**UNIVERSIDAD DE SANTIAGO DE CHILE**
**FACULTAD DE INGENIERÍA**
**Departamento de Ingeniería Informática**


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

**Daniel Pedro Pablo Wladdimiro Cottet**

Profesor guía: Nicolás Andrés Hidalgo Castillo
Profesor co-guía: Erika Susana Rosa Olivos

Tesis de grado presentada en conformidad a los requisitos para obtener el grado de Magíster en Ingeniería Informática

Santiago – Chile
2015

*Gracias a la vida que me ha dado tanto.*
*Me ha dado la risa y me ha dado el llanto.*
*Así yo distingo dicha de quebranto,*
*los dos materiales que forman mi canto*
*y el canto de ustedes que es el mismo canto,*
*y el canto de todos, que es mi propio canto.*

# AGRADECIMIENTOS

Me cuesta encontrar las palabras para expresar lo que siento, se cierra un ciclo en mi vida, y con ello una larga travesía de mucho esfuerzo y trabajo, donde espero que de inicio a muchos más. Recibí apoyo, cariño y paciencia de diferentes personas, las cuales subían y bajaban de este excéntrico tren en cada parada que hacíamos, y aunque a veces entre paradas se hacía un viaje muy corto, habían otras veces que se hacían unos viajes eternos, y no terminaba de distinguir realmente bien dónde iba, pero eso daba igual, porque siempre en cada trayecto aprendía y recibía algo de alguna persona, y eso me hizo ir creciendo con el tiempo, lo cual estoy eternamente agradecido. De antemano pido disculpas si me he olvidado de alguien, no fue mi intención, y espero que pueda agradecerlo en un tiempo futuro.

Aunque uno a veces siente que todo cambia; la gente, el barrio, la universidad, la sociedad, el trabajo, hay sentimientos y pasiones que nunca cambian, y son esas que te hacen vibrar y llenarte el corazón por estar ahí. Sin duda alguna, Izquierda Libertaria es una mis grandes pasiones, donde a través del FeL me brindó una escuela de lucha para poder construir un pueblo digno y soberano. No saben como agradezco estar ahí y conocer a mis amigas y amigos que además son compañeras y compañeros de lucha: Thomi, Neto, Cachorro, Sussan, Andrés, Zarri, Cata, Pato, Cristián, Tuto, Nati, Joaco, Fofi, de corazón gracias por todo su apoyo incondicional. Y más todavía agradezco al FeL porque aquí conocí a mi amada polola, quien no sólo le agradezco su apoyo, sino también por su cariño, alegría y paciencia, y sin duda todo sería muy distinto sin ti, no sabes lo agradecido y feliz que soy de estar con vos. Te amo Cami.

Cuando uno va viajando, vas creciendo, vas madurando, te vas dando cuenta que puedes ir resolviendo problemas que antes se hacían imposibles, van dándose nuevas herramientas, nuevas habilidades. Por lo mismo es que estoy completamente agradecido de la oportunidad que me brindaron mis queridos profesores guías Erika y Nicolás, porque confiaron en mí y se la jugaron por sacar este trabajo, y no sólo eso, de abrirme puertas para poder sustentarme y tener un porvenir más tranquilo. De verdad no saben lo orgulloso que estoy de tenerlos como profesores, no sólo me han ayudado a formarme como profesional, sino también como persona y eso se los agradezco desde el fondo de mi corazón. También agradezco a Pamela, por su apoyo y entrega incondiconal, y la profesora Carolina y el profesor Mauricio, que gracias a CITIAPS me sentí en mi segundo hogar, y siento que mucha de las cosas que aprendí hoy en día es gracias a esa pequeña salita de gran corazón. Y por lo mismo, no puedo olvidar a mi compañero de trabajo, Pablo, siempre me acuerdo como nos conocimos en ese primer día de las clases de Magíster, nunca creí que gracias a ese saludo pudiéramos tener este viaje, donde salió un paper, un proyecto y una bonita amistad, de verdad gracias Pablo, sos un hermano para mí. Gracias a todos los que pasaron y están en CITIAPS, Álvaro, Jeff, Diego, Farisori, no me olvidaré de ustedes, porque sin duda me dejaron marcado y me ayudaron a formarme como profesional y persona. También a mis amigas y amigos de la Usach, Miguel, Gabo, Clau, Karla, Jorge, los quiero caleta y gracias por todo, porque fueron un pilar en mi vida universitaria, donde no los olvidaré, porque todo lo que viví con ustedes fue una de mis grandes alegrías.

