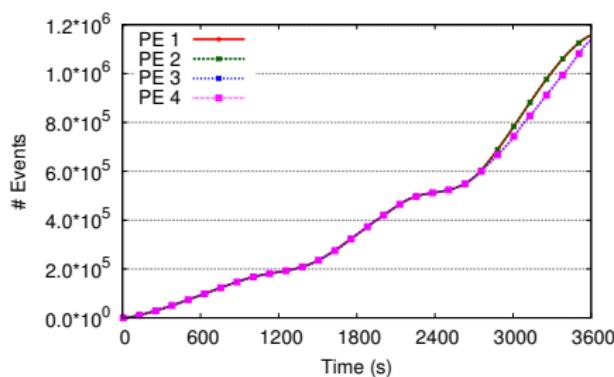


Experimentos y evaluación

Aplicación funcional - Experimento 1 - Cantidad total de eventos procesados



- 1.139.537 eventos procesados con uso del modelo vs 467.466 eventos procesados sin uso del modelo
- Mejora de 2 veces la cantidad de eventos procesados

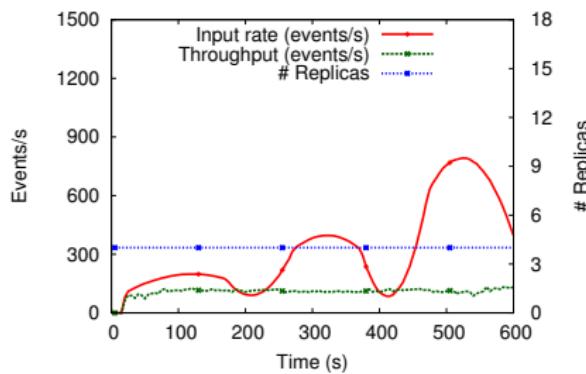
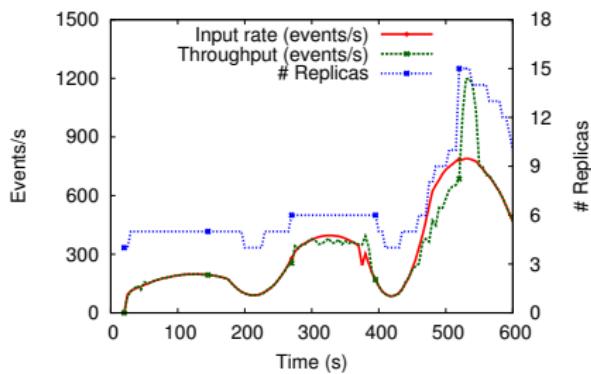


Experimentos y evaluación

Aplicación funcional - Experimento 2 - Rendimiento y cantidad de réplicas



- 72 eventos/segundo con uso del modelo vs 19 eventos/segundo sin uso del modelo
- Mejora de 3 veces más eventos/segundo

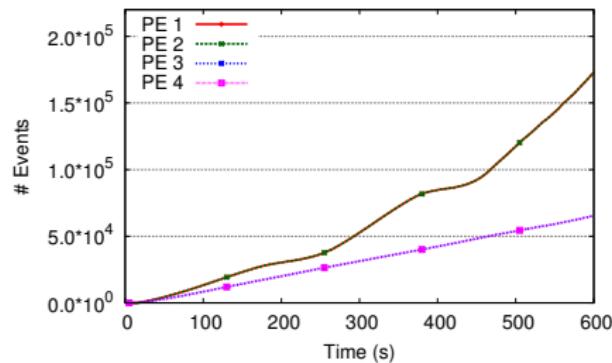
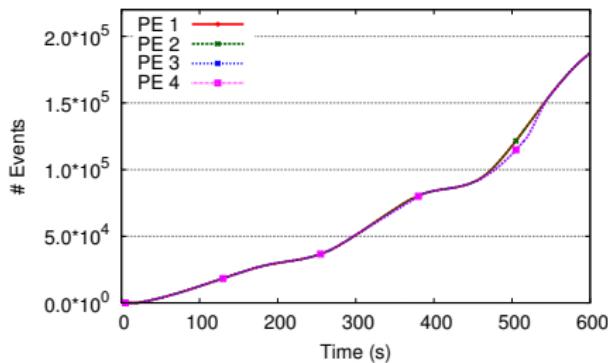


Experimentos y evaluación

Aplicación funcional - Experimento 2 - Cantidad total de eventos procesados



- 201.751 eventos procesados con uso del modelo vs 76.502 eventos procesados sin uso del modelo
- Mejora de 2 veces la cantidad de eventos procesados

