

INSTITUTO TECNOLÓGICO BUENOS AIRES

PROYECTO FINAL

PROYECTO TIX

Desarrollo e implementación de una arquitectura horizontalmente escalable

Autores

Matías Gabriel
DOMINGUES
Facundo Nahuel
MARTINEZ CORREA
Javier PÉREZ CUÑARRO

Tutor

Dr. Ing. José Ignacio
ALVAREZ-HAMELIN

18 de diciembre de 2017

Índice

1. Introducción	3
1.1. Sobre el Proyecto TiX	3
1.1.1. Sincronización de Relojes	3
1.1.2. Estimación de Calidad y Uso de la Conexión	3
1.2. Sobre la presente iteración	4
1.2.1. Motivación	4
1.2.2. Objetivos	4
1.2.3. Desarrollo del presente documento	5
2. Arquitectura	6
2.1. Arquitectura previa	6
2.1.1. Responsabilidades e implementación de los subsistemas	6
2.1.2. Funcionamiento conjunto del sistema	7
2.1.3. Fortelazas de esta arquitectura	7
2.1.4. Problemas y debilidades de esta arquitectura	8
2.2. Arquitectura actual	9
2.2.1. Responsabilidades y consideraciones de los subsistemas	11
2.2.2. Funcionamiento conjunto del sistema	16
2.2.3. Fortelazas de esta arquitectura	16
2.2.4. Debilidades y problemas de esta arquitectura	17
2.3. Conclusiones de la arquitectura actual	17
3. Implementación	18
3.1. Protocolo TiX	18
3.1.1. Diseño general del Protocolo	18
3.1.2. Funcionamiento del Protocolo	19
3.1.3. Taxonomía de los paquetes	19
3.1.4. Taxonomía de los reportes	21
3.2. El Subsistema de Ingesta y Procesamiento	21
3.2.1. Servicio tix-time-server	21

3.2.2.	Servicio tix-time-condenser	22
3.2.3.	Servicio tix-time-processor	23
3.2.4.	Despliegue y puesta en producción	27
3.3.	El Subsistema Cliente	27
3.3.1.	Aplicación Cliente	27
3.3.2.	Reporter	33
3.4.	El Subsistema de Presentación y Administración de Datos	34
3.4.1.	Despliegue y puesta en producción	35
4.	Ensayos	41
4.1.	Prueba de Sistema	41
4.1.1.	Objetivos	41
4.1.2.	Indicadores propuestos	41
4.1.3.	Consideraciones	41
4.1.4.	Metodología	42
4.1.5.	Resultados	43
4.2.	Prueba de Carga	44
4.2.1.	Objetivos	44
4.2.2.	Indicadores propuestos	44
4.2.3.	Consideraciones	44
4.2.4.	Metodología	45
4.2.5.	Resultados	47
5.	Conclusiones	52
5.1.	Aprendizajes y resultados	52
5.2.	Trabajo pendiente y posibles mejoras o ampliaciones	53
	Referencias	55

5. Conclusiones

5.1. Aprendizajes y resultados

El proyecto TiX lleva ya unos años en desarrollo y ha tenido unos cuantos colaboradores a lo largo del tiempo. Antes de comenzar esta iteración, han pasado dos equipos de desarrolladores cuyos objetivos eran distintos, principalmente crear el sistema y sumar funcionalidades. Cada uno de ellos se concentró en resolver los problemas propuestos y lograr que el proyecto siga funcionando. Tres años más tarde, el principal desafío fue llevar adelante el desarrollo de un código que estaba en un estado más cercano a una prueba de concepto que a un código apto para ser puesto en producción. Al revisar el material disponible, faltaba documentación de varias funcionalidades (o la misma estaba desactualizada), el sistema no estaba diseñado para ser usado por más de una centena de usuarios, se observaban problemas en distintos módulos y existían incompatibilidades con ciertas versiones de sistemas operativos populares.

Tal como se comentada en la Introducción, esta iteración se centró en que TiX sea un sistema estable y escalable, permitiendo contar con cinco mil usuarios concurrentes. Alcanzar esta meta permitiría abrir el proyecto a una cantidad razonable de usuarios, siempre y cuando quede como un proyecto de uso dentro de la comunidad académica y no de uso masivo.