Pero a veces uno se queda dormido, recuerdas el pasado, y te acuerda que eso fue lo que te formó y lo que explica el cómo eres hoy en día. Nunca los voy a olvidar, para mi son mi fuerza, mis ganas, mi apañe, quizá todos estamos en nuestras vidas, pero los tengo mas presente que nunca, José, Eduardo, Rubén y Felipe, los quiero demasiado. Y como te voy a olvidar, si sos mi hermano de leche, Simón, no sólo te tengo que agradecer, te debería hacer una oda por lo grande que has sido conmigo. Tantas cosas que pasamos, tanto que vivimos, tantos recuerdos, y como fuimos creciendo, donde sea que estés tengo claro que puedo contar contigo, y estoy seguro que tú sientes lo mismo de mí.

Y alguien tuvo que crear la primera estación del viaje, alguien tuvo que armar ese tren, y es algo que aunque suba y bajen mil personas, pasen mil años, nunca podrás borrarlo, y me alegro que sea así, porque es lo más valioso que uno tiene, la familia. Como no quererlas, si me mimaron, me criaron, me abrazaron, me apoyaron, y eso lo agradezco, a ti Amandi, a ti Sofi y a ti Mamá, las amo. Obvio, no me voy a olvidar de ti, gracias por todo tu apoyo Papá, siento que mucho de lo que aprendí fue gracias a ti, y me alegro haber tenido esa oportunidad.

Agradezco también al PMI-USA 1204 y al proyecto DICYT-USACH 061419HC por darme la tranquilidad de poder trabajar en este largo proyecto.

# TABLA DE CONTENIDO

# ÍNDICE DE TABLAS

# ÍNDICE DE ILUSTRACIONES

# ÍNDICE DE ALGORITMOS

# RESUMEN

En el mundo actual de la información, grandes cantidades de datos son generados cada segundo desde las más diversas fuentes: redes sociales, redes de sensores, buscadores Web, entre otros. Extraer información de dichos datos muchas veces requiere que este procesamiento sea llevado a cabo en tiempo real, debido a que el análisis que se debe realizar depende de la temporalidad en que son generadas los eventos. Para lograr procesar grandes cantidades de datos con estas restricciones, existen sistemas especializados llamados sistemas de procesamiento de *stream* (SPS), los cuales pueden procesar en tiempo real los datos que van llegando por una o más fuentes de datos. Estos sistemas están basados en grafos, cuyos vértices realizan operaciones sobre un flujo de datos reflejado por las aristas del grafo. La topología del grafo le brinda flexibilidad al SPS para generar diversas aplicaciones de procesamiento, sin embargo, dicha topología es estática una vez el sistema se ejecuta. Dado este problema, este trabajo se plantea un modelo elástico que sea capaz de adaptar la topología del grafo a las condiciones del tráfico existente. Para esto se ha diseñado un algoritmo reactivo, usando la técnica de fisión, y otro predictivo, usando cadena de Márkov, ambas técnicas permiten estimar la carga de los operadores, y adaptar el grafo acorde a lo indicado por estos. Dicha modificación consiste en incrementar o disminuir la cantidad de réplicas de un operador según su nivel de carga. Los resultados obtenidos de los experimentos realizados en el SPS S4, muestran una mejora de hasta nueve veces más eventos procesados, con un costo asociado a un aumento de $0,01\,\%$ del uso de la CPU, pero una disminución de un $1,5\,\%$ en el consumo de memoria RAM.

**Palabras Claves**: SPS; Elasticidad; Distribución de carga; Balance de carga; Algoritmos reactivos; Fisión; Algoritmos predictivos; Cadena de Márkov

# ABSTRACT

Nowadays, information generated by the Internet's interactions is growing exponentially, creating massive and continuous flows of events from the most diverse sources. These interactions contain valuable information for domains such as government, commerce, and banks, among others. Extracting information from such data requires powerful processing tools to cope with the high-velocity and high-volume stream of events with near real-time results. Specially-designed distributed processing engines build a graph-based topology of a static number of processing operators creating bottlenecks and load balance problems when processing dynamic flows of events. In this work we propose a self-adaptive processing graph that provides elasticity and scalability, increasing or decreasing automatically the number of processing operators to improve performance and resource utilization. Our solution uses a model that monitors, analyzes and changes the topology of the graph with a control algorithm that is both reactive and proactive to the flow of events. We have compared our solution with a baseline approach and results show that our system improves performance in terms of the number of processed events at a very low cost.