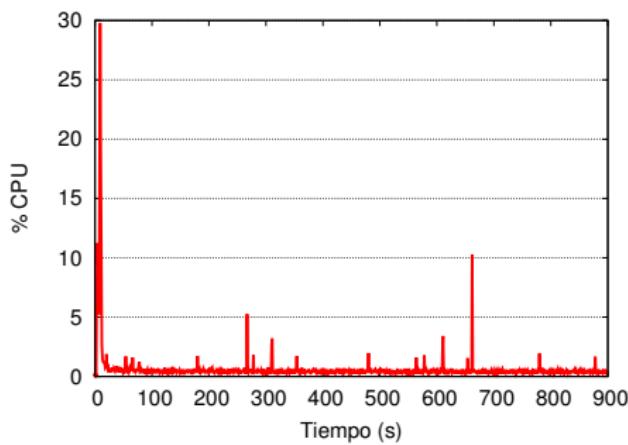
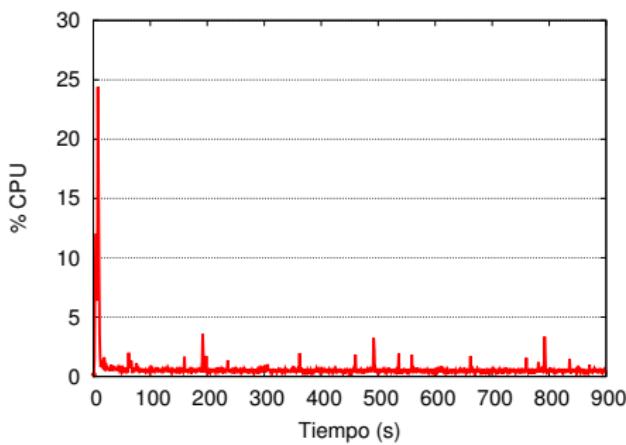


Experimentos y evaluación

Aplicación sintética - Utilización promedio de CPU



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

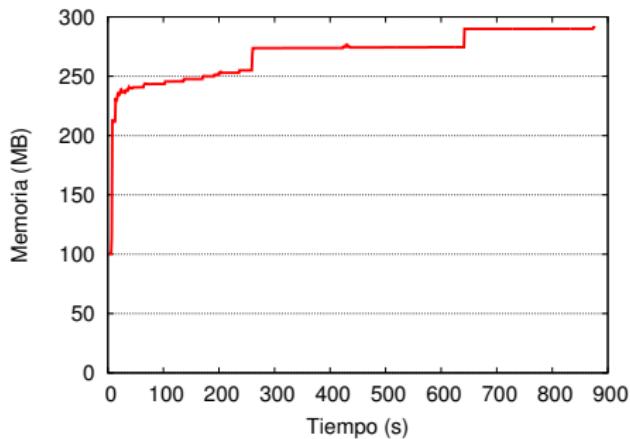
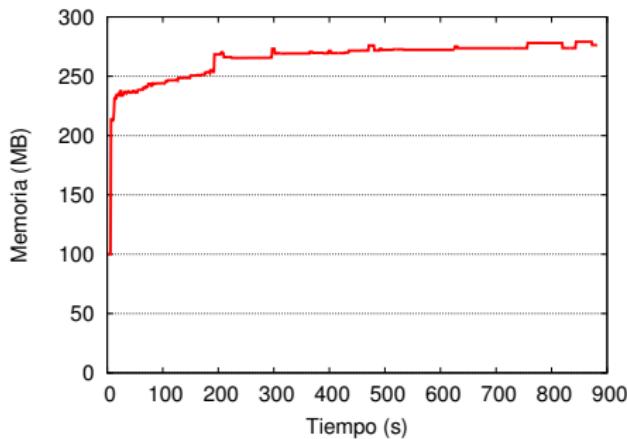