Gran parte del paso por el proyecto consistió en analizar el estado actual del código, del sistema en total y de la forma óptima para lograr alcanzar nuestro objetivo de escalabilidad. Luego de muchas discusiones, se acordó que era necesario llevar a cabo importantes cambios. Se replanteó el diseño del sistema, las tecnologías usadas en varios módulos, las interacciones entre ellos, etc. Todas estas modificaciones llevaron a retrabajar el alcance y los plazos que se habían acordado los primeros días de esta iteración.

Después de mucho trabajo se logró cumplir con las metas propuestas. El sistema es efectivamente escalable y se observa un funcionamiento estable con casi dos mil usuarios concurrentes. Esto está avalado por los distintos ensayos que se presentan en el capítulo dedicado a ellos.

Las pruebas de carga señalan que el sistema actualmente puede brindar servicio a unos 2000 usuarios al mismo tiempo, lo cual es un gran avance comparado con la iteración anterior. Cabe destacar que no todos los usuarios de un sistema suelen utilizarlo (correr la aplicación) al mismo tiempo - se asume que una fracción de los clientes permanecerán inactivos en cualquier instante. Esto lleva a creer que el sistema hoy en día puede funcionar con más de dos mil usuarios registrados.

El limitante actual de este número es el servidor de TiX, con lo cual habría que probar el sistema con una infraestructura más avanzada y llevar a cabo nuevas pruebas de carga. Más allá de los posibles resultados que arrojen, es importante

señalar que todas las partes del sistema han sido planteadas bajo la premisa de escalabilidad y estabilidad bajo un gran caudal de usuarios.

En cuanto a las pruebas de sistema, éstas indican que el funcionamiento completo del sistema es el esperado: tanto el Subsistema de Ingesta y Procesamiento, el Subsistema Cliente y el Subsistema de Presentación de Datos están cumpliendo debidamente con sus responsabilidades. Los indicadores observados en estas pruebas coinciden con los valores esperados. Y cualquier usuario puede comprobar por su cuenta que hoy es posible instalar el cliente en su sistema operativo, generar mediciones y analizar los resultados desde el sitio web.

Más allá del objetivo de escalabilidad, se ha trabajado en una serie de mejoras notables. La aplicación cliente es sumamente portable y ha incorporado facilidades para el usuario final tales como actualizaciones automáticas, se ha mejorado la seguridad del sistema mediante la encriptación de datos y comunicaciones entre distintos subsistemas, se presentan diversas mejoras en el Subsistema de Presentación de Datos, etc.

Otro logro importante ha sido preparar el sistema para un mantenimiento sencillo. Por ejemplo, se optó por usar Docker para agrupar partes de código en distintos componentes, y se incluyeron tests para permitir hacer cambios en código sin que peligre su funcionamiento. El resultado es un proyecto de Código Libre de calidad. Esto es de suma importancia en un proyecto de carácter académico ya que permite que las próximas iteraciones de código se hagan sin grandes sobresaltos y hasta posiblemente se sume gente externa.

Sin duda, se han puesto en práctica innumerables conocimientos adquiridos a lo largo de la carrera de Ingeniería Informática: nociones de diseño de sistemas, concurrencia, encriptado, aplicaciones web, aplicaciones de escritorio, programación orientada a objetos, protocolos de comunicación y muchos más. Pero lo más importante ha sido aplicar la capacidad de análisis crítico para concluir que sólo sería posible llegar al objetivo replanteando gran parte del trabajo hecho.

5.2. Trabajo pendiente y posibles mejoras o ampliaciones

A lo largo del proceso de diseño y desarrollo, fueron surgiendo ideas o posibles mejoras para la plataforma, que por no considerarse prioritarias para llegar al objetivo planteado, se dejaron como propuestas a ser resueltas cuando el objetivo sea distinto o se cuente con los recursos para hacerlos realidad.

Cabe destacar, que si bien el objetivo principal del proyecto era soportar hasta 5000 usuarios concurrentes, esto no pudo ser verificado bajo las condiciones actuales dado la infraestructura con la que se cuenta, pero es posible dada la arquitectura del sistema. Este proceso, generó muchas propuestas sobre cómo llegar al requerimiento con la infraestructura actual, pero la complejidad de este trabajo no justifica la necesidad de nueva infraestructura, además de no existir seguridad que dicha mejora funcione.

