Un enfoque para la detección de anomalías en el tráfico de red usando imágenes y técnicas de Computación de Alto Desempeño.

Mercedes Barrionuevo, Mariela Lopresti, Natalia Miranda, Fabiana Píccoli
LIDIC. Universidad Nacional de San Luis,
Ejército de los Andes 950 - 5700 - San Luis - Argentina
{mdbarrio, omlopres, ncmiran, mpiccoli}@ unsl.edu.ar

Abstract. La variedad y la complejidad del tráfico actual en Internet superan lo imaginado por los diseñadores de su arquitectura. Detectar posibles ataques en la red requiere contar con tecnologías para su clasificación, asociando flujos de datos con las aplicaciones que los generan. Uno de los desafíos actuales es trabajar con un conjunto de datos que crece a mayor velocidad que su capacidad de procesamiento. Utilizar y procesar imágenes para representar el tráfico de red a fin de detectar tráfico anómalo, tiene como ventajas no sólo contar con una herramienta de visualización de tráfico, sino también con las propiedades de las imágenes y su procesamiento: técnicas bien conocidas y naturaleza paralela de las computaciones. Este trabajo presenta la arquitectura de una herramienta eficiente para detectar anomalías en el tráfico de una red mediante un trabajo computacional sobre imágenes y aplicando técnicas de computación de alto desempeño, mostrando algunos resultados.

Palabras claves: Tráfico de Red. Anomalías. Procesamiento de Imágenes. SIFT. Computación de Alto Desempeño. GPUs.

1 Introducción

En la actualidad, Internet constituye uno de los principales actores de nuestra sociedad, juega un papel crucial en la educación, el entretenimiento, los negocios y la vida social. Este nuevo rol trae como consecuencia un rápido y sostenido tráfico en la red mundial, no sólo en cuanto al volumen de los datos sino también en la heterogeneidad y complejidad de los mismos. Varias son las causas del incremento del tráfico global, destacándose el constante crecimiento tanto de usuarios conectados como de dispositivos de Internet, la mayor velocidad de las conexiones, el creciente consumo de vídeo. Se estima para el año 2018 una generación de tráfico de alrededor de los 1,6 Zettabytes, producto de las conexiones fijas y móviles [8]. Teniendo en cuenta esto, realizar un análisis de tráfico en redes de gran escala resulta una tarea costosa y desafiante.

El tráfico circulante en Internet debe ser caracterizado con el fin de lograr una comprensión completa de su composición real, otorgando información detallada de las distintas actividades de gestión de red tales como la ingeniería de tráfico, diagnóstico de fallas, rendimiento de las aplicaciones, detección de anomalías, etc. La caracterización de tráfico tiene como principales obstáculos: la falta de una variedad de huellas, las cuales puedan servir como datos de prueba; la ausencia de objetos de flujo utilizados como referencia para su validación; y los inesperados problemas de congestión en la red central causado por distintas anomalías [8,9].

Detectar anomalías en redes de datos consiste en la identificación de patrones que se desvían del comportamiento normal de tráfico, por lo que está íntimamente relacionado con el modelado de tráfico. Con el fin de descubrir comportamientos anormales, se deben utilizar modelos de tráfico preciso y estable para describir lo que constituye un comportamiento de tráfico libre de anomalías. Este es de hecho un paso crítico en la detección de anomalías, ya que un modelo de tráfico incorrecto o inestable puede echar a perder completamente la capacidad de una correcta detección y causar un alto número de falsas alarmas.

Para modelar el tráfico de red de una forma entendible, varias de las investigaciones se han enfocado en el análisis de señales como ondas y en el modelado de tráfico como imágenes. Nuestro interés está puesto en este último dado que la visualización proporciona una capacidad de comprender enormes cantidades de datos, permitiendo percibir propiedades emergentes no anticipadas. Esto facilita la comprensión de las características tanto a gran como a pequeña escala de los datos, permitiendo revelar propiedades no sólo relacionada a los datos en sí, sino a la forma en la cual son recolectados [18].

En base a lo expresado anteriormente y teniendo en cuenta la alta demanda de recursos computacionales para producir un análisis eficiente del tráfico de red mediante imágenes produciendo alertas en tiempo real, es necesario analizar la posibilidad de aplicar técnicas de programación, las cuales permitan tomar ventajas de las características del procesamiento de imágenes como son el paralelismo natural que contiene. Por todo ello, considerar la inclusión de técnicas de computación de alto desempeño (HPC) en la solución, particularmente GPGPU (General Purpose Graphics Programming Unit) resulta ser una buena opción.