Experimentos y evaluación

Aplicación sintética - Consumo de memoria RAM



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

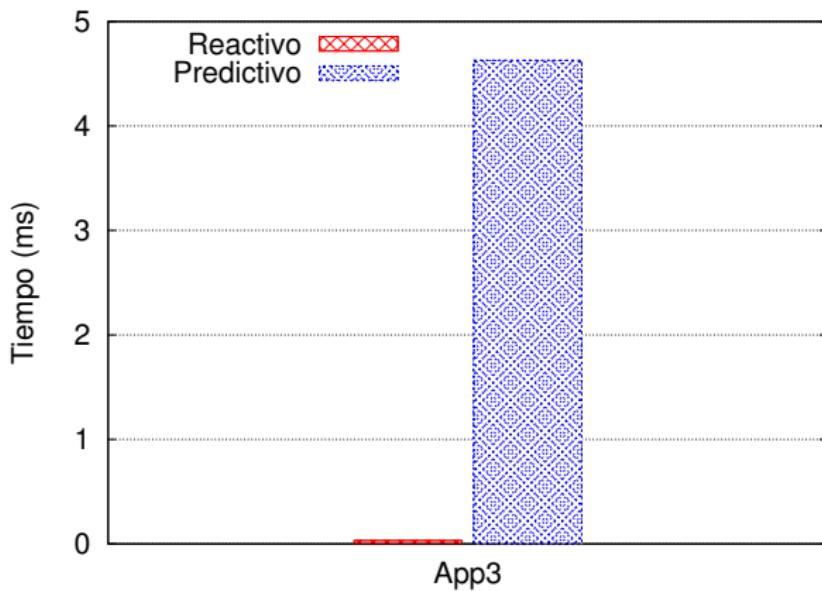


Experimentos y evaluación

Aplicación sintética - Tiempo de ejecución de cada algoritmo



- Algoritmo reactivo: 0.03 ms
- Algoritmo predictivo: 4.63 ms

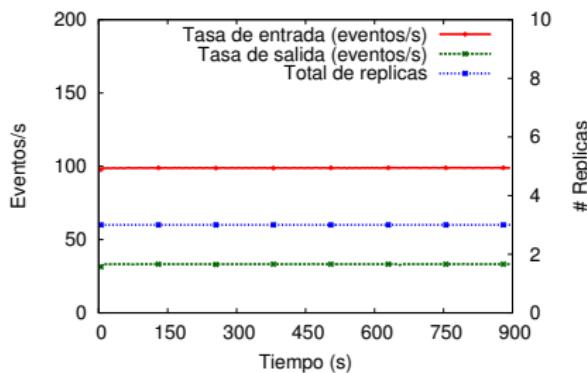
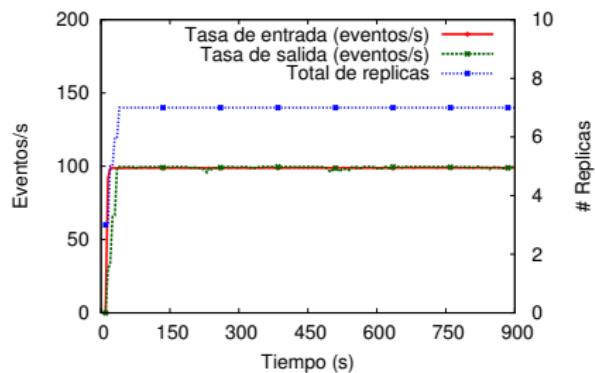


Experimentos y evaluación

Aplicación sintética - Rendimiento y cantidad de réplicas



- 97 eventos/segundo con uso del modelo vs 33 eventos/segundo sin uso del modelo
- Mejora de 2 veces más eventos/segundo

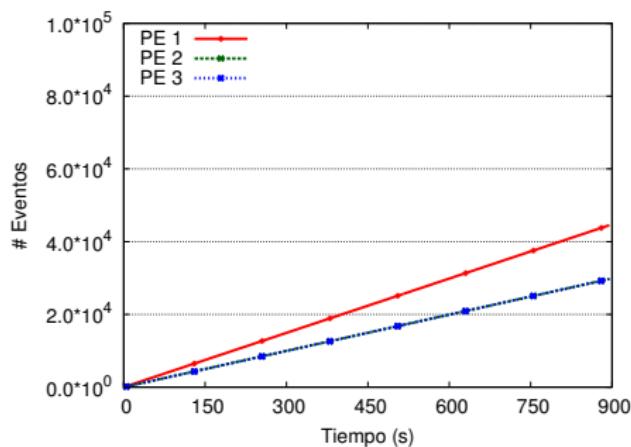
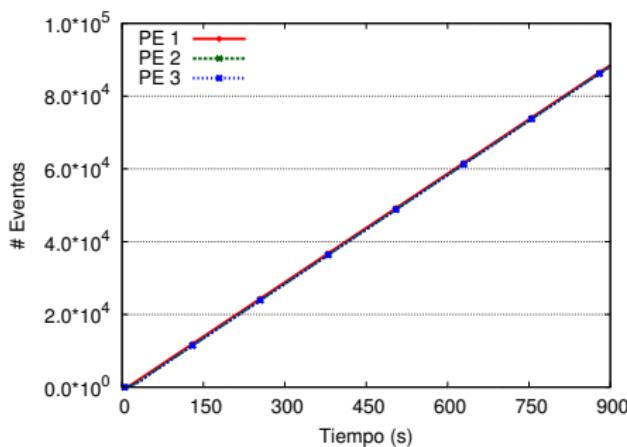


Experimentos y evaluación

Aplicación sintética - Cantidad total de eventos procesados



- 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





1. Introducción
2. Balance de carga
3. Solución propuesta
4. Diseño del modelo elástico
5. Experimentos y evaluación
6. Conclusiones

Conclusiones

Detalles de la contribución



- Diseño e implementación de un modelo elástico
 - Algoritmo reactivo
 - Algoritmo predictivo
 - Módulo de administración de réplicas
- Construcción de distintos escenarios → Validación del modelo diseñado
- Evaluado y analizado el rendimiento del sistema con y sin uso del modelo elástico



- 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

Conclusiones

Trabajo a futuro



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

Preguntas





-  [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.



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.