Finalmente, el grupo plantea las siguientes propuestas:

- Aplicación móvil con un protocolo optimizado.
- Actualizaciones automáticas del Cliente en todo momento y no sólo cuando se inicia.
- Implementar y poner en producción un conjunto de herramientas que permitan realizar el trabajo de operaciones del día a día en el Sistema TiX instalado en el servidor.
- Cambiar el diseño del protocolo para que ya no sea solamente por UDP sino que sea por UDP y HTTP. La idea es que los paquetes UDP sean de PING a un servicio especial para esto y los paquetes de reportes vayan a otro servicio que sea HTTP y sea el nuevo tix-time-server.
- Hacer una biblioteca aparte para el Analyzer y el ReportsHandler en el tix-time-processor, que estén distribuidas por PyPi para una fácil instalación. Esto también permitiría poner los notebooks de análisis en otro repositorio que no sea el del tix-time-processor.
- Mejorar el paquete tix-time-core incluyendo la parte de serialización y deserialización de paquetes no sólo en UDP, sino también en otros formatos como JSON.
- Hacer pruebas con Celery usando Redis en vez de contra RabbitMQ como soporte para analizar el impacto en la escalabilidad y la performance del servicio.
- Analizar la posibilidad de hacer el procesamiento por lotes con herramientas de análisis de grandes cantidades de datos, como lo son Apache Hadoop, Apache Spark y Presto.
- Implementar mejoras a IP2AS que permita hacer queries más eficientemente.
- Implementar cache en la API para reducir el tiempo de respuesta y no consumir datos desde la base de datos.

Referencias

- [1] J. Postel, “User datagram protocol,” Internet Engineering Task Force, Tech. Rep., aug 1980. [Online]. Available: <https://www.ietf.org/rfc/rfc768.txt>
- [2] D. Mills, “Network time protocol (NTP),” Internet Engineering Task Force, Tech. Rep., sep 1985. [Online]. Available: <https://tools.ietf.org/html/rfc958>
- [3] M. E. Crovella and A. Bestavros, “Self-similarity in world wide web traffic: evidence and possible causes,” *IEEE/ACM Transactions on networking*, vol. 5, no. 6, pp. 835–846, 1997.
- [4] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic (extended version),” *IEEE/ACM Transactions on networking*, vol. 2, no. 1, pp. 1–15, 1994.
- [5] Oracle, “Java Language Documentation.” [Online]. Available: <https://docs.oracle.com/en/java/>
- [6] R. Johnson, J. Hoeller, K. Donald, C. Sampaleanu, R. Harrop, A. Arendsen, T. Risberg, D. Davison, D. Kopylenko, M. Pollack, T. Templier, E. Vervaet, P. Tung, B. Hale, A. Colyer, J. Lewis, C. Leau, M. Fisher, S. Brannen, R. Laddad, A. Poutsma, C. Beams, T. Abedrabbo, A. Clement, D. Syer, O. Gierke, and R. Stoyanchev, “Spring Framework Documentation.” [Online]. Available: <https://docs.spring.io/spring/docs/3.1.1.RELEASE/spring-framework-reference/html/>
- [7] Red Hat JBoss Middleware, “Hibernate ORM Documentation.” [Online]. Available: <http://hibernate.org/orm/documentation/>
- [8] Apache Software Foundation, “Apache Wicket Framework Documentation.” [Online]. Available: <https://ci.apache.org/projects/wicket/apidocs/1.5.x/index.html>
- [9] R. Mordani, Oracle, N. Abramson, Apache Software Foundation Art Technology Group Inc.(ATG), K. Avedal, BEA Systems, H. Bergsten, Boeing, Borland Software Corporation, Developmentor, J. Hunter, IBM, InterX PLC, R. Johnson, Lutris Technologies, New Atlanta Communications, LLC, Novell, Inc., Persistence Software, Inc., Pramati Technologies Progress Software, SAS Institute, Inc., Sun Microsystems, Inc., Sybase, and G. Wilkins, “Java™ Servlet 2.4 Specification,” Java Community Process, techreport 154, Nov. 2003. [Online]. Available: <https://jcp.org/en/jsr/detail?id=154>
- [10] Python Software Foundation, “Python 2 Language Documentation.” [Online]. Available: <https://docs.python.org/2/>
- [11] R Development Core Team, “The R Language Manuals.” [Online]. Available: <https://cran.r-project.org/manuals.html>

