Astroinformatics
Proceedings IAU Symposium No. 325, 2016
M. Brescia, S.G. Djorgovski, E. Feigelson,
G. Longo & S. Cavuoti, eds.

doi:10.1017/S174392131601303X

AlertSim - Serbian Contribution to the LSST

Darko Jevremović[1], Veljko Vujčić[1,2], Vladimir A. Srećković[3], Jovan Aleksić[1], Sanja Erkapić[1] and Nenad Milovanović[1]

[1]Astronomical Observatory, Volgina 7, 11060 Belgrade, Serbia
email: darko, veljko, jaleksic, serkapic, nmilovanovic@aob.rs
[2]Faculty of Organizational Sciences, Univesity of Belgrade, Serbia
[3]Institute of Physics Belgrade, University of Belgrade,
P.O. Box 57, 11001, Belgrade, Serbia
email: vlada@ipb.ac.rs

Abstract. We present simulator of alerts for the Large Synoptic Survey Telescope (LSST) developed by Belgrade group. This simulator will be used in testing the functionality of external event brokers/Complex Event Processing (CEP) engines. It is based on current LSST Simulation framework and allows for different classes of objects to be 'alerted'. A Web service based on our simulator is prototyped and can be accessed by developers of brokers/CEP engines.

Keywords. LSST, simulations, transients, alerts, web service

1. Introduction

The Large Synoptic Survey Telescope (LSST, http://lsst.org) is a large-aperture, wide-field, ground-based survey system that will image the sky in six optical bands from 320 to 1050 nm, uniformly covering approximately 18,000deg^2 of the sky over 800 times (Ivezić *et al.* 2008). The LSST is currently under construction on Cerro Pachón in Chile, and expected to enter operations in 2022.

The LSST will generate on average 15 TB of data per night, and will require a comprehensive Data Management system to reduce the raw data to scientifically useful catalogs and images with minimum human intervention. These reductions will result in a real-time alert stream, and eleven data releases over the 10-year duration of LSST operations.

As a part of its nightly operations, LSST is expected to generate a transient alert stream of $\tilde{1}$0,000 events/visit. That stream is planned to be forwarded a) unfiltered, to a number of public event brokers/CEP engines, and b) to an internal simple transient filtering service designed to support the end-users (per Section 3.5 of the Science Requirement Document, Ivezić *et al.* 2011). It is anticipated that due to bandwidth constraints only a limited number of public brokers (on order of ∼2 to ∼4) will be able to connect directly to LSST.

Given the limited number of external brokers that will serve as the primary delivery mechanism of LSST transient data to the public, it is important to assure they are capable of receiving and processing the LSST event stream, both initially and throughout operations. An inefficient or buggy event broker results in opportunity loss for the LSST user community, and in helpdesk/technical support costs to LSST. To achieve the projected return of investment for LSST transient science, it is important to initially validate each external broker's capability, continually monitor it, and have the tools and support personnel to understand and resolve any issues as they arise.

In order to do so, the group in Belgrade develops transient alert simulator (AlertSim) and will contribute, in time for LSST Early Operations, an event broker validation suite.

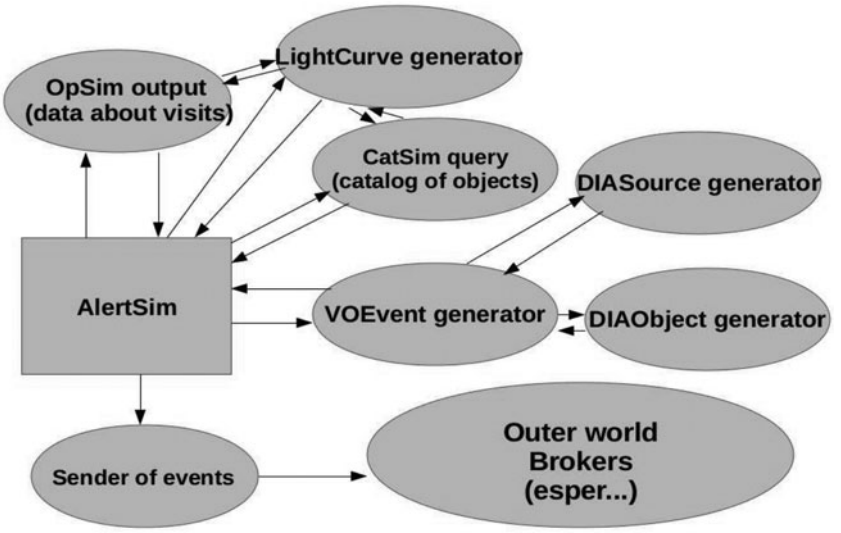

Figure 1. Flow chart for AlertSim

2. AlertSim - Current status

AlertSim is a suite of Python programs which generates simulated alerts and emits them to the outer world. At present time, it accesses different databases developed by the Simulation group of the LSST Systems Engineering. The main components of AlertSim and the basic flow is shown in Fig.1.