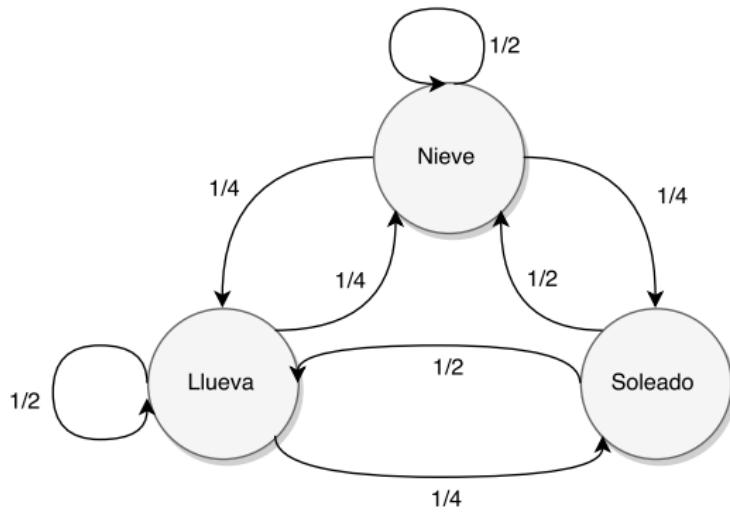
-  Ishii, A. and Suzumura, T. (2011).
Elastic stream computing with clouds.
In *IEEE International Conference on Cloud Computing, CLOUD 2011, Washington, DC, USA, 4-9 July, 2011*, pages 195–202.
-  Nguyen, H., Shen, Z., Gu, X., Subbiah, S., and Wilkes, J. (2013).
AGILE: elastic distributed resource scaling 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.



-  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.

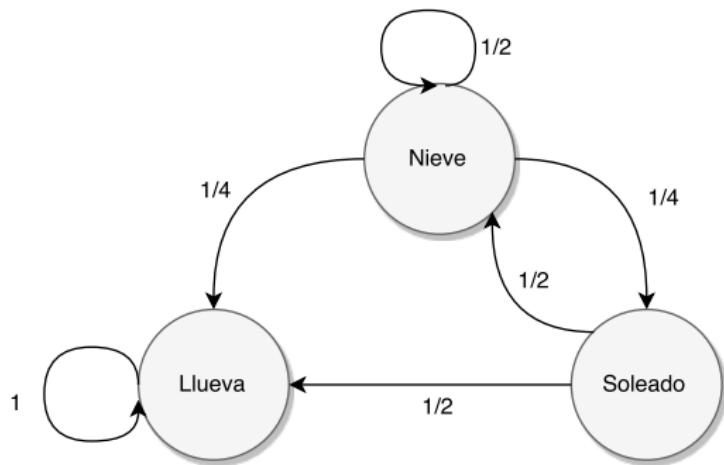


- Ejemplo de Cadena de Markov



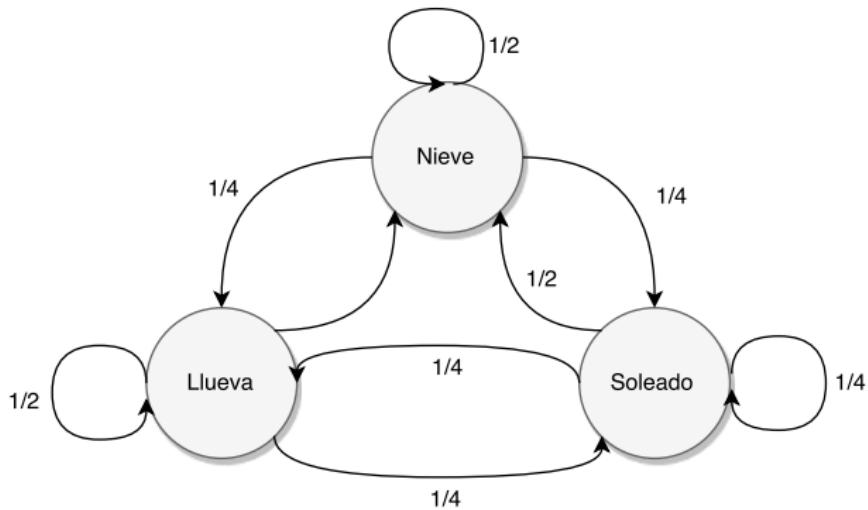


- Ejemplo de Cadena de Markov reductible





- Ejemplo de Cadena de Markov aperiódica



Experimentos y evaluación

Parámetros



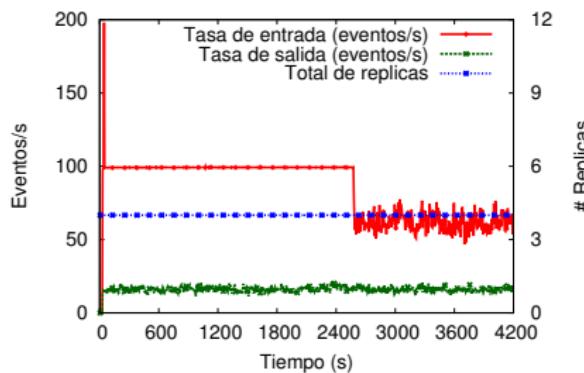
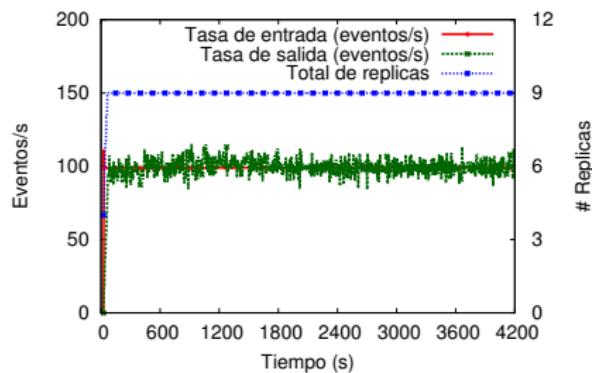
- Monitor de carga
 - $T_m = 1\text{seg}$
- Algoritmo reactivo
 - $\omega = 1$ réplica
- Algoritmo predictivo
 - $\theta = 5$ réplicas
 - $n = 100$ muestras
 - $\beta = 0.25$
- Administrador de réplicas
 - Ventana de tiempo de ejecución
 - $T_r = 5\text{seg}$
 - $T_p = 100\text{seg}$
 - $\beta = 2$ alertas consecutivas