- [12] J. Buck and L. Hambley, “Capistrano Documentation.” [Online]. Available: <http://capistranorb.com/>
- [13] Y. Matsumoto, “Ruby Language Documentation.” [Online]. Available: <https://www.ruby-lang.org/en/documentation/>
- [14] The Open Group and IEEE Std, “crontab,” The Open Group, techreport, 2001. [Online]. Available: <http://pubs.opengroup.org/onlinepubs/9699919799/utilities/crontab.html>
- [15] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” phdthesis, University of California, 2000. [Online]. Available: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [16] Travis-CI Community, “Travis-CI.” [Online]. Available: <http://travis-ci.org/>
- [17] Docker, Inc., “Docker.” [Online]. Available: <https://www.docker.com/>
- [18] —, “Dockerhub.” [Online]. Available: <https://hub.docker.com/>
- [19] —, “Docker Compose Documentation.” [Online]. Available: <https://docs.docker.com/compose/>
- [20] Apache Software Foundation, “Apache Maven.” [Online]. Available: <https://maven.apache.org/>
- [21] T. Lindholm, F. Yellin, G. Bracha, and A. Buckley, *The Java Virtual Machine Specification, Java SE 8 Edition*, A.-W. Professional, Ed. Addison-Wesley Professional, 2014. [Online]. Available: <https://docs.oracle.com/javase/specs/jvms/se8/html/index.html>
- [22] Facebook, Instagram, and Community, “React framework (reactjs).” [Online]. Available: <https://reactjs.org/>
- [23] T. Koppers, S. Larkin, J. Ewald, J. Vepsäläinen, K. Kluskens, and W. contributors, “Webpack.” [Online]. Available: <https://webpack.js.org/>
- [24] M. Jones, J. Bradley, and N. Sakimura, “JSON web token (JWT),” IETF, Tech. Rep., may 2015. [Online]. Available: <https://tools.ietf.org/html/rfc7519>
- [25] E. R. Fielding and E. J. Reschke, “Hypertext transfer protocol (HTTP/1.1): Authentication,” IETF, Tech. Rep., jun 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7235>
- [26] Netty Project Community, “Netty.” [Online]. Available: <https://netty.io/>
- [27] P. Webb, D. Syer, J. Long, S. Nicoll, R. Winch, A. Wilkinson, M. Overdijk, C. Dupuis, S. Deleuze, and M. Simons, “Spring Boot Documentation,” 2017. [Online]. Available: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

- [28] The Scipy Community, “NumPy Reference Guide.” [Online]. Available: <https://docs.scipy.org/doc/numpy-1.13.0/reference/>
- [29] —, “Scipy reference guide.” [Online]. Available: <https://docs.scipy.org/doc/scipy-1.0.0/reference/>
- [30] J. D. Hunter, M. Droettboom, T. A. Caswell, and The Matplotlib Community, “Matplotlib user’s guide.” [Online]. Available: <http://matplotlib.org/users/index.html>
- [31] The scikit-learn Community, “scikit-learn documentation.” [Online]. Available: <http://scikit-learn.org/stable/documentation.html>
- [32] Jupyter Team, “Jupyter documentation.” [Online]. Available: <https://jupyter.readthedocs.io/en/latest/contents.html>
- [33] A. Solem, “Celery user manual.” [Online]. Available: <http://docs.celeryproject.org/en/latest/>
- [34] Pivotal, “Rabbitmq documentation.” [Online]. Available: <https://www.rabbitmq.com/documentation.html>
- [35] J. Postel, “Transmission control protocol,” DARPA Internet Program, Tech. Rep., sep 1981. [Online]. Available: <https://tools.ietf.org/html/rfc793>
- [36] J. Nagle, “Congestion control in IP/TCP internetworks,” Internet Engineering Task Force, Tech. Rep., jan 1984. [Online]. Available: <https://tools.ietf.org/html/rfc896>
- [37] F. Yergeau, “UTF-8, a transformation format of ISO 10646,” Internet Engineering Task Force, Tech. Rep., nov 2003. [Online]. Available: <https://tools.ietf.org/html/rfc3629>
- [38] R. L. Rivest, A. Shamir, and L. M. Adleman, “Cryptographic communications system and method,” United States Patent Patent 4,405,829, 1983. [Online]. Available: <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnethtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=4405829.PN.&OS=PN/4405829&RS=PN/4405829>
- [39] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas, “Internet x.509 public key infrastructure: Certification path building,” Internet Engineering Task Force, Tech. Rep., sep 2005. [Online]. Available: <https://tools.ietf.org/html/rfc4158>
- [40] Q. H. Dang, “Secure hash standard,” National Institute Of Standards and Technology, Tech. Rep., jul 2015. [Online]. Available: <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>