El objetivo de este trabajo es detectar posibles anomalías en el tráfico de una red de alta velocidad haciendo uso de una combinación de técnicas de análisis de tráfico de red, procesamiento de imágenes y computación de alto desempeño. Para cumplir con el objetivo se propone analizar los paquetes de red, generando a partir de los datos capturados imágenes, procesarlas buscando similitudes para encontrar anomalía en el tráfico de la red de interés. En cada tarea se pretende aplicar técnicas de HPC.

Este documento está organizado como sigue: la próxima sección describe los conceptos previos necesarios para los desarrollos. La sección 3 describe al sistema propuesto y sus módulos. En la sección 4 se muestra resultados experimentales del módulo analizador de Imágenes, y finalmente, se detallan las conclusiones y trabajos futuros.

2. Conceptos Básicos

El presente trabajo involucra diferentes conceptos, los cuales pertenecen a distintas áreas de la informática, podemos considerar tres grandes áreas: tráfico en las redes de computadoras y sus anomalías, imágenes y su procesamiento, y computación de alto desempeño. En esta sección describimos los conceptos básicos de cada una de las áreas involucrados en el desarrollo del presente trabajo..

2.1 Redes de Computadoras: Tráfico y Anomalías

En una red, las computadoras se comunican entre ellas a través de mensajes, los cuales son transportados por la red en paquetes de datos. Para que la comunicación se lleve a cabo,

las computadoras partícipes siguen un protocolo, el cual determina la forma en que los mensajes viajan en los paquetes y cómo éstos son armados.

Una secuencia unidireccional de paquetes con características comunes es denominada flujo de datos. Entre las características comunes consideradas en un flujo (f) de datos se destacan: las direcciones del origen (SRC_{IP}) y destino ($DEST_{IP}$), la identificación del protocolo a nivel de capa 3 (*Transport Protocol*) y los puertos origen (SRC_{Port}) y destino de la comunicación ($DEST_{Port}$), f = {SRC_{IP}, SRC_{Port}, $DEST_{IP}$, $DEST_{Port}$, *Transport Protocol* }. Por ejemplo los flujos:

$$f_{Skipe} = \{\ 10.0.0.1,\ 31215,\ \textit{212.48.72.19,}\ 80,\ \textit{UDP}\}$$
$$f_{SMTP} = \{\ 10.0.0.1,\ 2233,\ 13.29.10.199,\ 25,\ \textit{TCP}\}$$

describen al flujo de datos de la aplicación Skype y del protocolo SMTP respectivamente.

En el tráfico de una red, co-existen distintos flujos, dependiendo de las comunicaciones que se establezcan, entre los cuales podemos encontrar tráfico normal o alteraciones del mismo. Detectar estas alteraciones se refiere a encontrar patrones no conformes con el comportamiento esperado [8]. Para lograrlo es necesario, primero, clasificar el tráfico de una red.

La clasificación del tráfico de red no sólo consiste en la identificación de datos, sino también en la extracción de la información de interés [18]. Si bien existen varios métodos de clasificación de tráfico, los cuatro principales son aquellos basados en puertos, payload, comportamiento de host y basado en características de flujos.

A continuación se da una breve descripción de cada uno de ellos.

- Basado en *puertos*: tienen en cuenta la asignación de puertos bien conocidos asignados por la Autoridad de Asignación de Números de Internet (IANA).
- Basados en *payload*: consisten en inspeccionar el contenido de los paquetes a nivel de transporte para identificar patrones característicos o para identificar secuencias relacionadas con el protocolo a nivel de aplicación, coincidente con un conjunto de reglas predefinidas.
- Basado en *comportamiento de host*: este enfoque considera el número de conexiones abiertas, los puertos usados y alguna combinación de ellas para trazar un perfil típico del host y compararlo con perfiles previamente almacenados.
- Basado en *características de flujo de paquetes*: este enfoque usa características de flujo tales como tamaño promedio de paquetes, tiempos entre arribos, duración de flujos, entre otras. La clasificación consiste en realizar una comparación estadística entre el tráfico desconocido y las reglas aprendidas previamente [15]. Este enfoque incluye principalmente técnicas de minería de datos y algoritmos de aprendizaje automático.