**Keywords**: SPS; Elastic; Load balancing; Reactive Algorithm; Fision; Predictive Algorithm; Markov chain

# CAPÍTULO 1. INTRODUCCIÓN

## 1.1 ANTECEDENTES Y MOTIVACIÓN

La gran contribución de información en la Internet se ha debido al origen de la Web 2.0 donde ésta se caracteriza por la participación activa del usuario, como en blogs, redes sociales u otras aplicaciones Web (Oberhelman, 2007).

Con el paso del tiempo, más y más información es generada por distintas interacciones generadas por los usuarios. Por lo que analizar o extraer esta información no es una tarea fácil, más aún cuando muchas de estas interacciones deben ser analizadas en tiempo real, dada su dependencia temporal. Por esta última característica es que sistemas tradicionales de procesamiento basados en MapReduce (Lin & Dyer, 2010) o *bash processing* (Hawwash & Nasraoui, 2014) no son ideales para el análisis de esta información.

Es así como con el tiempo se han ido creando distintos sistemas de procesamiento capaces de lidiar con las restricciones de temporalidad, debido al interesante funcionamiento que poseen, las que se caracterizan por ser capaces de procesar grandes flujos de datos en tiempo real (Chen & Zhang, 2014). El requisito de procesar información en tiempo real surge por la necesidad de los usuarios en obtener respuestas rápidas y actualizadas que le permitan tomar decisiones en períodos cortos de tiempo. Dentro de los ejemplos existentes se encuentran: análisis de sentimientos de los mensajes de usuarios, análisis de los precios de la bolsa de valores, recopilación de información en caso de emergencia, entre otros. Este tipo de aplicaciones son necesarias para sus usuarios, debido que proveen de información actualizada que permite mejorar entre otros casos la toma de decisiones (Wenzel, 2014).

Un ejemplo de esto son las aplicaciones que analizan redes sociales en caso de un desastre natural, donde grandes cantidades de información son generadas, y se requiere procesar esta información lo más cercano al tiempo real para obtener información que permita un trabajo de recuperación más eficiente dado el suceso (Andrade et al., 2014). De esta manera, se puede construir un sistema que procese los datos realizando un análisis de la percepción de la gente, búsqueda de personas desaparecidas o focos de necesidad. Con esta información, se puede establecer sectores críticos, facilitar la búsqueda de personas, distribución de alertas, o detección de necesidades, lo cual es crucial para tomar decisiones en esos momentos.

Por otra parte, también son utilizados para llevar a cabo predicciones en la bolsa de comercio. De esta manera, se crean sistemas de procesamiento que apliquen modelos matemáticos, los cuales permiten predecir el comportamiento para el siguiente día en el mercado. Con estos sistemas, la ganancia que existe por parte de las personas interesadas puede aumentar

considerablemente, por lo que ha generando un alto interés en el desarrollo e investigación en esta área.

También se aplica en casos de seguridad en redes, donde se permite realizar un monitoreo de las actividades ocurridas en la red. Como la información es procesada en tiempo real, ayuda a detectar a tiempo las posibles acciones de usuarios maliciosos. Dentro de las aplicaciones que existen sobre este tema, son el análisis de los logs de la red cerrada, donde con esta información se puede verificar si existe algún *bug*, error o anomalía, además de ver si existe algún intruso o violación al sistema.