- [41] B. Kaliski, “PKCS #1: RSA encryption version 1.5,” Internet Engineering Task Force, Tech. Rep., mar 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2437>
- [42] S. Aiyagari, A. Richardson, M. Arrott, M. Ritchie, M. Atwell, S. Sadjadi, J. Brome, R. Schloming, A. Conway, S. Shaw, R. Godfrey, M. Sustrik, R. Greig, C. Trieloff, P. Hintjens, K. van der Riet, J. O’Hara, S. Vinoski, and M. Radestock, “Advanced message queuing protocol,” , techreport, 2008. [Online]. Available: <https://www.rabbitmq.com/resources/specs/amqp0-9-1.pdf>
- [43] T. Bray, “The JavaScript object notation (JSON) data interchange format,” Internet Engineering Task Force, Tech. Rep., mar 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7159>
- [44] F. Wasilewski, “Pywavelets documentation.” [Online]. Available: <https://pywavelets.readthedocs.io/en/latest/#license>
- [45] Oracle, “Javafx.” [Online]. Available: <http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html>
- [46] —, “Swing.” [Online]. Available: <https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>
- [47] —, “FXML.” [Online]. Available: https://docs.oracle.com/javafx/2/api/javafx/fxml/doc-files/introduction_to_fxml.html
- [48] —, “Java preferences api.” [Online]. Available: <https://docs.oracle.com/javase/8/docs/technotes/guides/preferences/index.html>
- [49] E. Syse and F. contributors, “Fxlauncher.” [Online]. Available: <https://github.com/edvin/fxlauncher>
- [50] Call-Em-All, “Material-ui.” [Online]. Available: <http://www.material-ui.com>
- [51] E. Rasmussen, “Redux.” [Online]. Available: <https://redux.js.org/>
- [52] —, “Redux-form.” [Online]. Available: <https://redux-form.com>
- [53] J. y. c. Linux Foundation, “Node.js.” [Online]. Available: <https://nodejs.org>
- [54] I. y. c. StrongLoop, “Expressjs.” [Online]. Available: <https://expressjs.com/>
- [55] A. y colaboradores, “Passportjs.” [Online]. Available: <http://www.passportjs.org/>
- [56] T. G. y colaboradores, “Knexjs.” [Online]. Available: <http://knexjs.org/>
- [57] B. y colaboradores, “Bookshelfjs.” [Online]. Available: <http://bookshelfjs.org/>

- [58] R. y colaboradores, “Ramdajs.” [Online]. Available: <http://ramdajs.com/>
- [59] Google, “Chrome v8.” [Online]. Available: <https://developers.google.com/v8/>
- [60] I. Z. S. y colaboradores, “Npm.” [Online]. Available: <https://www.npmjs.com/>
- [61] Azureus Software, Inc., “Vuze bittorrent client web page.” [Online]. Available: <https://www.vuze.com/>
- [62] H. Hope, “inxi documentation.” [Online]. Available: <https://smxi.org/docs/inxi-man.htm>
- [63] Apache Software Foundation, “Apache JMeter Documentation.” [Online]. Available: <http://jmeter.apache.org/usermanual/index.html>