En base a varias evaluaciones existentes [2] se ha podido observar que no es posible proponer el mejor enfoque para la selección de un único algoritmo con un rendimiento óptimo para todos los escenarios posibles. Es por ello que es necesario pensar en métodos alternativos o combinaciones de métodos de diversas áreas para realizar la tarea de clasificación y análisis de tráfico de red.

Respecto a las anomalías, identificarlas, diagnosticarlas y tratarlas en el momento oportuno es fundamental para el administrador de red, debiendo estar entre las tareas principales de éste, ya que sin esta capacidad, las redes pueden funcionar de manera ineficiente y no confiable. Para una identificación precisa y un correcto diagnóstico es necesario contar con datos sólidos y buenos métodos.

Si consideramos que las anomalías de red son aquellas situaciones en que el tráfico de una red difiere notablemente de su perfil habitual, teniendo en cuenta que el tráfico, intrínsecamente, presenta una serie de variaciones producidas por la diversidad de usuarios, servicios y aplicaciones, así como de ciclos temporales diarios y semanales; el problema de

analizar el tráfico de una red y obtener información de él es un problema con muchas dimensiones, tornándose difícil y computacionalmente costoso.

Como mencionamos antes, la primera tarea es la recolección y procesamiento de los paquetes que viajan en la red. Esta tarea puede ser realizada de distintas maneras y a través de distintas herramientas, entre ellas se destacan:

- Bibliotecas como *libpcap* (www.tcpdump.org/), portable en C/C++ con las funciones necesarias para la captura de tráfico de red. Es utilizada por Tcpdump (hwww.tcpdump.org/), Snort (www.snort.org/), Wireshark (www.wireshark.org) y Nmap (nmap.org/),
- Herramientas *NetFlow* (www.manageengine.com/products/netflow) o JFlow (www.juniper.net) pertenecientes a Cisco y Juniper, respectivamente, usadas por Flow-Tools y Ntop.
- Formatos *NLANR* (www.caida.org/projects/nlanr/), como por ejemplo DAG, Coral y TSH, Time Stamped Headers.

Una vez recolectados los paquetes, se inspeccionan las cabeceras extrayendo, por ejemplo las direcciones IP del origen y destino, los números de puertos, el volumen de tráfico en bytes, el protocolo, entre otros. Qué datos extraer dependerá de lo que se desee analizar.

En la siguiente sección introducimos las características principales relacionadas a las imágenes y su procesamiento.

2.2 Características de las Imágenes y su procesamiento.

Usar imágenes para representar datos ayuda a mostrarlos de manera más sencilla, y obtener conclusiones de forma más fácil. En el ámbito del análisis de tráfico de red, un conjunto de paquetes de datos puede ser representado como una imágen. Esto hace que se puedan descifrar patrones de tráfico más rápidamente.

Teniendo en cuenta esto, representar el flujo de datos del tráfico de una red mediante imágenes implica dos tareas, la primera es obtener la imagen a partir de los datos recolectados, y el procesamiento de la imagen obtenida a fin de extraer información y encontrar evidencia de anomalías.

Para la primer tarea, es sabido que un píxel es la menor unidad homogénea en color que forma parte de una imagen digital, la codificación de imágenes se puede realizar escribiendo en forma sucesiva los bits correspondientes a cada píxel línea por línea, o escribiendo los valores de los píxeles en formato ASCII. Si la imagen simboliza el tráfico de una red, el significado de un pixel puede ser, según NetViewer[11], la comunicación entre un origen y un destino, siendo la intensidad del color la cantidad de tráfico originado entre ellos.

Otra de las características a tener en cuenta es el formato de la imagen digital con la cual se va a trabajar, éste debe ser simple, sencillo y eficiente.

Extraer información relevante de las imágenes digitales constituye un gran reto en el campo de visión artificial, más precisamente en la detección y reconocimiento de objetos[3,13,15]. Para la búsqueda de características distintivas de las imágenes se han utilizado características locales, las cuales han demostrado ser muy efectivas. La idea es determinar las características distintivas de la imagen, encontrando sus puntos significativos y una descripción discriminante. A partir de esta descripción se podrán establecer similitudes con otras imágenes. Dos objetos iguales en diferentes imágenes deben ser identificados independientemente de la posición y escala.