Experimentos y evaluación

Aplicación 1 - Constante - Rendimiento y cantidad de réplicas



- 96 eventos/segundo con uso del modelo vs 16 eventos/segundo sin uso del modelo
- Mejora de 5 veces más eventos/segundo

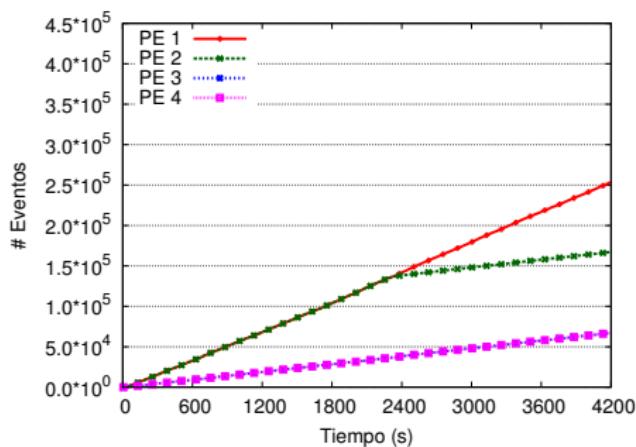
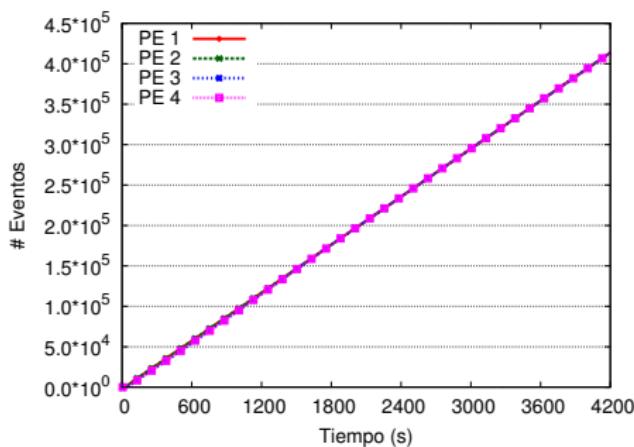


Experimentos y evaluación

Aplicación 1 - Constante - Cantidad total de eventos procesados



- 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

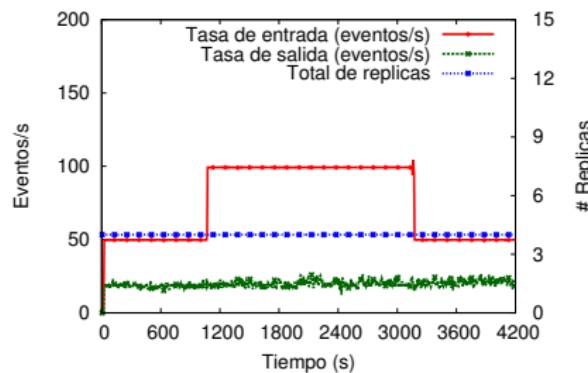
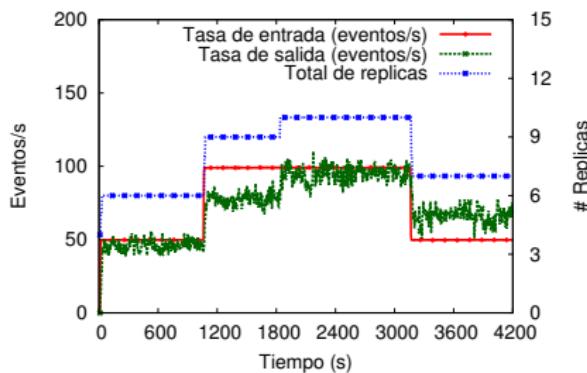


Experimentos y evaluación

Aplicación 1 - Variable - Rendimiento y cantidad de réplicas



- 72 eventos/segundo con uso del modelo vs 19 eventos/segundo sin uso del modelo
- Mejora de 3 veces más eventos/segundo