The first access is to the LSST Operation Simulator (OpSim) database (Delgado *et al.* 2014) which provides a list of visits for a given constraint (i.e. night, position filter etc.). It also provides the history of previous visits necessary for generating "historical" light curves. With a list of visits AlertSim accesses the LSST Catalog Simulaton database (https://www.lsst.org/scientists/simulations/catsim, CatSim) and for each visit retrieves simulated catalog data for all or part of the objects in "field of view". Next step involves generation of DiaSources and DiaObjects (see Jurić *et al.* 2016 for detailed explanation) from data for "known" variable sources. Now, all or parts of generated data are converted to the VOEvent format (XML Schema, Seaman *et al.* 2011) and emitted to the outer world. This emission is achived using "socket" - low level networking interface in Python. As test receivers, we use python/socket based simple receiver, VOEvent broker Comet (Swinbank 2014), and an Esper (http://www.espertech.com/products/esper.php) based suite.

The codebase for the alert simulation is accessible at https://github.com/lsst-sims/sims_alertsim.

3. Web service

As a part of the suite for testing external brokers, the prototype of the web service has been developed. Service is http://servo.aob.rs/alertsim_voevents and uses free open source Django web framework (https://www.djangoproject.com/). At the moment it is password protected due to the security considerations in accessing Simulation databases at University of Washington (UW, secure channels are not open all the time). Interested event broker/CEP engine developers should contact authors for the access to the service.

We show part of the current user interface in Fig.2. One can see that access to variety

https://doi.org/10.1017/S174392131601303X Published online by Cambridge University Press

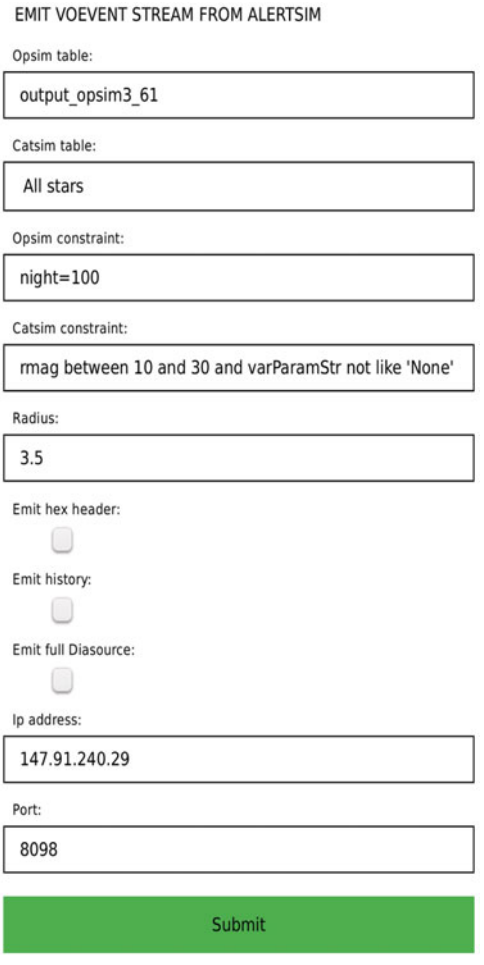

Figure 2. User interface with AlertSim service

of databases and free query constraints are implemented, so the user can get desired stream of alerts to chosen combination of web address/port. Using those queries user can control volume of data delivered to the receiving machine.

The speed of service depends mainly on the speed of execution of database queries and speed of data transfer from UW. The generation/transmition part takes relatively small portion of total time.

4. Future work

There are several steps to be implemented in the near future. The first one is to access output of end-to-end simulations. Currently we use our own DiaSource/DiaObject generator and aim is to use the output of DM-stack (Jurić *et al.* 2015) analysis of simulated images. Also this step will provide cutouts around the "sources" which are missing in the current implementation. The next step is to migrate the service to the space with unlimited access to the Simulation databases. Also we need better implementation of detection of readines of brokers/CEP engines for receiving data. And, of course, as both

https://doi.org/10.1017/S174392131601303X Published online by Cambridge University Press

OpSim and CatSim evolve (new scheduler, new types of objects, better population of catalogs on the sky) the AlertSim will become better and closer to the reality.

Acknowledgements

Authors would like to thank for support members of LSST System Engineering Simulation Group - especialy S. Daniel, A. Connolly and G. Angeli. Also we would like to thank M. Jurić and Ž. Ivezić for help in definition of the problem and encouragement in the development process.

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia Grant III44002 Astroinformatics: Application of IT in Astronomy and Close Fields. D.J. and V.S. would like to thank COST Action TD1403 "Big Data in Sky and Earth Observations" for travel suport.

References

Delgado, F., Saha, A., Chandrasekharan, S., Cook, K., Petry, C., & Ridgway, S. 2014. The LSST operations simulator. Modeling, Systems Engineering, and Project Management for Astronomy VI 9150, 915015.

Ivezić, Ž., *et al.* 2008, ArXiv e-prints. `arXiv:0805.2366`

Ivezić, Ž. and the LSST Science Collaboration, 2011, LPM-17: LSST Science Requirement Document http://ls.st/srd

Jurić, M., *et al.* 2015 The LSST Management System, ArXiv e-prints. `arXiv:1512.07914`

Jurić, M., *et al.* 2016 LSST Data Products Definition Document, LSE-163, 2016.

Seaman, R., *et al.* 2011 Sky Event Reporting Metadata Version 2.0, http://www.ivoa.net/documents/VOEvent/

Swinbank, J. 2014, *Astronomy and Computing* 7 , 12.