Para lograr esto, existen diferentes métodos, uno de los más ampliamente usados es el método SIFT (Scale Invariant Feature Transform) propuesto por Lowe en 1999 [14]. Este algoritmo extrae aquellas características invariantes a escala, rotación, modificación de iluminación, ruidos y pequeños cambios de perspectiva. En función de ellas, es posible comparar dos imágenes aplicando la diferencia euclidiana entre los respectivos descriptores.

La idea de este trabajo es tener imágenes representativas de las anomalías típicas en el tráfico de red, las cuales serán utilizadas para compararlas con las generadas por el tráfico actual. De la comparación de las imágenes, se determinará la presencia o no de alguna anomalía. Si el resultado de la comparación es un valor inferior a un umbral de tolerancia, el tráfico actual no presenta anomalías, caso contrario estaremos en presencia de una, pudiendo también establecer cuál. En la sección 3 se detalla el sistema propuesto.

2.3 Computación de Alto Desempeño usando GPUs

El bajo costo y marcado incremento del rendimiento en las aplicaciones, permitieron a las unidades de procesamiento gráfico GPU, su fácil incorporación al mercado, convirtiéndose en una parte integral de los sistemas actuales de computación. La GPU tiene una estructura altamente paralela y buen ancho de banda de memoria, ofreciendo aceleración a cualquier aplicación de gráficos [12].

Hoy en día el poder computacional de la GPU es utilizada para resolver cualquier tipo de problemas, no solo de naturaleza gráfica. Esto se conoce como GPGPU (Computación de Propósito General en GPU)[12, 17]. Cuando se usa la GPU como computadora paralela se tienen en cuenta el número de unidades de procesamiento, la estructura híbrida de memoria y los posibles hilos trabajando en paralelo.

El modelo de programación CUDA, desarrollado por NVIDIA y soportado a partir de la GeForce 8 Series, permite utilizar la GPU como una computadora altamente paralela para aplicaciones no gráficas [16,17], proporciona un entorno de programación de alto nivel con el lenguaje estándar C/C++. Define una arquitectura para la GPU como una unidad programable que actúa como un coprocesador para la CPU. Ésta tiene múltiples multiprocesadores de streaming (SMS), cada uno de ellos con varios procesadores escalares (SP).

CUDA tiene dos características principales: el trabajo paralelo a través de threads concurrentes y la jerarquía de memoria. El usuario provee un programa fuente, el cual incluye el código para la CPU y el de la GPU (kernel). Un programa CUDA se compone de varias fases a ejecutarse en la CPU o la GPU. En la GPU múltiples threads resuelven el problema en paralelo aprovechando la jerarquía de memoria.

En [5,6,7,10] utilizan la GPU, por las características mencionadas antes, con diferentes técnicas de análisis de datos masivos. Estas técnicas se han desarrollado para ámbitos tales como aprendizaje automático, búsqueda y ordenación, minería de datos, base de datos, entre otros.

El problema abordado en este trabajo se caracteriza por tener varias etapas, la mayoría de las cuales son aptas para resolverse usando técnicas de computación de alto desempeño y, en particular para GPU. El objetivo es acelerar la detección de anomalías en el tráfico de una red casi inmediata a su ocurrencia.

3. Sistema de Detección de Anomalías

Para detectar posibles anomalías en una red de computadoras, combinando técnicas de procesamiento de imágenes y computación de alto desempeño, se propone un sistema encargado de capturar el tráfico de la red, organizarlo en flujos de datos, generar las imágenes que los representan para luego analizarlas respecto a un repositorio de imágenes de anomalías preestablecidas. En la Figura 1 mostramos una representación gráfica del sistema.

De la arquitectura del sistema planteado, se puede distinguir claramente cuatro módulos cada uno con funciones bien diferenciadas. Las funciones y características de cada uno de los ellos son:

- **Módulo de Captura de tráfico:** tiene la función de capturar el tráfico circulante en la red y clasificarlo en flujos. Para la primer tarea, como se mencionó en la sección anterior existen varias herramientas, en nuestro caso seleccionamos la ampliamente utilizada herramienta Wireshark por su robustez.
  Del gran conjunto de paquetes obtenidos, se seleccionan sólo aquellas características necesarias para el análisis. En este primer enfoque, y dada su relevancia, las características que consideramos son: direcciones IP de origen y destino, puertos origen y destino de la comunicación y protocolo a nivel de capa 3. A partir de estos datos se realiza la clasificación de los paquetes en flujos de datos, los cuales son la entrada del módulo de Generación de Imágenes.
- **Módulo de Generación de imágenes:** Este módulo toma como entrada los distintos flujos y los transforma en imágenes, salida del módulo.
  Como se mencionó anteriormente, las imágenes generadas deben tener un formato simple, sencillo y eficiente, por dicha razón se seleccionó el formato pgm. Una imagen *pgm* (Portable Gray Map) se caracteriza por ser una representación en escala de grises.
  Es un formato muy fácil de entender y usar en programación, además de muy versátil. Una imagen está formada por sólo 3 líneas para el descriptor: tipo de formato (P2 si es formato ASCII o P5 si es binario), ancho y alto de la imagen, y niveles de grises (máximo 255). El resto son los píxeles. La Figura 2 ilustra una codificación y su imagen correspondiente.

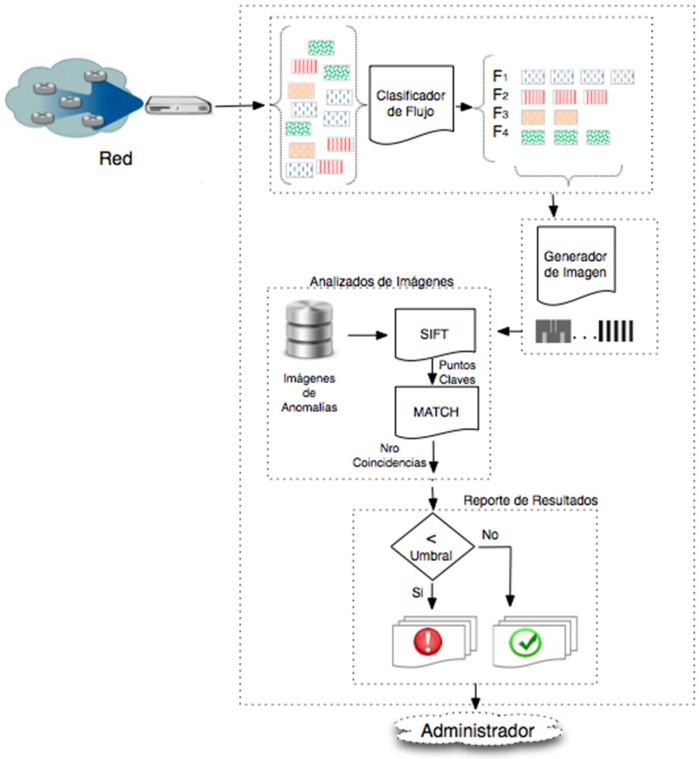


Figura 1. *Estructura del Sistema de Detección de Anomalías Propuesto*

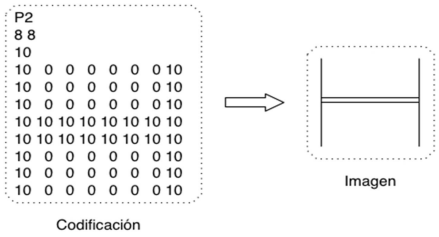


Figura 2. *Código digital de una imagen PGM y la imagen correspondient***e**

Una vez logradas las imágenes, éstas son remitidas para su análisis al módulo siguiente.

- **Módulo de Análisis de Imágenes:** A partir de las imágenes generadas, este módulo es responsable de encontrar similitudes entre ellas con cada una de las imágenes que representan anomalías. Las mismas conforman un repositorio en constante actualización. Esto se debe a diversos criterios o nuevas características de los ataques ya conocidos. Ejemplos de imágenes de anomalías son:

1. Líneas Horizontales: definirían una exploración desde la misma dirección IP de origen a múltiples direcciones IP destino.
2. Líneas Verticales: Representan un recorrido secuencial de varios equipos (en una subred) a una única dirección de destino. Se estaría llevando a cabo un probable DDoS (Denegación de Servicio Distribuido) a un nodo.

Como se mencionó en secciones previas, la comparación de imágenes debe ser capaz de encontrar similitudes entre imágenes, o partes de ellas, digitalmente diferentes. El algoritmo SIFT, tiene la propiedad de tomar como entrada dos imágenes y encontrar puntos característicos de cada una de ellas. Si los puntos claves de una imagen de la anomalía son identificados en la imagen de consulta, un posible ataque o anomalía sería detectado, caso contrario el tráfico actual no presentaría intrusiones conocidas.
En este módulo y como primer paso, proponemos aplicar técnicas de alto desempeño y GPU. Tanto el algoritmo SIFT como el analizador de coincidencias (MATCH) son realizadas en paralelo. Si bien existen varias implementaciones, nosotros utilizamos la propuesta por [1,4].