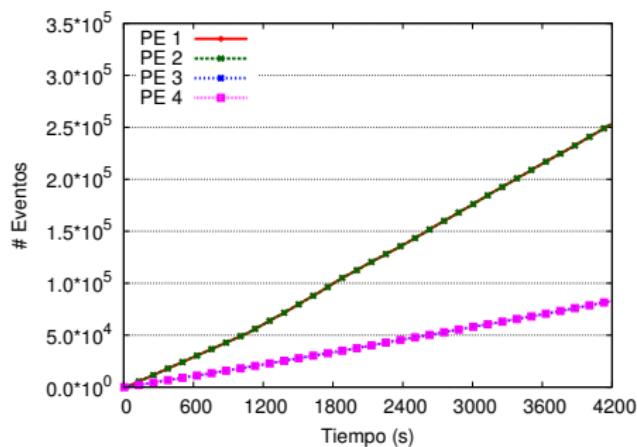
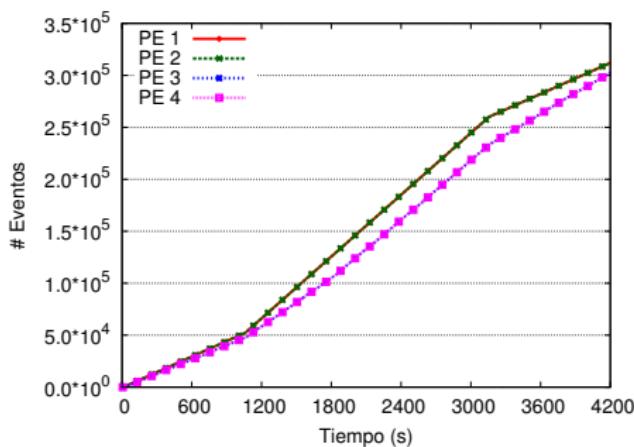


Experimentos y evaluación

Aplicación 1 - Variable - Cantidad total de eventos procesados

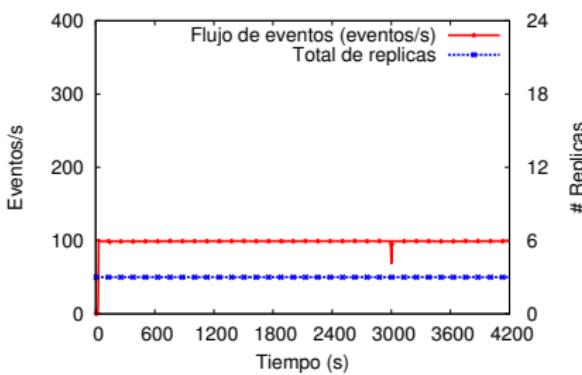
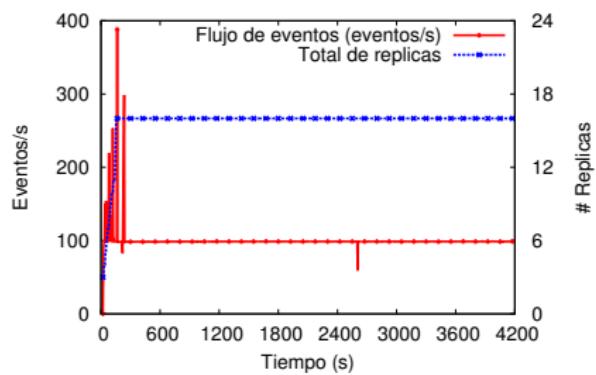


- 303.156 eventos procesados con uso del modelo vs 82.770 eventos procesados sin uso del modelo
- Mejora de 3 veces la cantidad de eventos procesados



Experimentos y evaluación

Aplicación 2 - Constante - Rendimiento y cantidad de réplicas

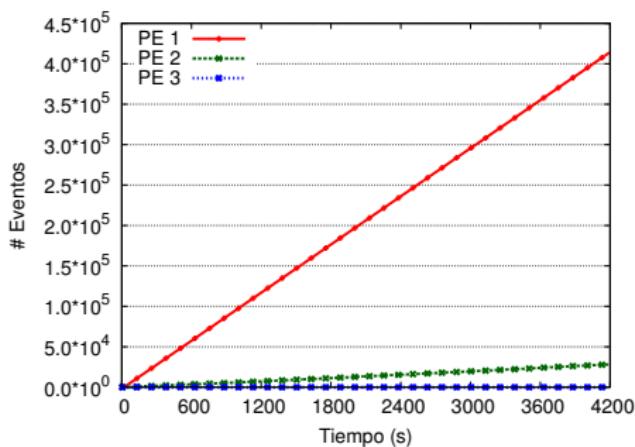
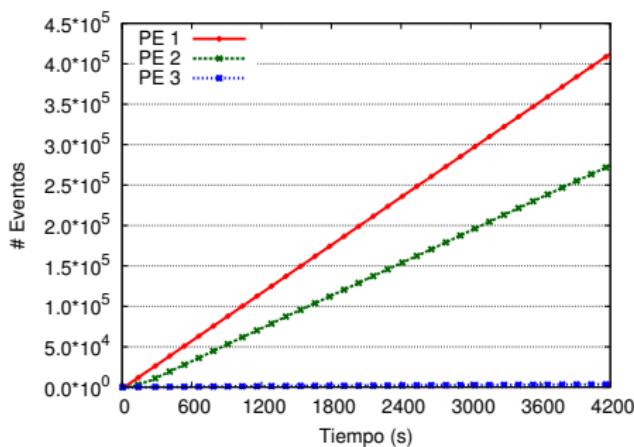