Para dar soporte a estas aplicaciones existen los SPS (Sistemas de Procesamiento de *Stream*) tales como S4 (Neumeyer et al., 2010), Storm (Storm, 2014), Samza (Samza, 2014), entre otros. El paradigma de procesamiento de estos sistemas se basa en grafos, donde los vértices son operaciones realizadas al flujo de datos, representadas por las distintas aristas. Para la generación de una aplicación, el usuario debe diseñar una topología de procesamiento compuesto por las tareas u operaciones deseadas. Cada grafo o topología tiene un *input* desde una fuente de datos, y un *output* del flujo de salida proveniente del último operador. Aunque poseen bastante flexibilidad para la creación de diversas aplicaciones, por la facilidad de crear distintas topologías, no lo tiene para adaptarse con el tiempo a las condiciones del tráfico entrante cuando se encuentran en funcionamiento, esto debido a que las topologías de procesamiento generadas son estáticas en tiempo de ejecución. Dada la naturaleza dinámica de las interacciones, pueden surgir problemas de distribución de carga en la topología asociada a la aplicación.

El problema de sobrecarga conlleva a una baja en el rendimiento, produciendo una pérdida de recursos, tiempo e información. Abordar este problema es crítico, puesto que al realizar una optimización en el sistema, implica un aumento en la cantidad de datos procesados, y una mayor precisión en los resultados por parte de la aplicación.

Lo anterior lo podemos entender de mejor manera con el siguiente ejemplo: se posee un tiempo $t$ para procesar $n$ datos. Si se aumenta la cantidad de datos procesados, se tiene que en el mismo tiempo $t$ se procesan una cantidad $n + m$ de datos, donde $m$ son los datos adicionales a analizar debido a la mejora del rendimiento. Como existe un aumento en la cantidad de datos procesados, la información obtenida puede ser más precisa, dado que se posee una mayor cantidad de datos. Por ejemplo, al procesar una mayor cantidad de transacciones en la bolsa de comercio, se puede poseer una predicción más precisa de cómo se comporta la bolsa a futuro. Desde otro punto de vista, se efectúa una mejora en los recursos utilizados, habiendo una disminución de los recursos ociosos.

## 1.2 DESCRIPCIÓN DEL PROBLEMA

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

## 1.3 SOLUCIÓN PROPUESTA

La solución propuesta consiste en un modelo elástico de replicación de operadores para los sistemas de procesamiento de *stream*, el cual permite adaptar el grafo de procesamiento a las variaciones del tráfico. Para esto se ha implementado cuatro módulos que componen la estructura del modelo elástico: monitor de carga, analizador de carga, predictor de carga y administrador de réplicas.

El monitor de carga está encargado de recuperar el nivel de carga de cada uno de los operadores. Esta información es entregada a los módulos analizador y predictor de carga, los cuales están encargados de medir el nivel de carga del operador, y según eso modificar la cantidad de réplicas necesarias. Cada uno de estos módulos trabaja de forma independiente y poseen distintos enfoques: reactivo y predictivo.

El analizador de carga aplica un enfoque reactivo, el cual analiza el tráfico de los operadores en el tiempo actual y cuantifica su carga. El estado de la carga de cada operador depende de un umbral, por lo que según éste se solicita al administrador de réplica incrementar o disminuir la cantidad de réplicas del operador.

El predictor de carga aplica un enfoque predictivo, el cual analiza la carga de los distintos operadores en una ventana de tiempo y predice la carga para la siguiente ventana. Con esta información el administrador de réplicas determina la mejor configuración de los operadores para dicho período.

El administrador de réplicas por su parte se alimenta de la información entregada por los dos módulos anteriores, y en base a esto, toma una decisión respecto a los recursos asignados al operador. En otras palabras, verifica cuántas réplicas son necesarias según el tamaño del tráfico.

## 1.4 OBJETIVOS Y ALCANCE DEL PROYECTO

### 1.4.1 Objetivo general

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

### 1.4.2 Objetivos específicos

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*.

### 1.4.3 Alcances

Dentro de los alcances y limitaciones que se tiene en el proyecto son:

1. La evaluación de la solución se ha implementado sobre un solo sistema de procesamiento de *stream*.
2. Los datos emitidos de la fuente de datos son homogéneos, teniendo una tasa de procesamiento constante para cada operador.
3. La distribución de flujo de datos es a nivel de operadores y no de máquinas, por lo que no se ha analizado la carga de estos últimos.

4. Los algoritmos propuestos no incluyen técnicas que garanticen el procesamiento de todo el flujo de datos.
5. En la evaluación de los algoritmos propuestos se ha considerado el costo de comunicación de manera igualitaria para todos los operadores.