- **Módulo de Reporte de Resultados:** En base a los resultados del analizador, este módulo es el responsable de determinar si existe una posible anomalía y generar las advertencias correspondientes. La decisión se toma en base a la cantidad de coincidencias o puntos claves reportados por la etapa anterior, si éstas superan un umbral preestablecido, se estaría en presencia de una anomalía. Entre las advertencias, se encuentra la generación de mensajes de log del sistema y alarmas de red. Cada una con su propio formato, formas o métodos de reporte y frecuencia de alerta.

4. Resultados Experimentales

En este primer paso nos hemos enfocamos en el módulo de Análisis de Imágenes. Para ello hemos evaluado dicho módulo a través de varios experimentos.

Como las imágenes son generadas a partir de los datos del flujo que se desean analizar, el tamaño de ellas va a depender de la información considerada de cada uno de los paquetes capturados. Es por ello que los experimentos fueron direccionados a evaluar el comportamiento del módulo para imágenes de diferentes tamaños, los cuales oscilan entre los 50KB y los 20MB.

Para la evaluación, se consideraron las mejoras en el tiempo, respecto al algoritmo secuencial, para obtener los puntos claves de las imágenes y compararlos.

Los resultados del algoritmo SIFT secuencial fueron tomados un procesador AMD FX(tm)-6300 Six-Core Processor x 6 con Memoria de 7,8 GB y un disco de 80,5 GB. Para el SIFT Paralelo, la GPU es una Tesla K20c con 2496 procesadores CUDA, 4,6 GB de Memoria y Frecuencia de reloj del procesador y memoria 706 MHz y 2600 Mhz respectivamente.

En la Tabla 1 se detallan los segundos necesarios para obtener los puntos claves de una imagen mediante el algoritmo SIFT en sus dos versiones: secuencial y paralela.

Tamaño de Imagen entre	SIFT Secuencial	SIFT Paralelo
10KB y 300 KB	1,78092	0,014045
300KB y 1MB	4,650075	0,020615
1MB y 3MB	22,08463	0,0646675
3MB y 5MB	40,114555	0,09601

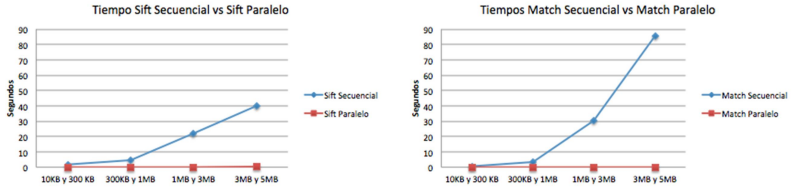
Tabla 1. Tiempos promedios en segundos del Sift Secuencial y Paralelo para distintos tamaños de imágenes.

La Tabla 2 muestra cuantos segundos en promedio se requieren para determinar si las imágenes comparadas tienen puntos claves en común (Match). Los tiempos se especifican también para las versiones secuencial y paralela.

Tamaño de Imagen entre	Match Secuencial	Match Paralelo
10KB y 300 KB	0,63709	0,004635
300KB y 1MB	3,325095	0,00805
1MB y 3MB	30,42166	0,026885
3MB y 5MB	85,51374	0,07112

Tabla 2. Tiempos promedios en segundos del Match Secuencial y Paralelo para distintos tamaños de imágenes.

La Figura 3 representa gráficamente la información de ambas tablas. Para todos los resultados se tomó el promedio de 10 ejecuciones para cada imagen considerada.



(a) (b)

Figura 3. Tiempos en segundos de cada una de las etapas involucradas en el módulo de Análisis de Imágenes.

Como se puede ver en la Figura 3, tanto para el algoritmo SIFT(a) como para el proceso de Match(b), los tiempos de la solución paralela son sensiblemente inferiores a los de la solución secuencial para todos los tamaños de imágenes considerados, siendo mayor la diferencia en el Match. La reducción de los tiempos es mayor a los 100x.