Experimentos y evaluación

Aplicación 2 - Constante - Cantidad total de eventos procesados

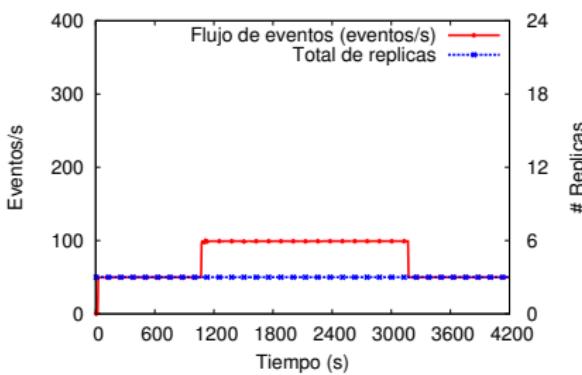
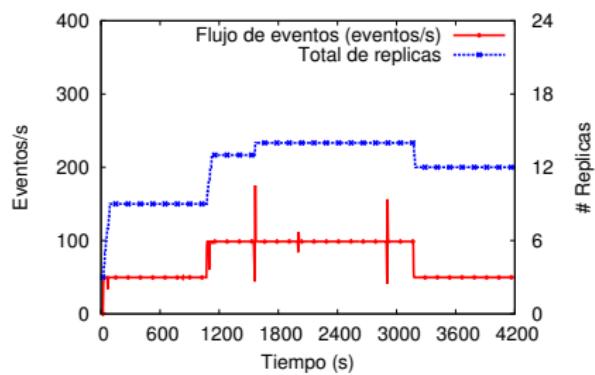


- 275.290 eventos procesados con uso del modelo vs 28.152 eventos procesados sin uso del modelo
- Mejora de 9 veces la cantidad de eventos procesados



Experimentos y evaluación

Aplicación 2 - Variable - Rendimiento y cantidad de réplicas

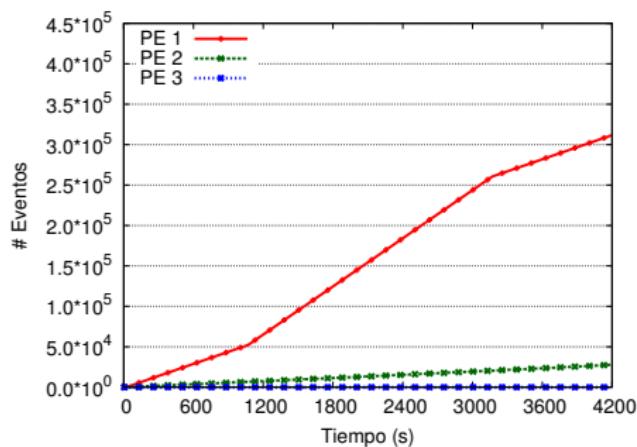
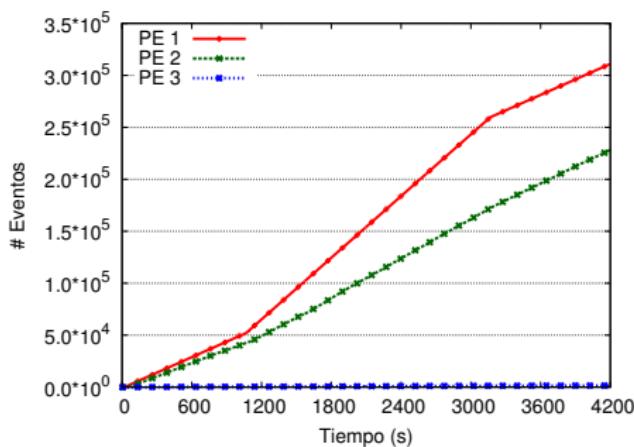


Experimentos y evaluación

Aplicación 2 - Variable - Cantidad total de eventos procesados

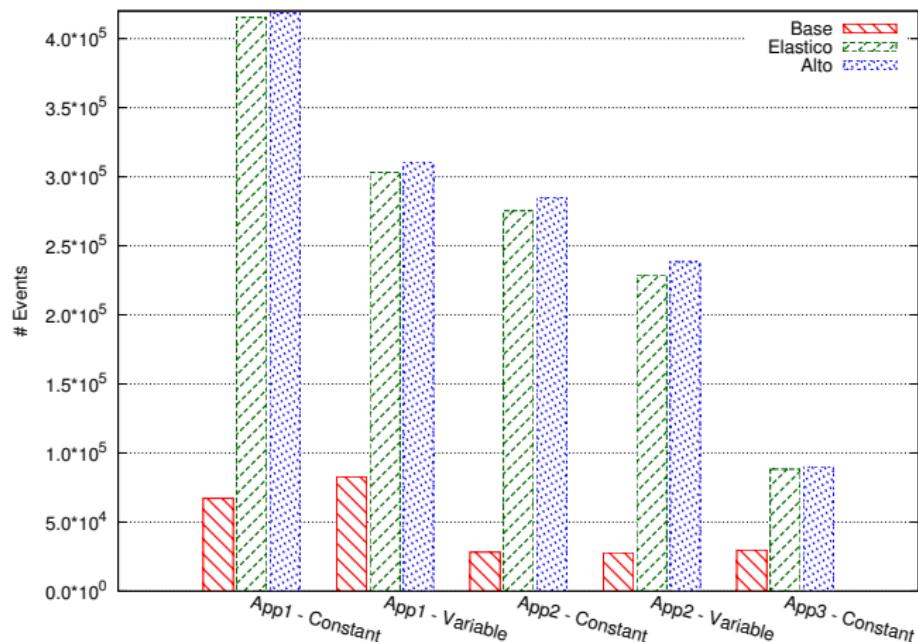


- 228.942 eventos procesados con uso del modelo vs 27.751 eventos procesados sin uso del modelo
- Mejora de 8 veces la cantidad de eventos procesados