## 1.5 METODOLOGÍA Y HERRAMIENTAS UTILIZADAS

### 1.5.1 Metodología

Dado el carácter de investigación del trabajo propuesto, se ha utilizado el método científico para la realización de éste. Dentro de las etapas propuesta por (Sampieri et al., 2010) están:

1. Formulación de la hipótesis: “La utilización de un modelo elástico en un sistema de procesamiento de *stream* permitirá aumentar la cantidad de eventos procesados, y con ello la precisión de los resultados”.
2. Elaboración del marco teórico: Exponer los trabajos que existen sobre problemas de carga en los operadores de SPS. Así mismo, los conceptos fundamentales de estos sistemas.
3. Seleccionar el diseño apropiado de investigación: Diseñar el modelo elástico para el problema de balance de carga a nivel lógico en un SPS, vale decir, los distintos módulos, donde se posea un algoritmo reactivo y otro predictivo. Cada diseño y ejecución de los experimentos se basan en los principios de un SPS definiendo métricas acordes para dicho modelo de procesamiento.
4. Analizar los resultados: De debe analizar los resultados según las métricas establecidas y el modelo propuesto.
5. Presentar los resultados: Elaborar el reporte de investigación y presentar los resultados en gráficos y tablas.
6. Concluir en base a los resultados de la investigación.

### 1.5.2 Herramientas de desarrollo

Para el procesamiento de *stream* se ha utilizado Apache S4 0.6.0, por lo que es necesario para su configuración Java SE Development Kit 7. Dentro esto, el lenguaje de programación de cada uno de los módulos del modelo desarrollado se ha utilizado el lenguaje de programación Java, y se ha trabajado sobre el IDE Eclipse Standard 4.4.2. De forma complementaria, se ha utilizado Texmaker 4.1 para la confección de los distintos informes requeridos y la documentación correspondiente al trabajo.

## 1.6 ORGANIZACIÓN DEL DOCUMENTO

En el presente documento se divide en seis capítulos. En este capítulo se presenta la problemática y la solución propuesta, en conjunto con los objetivos y la metodología utilizada. En el segundo capítulo se exponen los conceptos teóricos involucrados. Posteriormente, el tercer capítulo aborda los distintos enfoques y técnicas que se han brindado en la literatura para dar soluciones al problema planteado. Luego, el cuarto capítulo se describe el diseño de los algoritmos utilizados en el modelo propuesto, explicando las distintas decisiones que se han tomado para el diseño de éste. En el quinto capítulo se presentan los distintos experimentos realizados para evaluar el sistema diseñado, donde se explica su implementación y evaluación según los experimentos diseñados. Finalmente, en el sexto capítulo se exponen las respectivas conclusiones obtenidas a partir del presente trabajo.

# REFERENCIAS BIBLIOGRÁFICAS


Andrade, H., Gedik, B., & Turaga, D. (2014). *Fundamentals of Stream Processing: Application Design, Systems, and Analytics*. Cambridge University Press.

Chen, C. L. P., & Zhang, C. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Inf. Sci.*, *275*, 314–347.

Hawwash, B., & Nasraoui, O. (2014). From tweets to stories: Using stream-dashboard to weave the twitter data stream into dynamic cluster models. In *Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, BigMine 2014, New York City, USA, Agosto 24, 2014*, (pp. 182–197).

Lin, J., & Dyer, C. (2010). *Data-Intensive Text Processing with MapReduce*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Neumeyer, L., Robbins, B., Nair, A., & Kesari, A. (2010). S4: distributed stream computing platform. In *ICDMW 2010, The 10th IEEE International Conference on Data Mining Workshops, Sydney, Australia, 14 December 2010*, (pp. 170–177).

Oberhelman, D. (2007). Coming to terms with Web 2.0. *Reference Reviews*, *21*, 5–6.

Sampieri, R. H., Collado, C. F., & Lucio, P. B. (2010). Metodología de la investigación. *México: Editorial Mc Graw Hill*.

Samza, A. (2014). Samza. [Online] `http://samza.incubator.apache.org/`.

Storm (2014). Distributed and fault-tolerant realtime computation. [Online] `http://storm.incubator.apache.org/`.

Wenzel, S. (2014). App'ification of enterprise software: A multiple-case study of big data business applications. In *Business Information Systems - 17th International Conference, BIS 2014, Larnaca, Cyprus, Mayo 22-23, 2014. Proceedings*, (pp. 61–72).