5. Conclusiones y Trabajos Futuros

Existen varias herramientas cuya función principal es monitorizar y analizar tráfico de red, a fin de detectar posibles ataques. La captura y visualización de las tramas de datos puede ser muy útil o eficiente si además se tienen los paquetes de datos y la información del flujo de tráfico. Todo esto tendrá éxito si se puede detectar cualquier posible alteración y tomar buenas decisiones tan rápido como sea posible.

En este trabajo proponemos la arquitectura de un sistema para detectar anomalías en el tráfico de una red mediante el procesamiento de imágenes y técnicas de computación de alto desempeño. El análisis de las mejoras logradas considerando la aplicación de estas técnicas en una única etapa del proceso general, nos habilita para seguir trabajando en la misma dirección y lograr un sistema integralmente paralelo, el cual permitirá la detección de anomalías rápidamente a fin de lograr la prevención de ataques.

Entre los trabajos futuros, no sólo está la implementación de cada una de las demás etapas del sistema usando técnicas paralelas en el proceso, sino también evaluar y analizar la posibilidad de mejorar la existente. La propuesta realizada aplica técnicas paralelas al

proceso de resolver la comparación de dos imágenes, paralelismo intra-consulta. Una posibilidad es incluir computación paralela a nivel inter-consulta, es decir analizar al mismo tiempo la imagen de consulta con varias del repositorio.

Referencias

1. A. Acharya, R. Venkatesh Babu, "Speeding up SIFT using GPU," Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference. Pp1-4, 2013.
2. A. Callado, J. Kelner, D. Sadok, C. Kamienski, S. Fernandes. "Better network traffic identification through the independent combination of techniques". J. of Network and Computer Applications. Vol 33, N 4. Pp 433-446, 2010.
3. S. Belongie, J Malik, and J Puzicha. "Shape matching and object recognition using shape contexts". Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on, pages 509–522, 2002.
4. M. Björkman, N. Bergström and D. Kragic, "Detecting, segmenting and tracking unknown objects using multi-label MRF inference". Comp. Vis. Image Underst. Pp 111-127. 2014.
5. S. Chen, J. Qin, Y. Xie, J. Zhao, P. Heng. "A Fast and Flexible Sorting Algorithm with CUDA". Algorithms and Architectures for Parallel Processing. LNCS Vol. 5574. Pp 281-290, 2009.
6. Y. Chen, Z. Qiao, S. Davis, H. Jiang, K. Li. "Pipelined Multi-GPU MapReduce for Big-Data Processing". Comp. and Inf. Science. Springer. Vol. 493. Pp 231-246, 2013.
7. D.Ciresan, A.Giusti, L.Gambardella, J. Schmidhuber. "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks". MICCAI 2013. Pp 411-418, 2013.
8. Cisco Systems, "Global IP Traffic Forecast and Methodology, 2006-2011", white paper available at http://www.cisco.com, 2007 - updated 2008.
9. Cisco Systems, "The Exabyte Era", white paper. http://www.cisco.com, 2008.
10. 10.S. Herrero-Lopez, "Accelerating SVMs by integrating GPUs into MapReduce clusters," Systems, Man, and Cybernetics. IEEE International Conference. Pp. 1298-1305, 2011
11. S. Kim, A. Reddy. "Image-Based Anomaly Detection Technique: Algorithm, Implementation and Effectiveness". IEEE Journal of Selected Areas in Communications 24. Pp 1942–1954, 2006.
12. D. Kirk and W. Hwu. "Programming Massively Parallel Processors, A Hands on Approach". Elsevier, Morgan Kaufmann, 2010.
13. D. Lowe. "Object recognition from local scale-invariant features". Computer Vision (ICCV), pages 1150–1157, 1999.
14. D. Lowe. "Distinctive image features from scale-invariant keypoints". International journal of computer vision. Pp 91-110, 2004.
15. K. Mikolajczyk, B. Leibe, and B. Schiele. "Local features for object class recognition". Computer Vision. 10º IEEE International Conference. Pp 1792–1799, 2005.
16. NVIDIA. "Nvidia cuda compute unified device architecture, C programming guide". Version 7.5. In NVIDIA, 2015.
17. J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips. GPU Computing. In IEEE, volume 96. Pp 879 – 899, 2008.
18. M.Thanigavel1, R. Mallika, T. Sujilatha. "An Open Platform for Internet Traffic Classification in TIE". IJRASET. Vol. 3 Issue VIII, 2015.