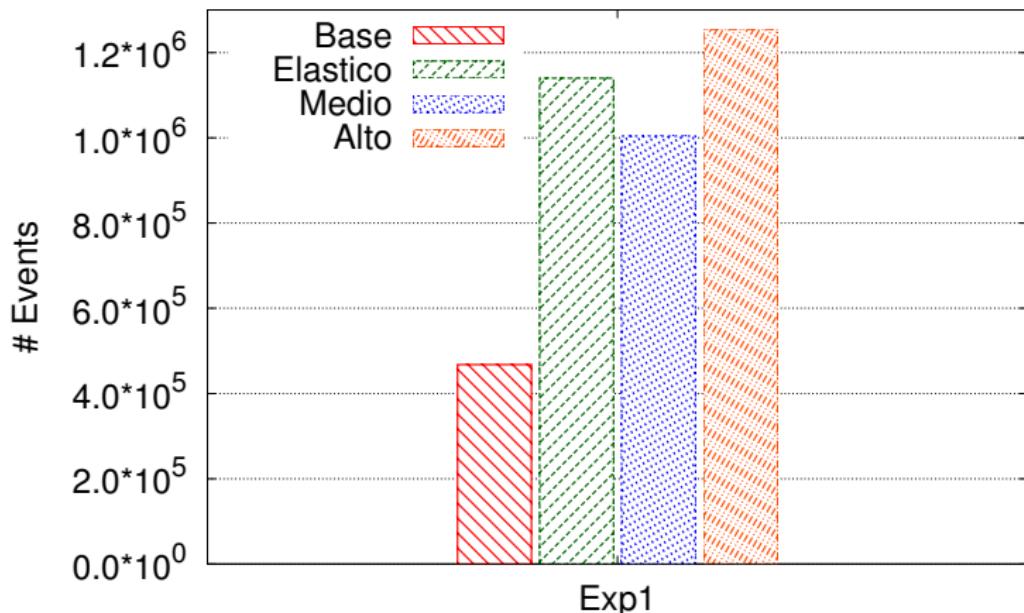
Experimentos y evaluación

Cantidad de eventos procesados



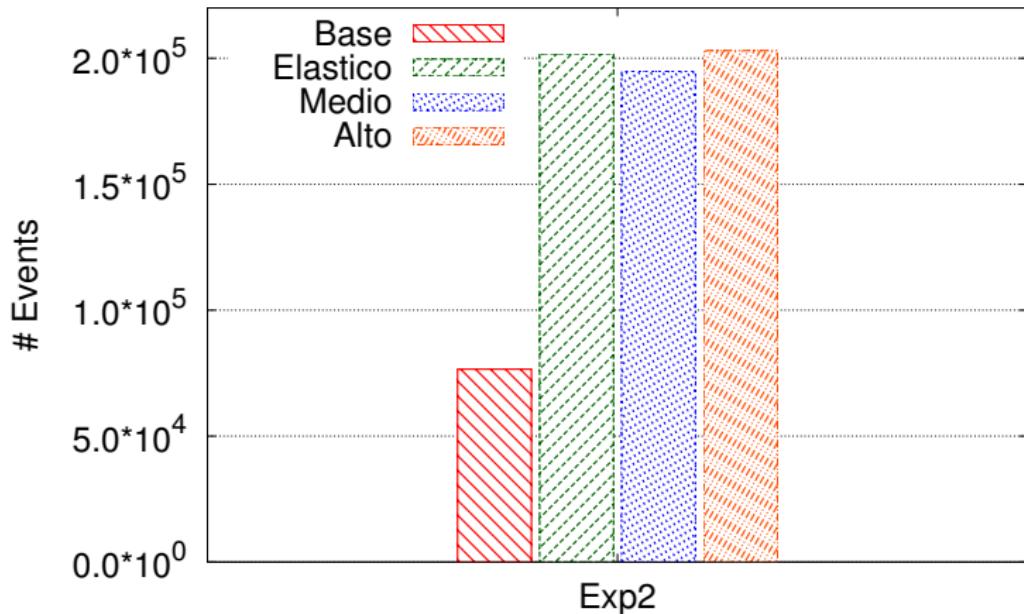
Experimentos y evaluación

Aplicación funcional - Experimento 1 - Cantidad de eventos procesados



Experimentos y evaluación

Aplicación funcional - Experimento 2 - Cantidad de eventos procesados



Twitter Stream



Twitter Stream in the five days

