Using business workflows to improve quality of experiments in distributed systems research

Tomasz Buchert, Lucas Nussbaum

Using business workflows to improve quality of experiments in distributed systems research

Tomasz Buchert
INRIA Nancy – Grand Est
Email: tomasz.buchert@inria.fr

Lucas Nussbaum
LORIA / Université de Lorraine
Email: lucas.nussbaum@loria.fr

***Abstract*—Distributed systems pose many difficult problems to researchers. Due to their large-scale complexity, their numerous constituents (e.g., computing nodes, network links) tend to fail in unpredictable ways. This particular fragility of experiment execution threatens reproducibility, often considered to be a foundation of experimental science.**

Our poster presents a new approach to description and execution of experiments involving large-scale computer installations. The main idea consists in describing the experiment as workflow and using achievements of Business Workflow Management to reliably and efficiently execute it. Moreover, to facilitate the design process, the framework provides abstractions that hide unnecessary complexity from the user.

I. Problem

The research in distributed systems is impeded by many technical problems:

- large, complex and unreliable communication networks,
- handling of heterogeneous infrastructure,
- software and hardware errors,
- shared computing environment.

There exist non-technical sources of problems as well:

- bad methodology,
- incomplete or missing description of the experiment,
- reluctance to share details of the work.

All these factors make the reproducibility of large-scale experiments a very difficult property to achieve. Without reproducibility, however, other researchers cannot obtain the same results or easily explore and extend ideas behind the experimental result. Actually, it happens that the researchers cannot reproduce their own results themselves (a property known as *repeatability*).

Luckily, the scientific community is aware of these problems and reproducibility of experiments gains more and more attention. For instance, a few major scientific publishers (e.g., Science[1]) expect the publication of programs used to obtain the scientific result. The necessity of source code "openness" seems to be an ongoing trend ([1]).

However, the availability of source code may not be enough. In case of large-scale experiments, a scientist interested in reproducing the result may not have access to large enough infrastructure, for example. Moreover, even if the whole experiment source code is available, the way it must be executed may not be provided. In fact, computer-based experiments suffer from the same problems as normal software does (lack of documentation, bugs, software dependencies, etc.).

One solution is to formalize the description of the experiment and automate its execution. There exist numerous tools in the domain of distributed systems: Expo [2], g5k-campaign[2], ZENTURIO [3], DART [4], Plush (Gush) [5], Weevil [6], Network eXperiment Engine[3] (NXE) and Experimentation workbench for Emulab [7], among others. The goal of these tools is to provide high-level description of the experiment and environment to execute it. Some of the solutions are dedicated to particular network testbeds – g5k-campaign is Grid'5000-specific tool, for instance.

In our approach, we formalize the experiment in the language of workflows. We draw our inspiration from Business Process Management (BPM), a domain that successfully helps with the coordination of distributed services. To achieve our goal, we built on the achievements of BPM and extend them to support our particular use case, i.e., orchestration and execution of complex, large-scale experiments.

II. Presented approach

Our method consists in harnessing methods of Business Process Management. More precisely:

- the experiment is expressed using a domain specific language (DSL) using experiment patterns and patterns identified in the domain of BPM,
- the functional parts of the experiment, or activities, implement atomic actions that realize a concrete and well-defined goal (e.g., installation of software on a computer node or execution of a command),
- finally, an experimentation engine executes the workflow and takes care of communication, data collection, monitoring, exceptional situations, etc.

The domain-specific language allows for semi-declarative description of the experiment that easily maps to a workflow representation (cf. an example on the poster). The language has restrictive scoping rules and does not allow for multiple assignments to one variable (an element shared with some functional languages like Erlang or Haskell). These restrictions allow for easy and efficient implementation of snapshotting.

[1] http://www.sciencemag.org/site/feature/contribinfo/prep/gen_info.xhtml

[2] http://g5k-campaign.gforge.inria.fr/

[3] http://ens-lyon.fr/LIP/RESO/Software/NXE/

Additionally, they simplify reasoning about experiment workflow, because side effects of the execution are limited.

The engine supports patterns identified in [8]. Nevertheless, we observed that some patterns present in experiments involving distributed systems are not well supported by existing BPM notations and software solutions. In particular, parallel execution of one task with different parameters known only at runtime, is not well supported. It is, however, a very common situation when some action has to be performed on multiple hosts in parallel (running a single command on all of them, for example).

Apart from the standard patterns, the engine features:

- integrated collection and analysis of results,
- external library loading system,
- robust error handling and retry policies,
- verbose logging and monitoring of activities.

Major features that at the moment of writing are still missing:

- snapshotting of the experiment (useful for debugging purposes),
- transparent execution of commands on remote nodes (using SSH transport),
- generating graphical representation of the workflow,
- visualization of experiment state,
- addressing possible scalability problems by distributing the execution of the experiment.

Note that the development is in an early stage, therefore details and features of the implementation are likely to change.

III. Advantages of a BPM-based approach

During our work, we identified the following requirements for an experiment description and its execution: descriptiveness, modularity, reusability, maintainability, support for common patterns, distributed execution of the experiment, snapshotting, error handling, integration with lower-level tools, monitoring, instrumentation and integrated data analysis. The assets of BPM approach address the aforementioned requirements and provide multiple advantages:

Understanding of the process
: The experiment description is more abstract, has a canonical graphical representation and therefore is easier to understand and improve.

Modularity
: Parts of the experiment can be reused and maintained separately. It is a large improvement over a standard approach of writing low-level scripts.

Monitoring
: The experiment execution and various metrics are continuously monitored. The user can use this information to spot optimization opportunities or to debug the experiment. If an activity within the experiment failed, it can be automatically restarted and the experiment will carry on. Moreover, with verbose error reporting it is easier to pinpoint bugs.

Workflow patterns
: Workflow-based description of business processes builds on common patterns. Although the common patterns differ between business processes and large-scale experiments, the missing patterns can be implemented.

IV. Future work

In the near future, we would like to concentrate our efforts on a complete implementation of an experiment engine according to our approach. Next, the engine has to be evaluated using exemplary experiments, to verify if it meets required criteria.

Our approach does not apply solely to the orchestration of the experiments. Other possible uses include:

- configuration management of computer nodes (cf. Chef and Puppet tools),
- orchestration of complex system installations,
- automation of complex, computer-based activities of a company,
- coordination of crowdsourcing.

V. Conclusion

Our poster presents a novel approach to orchestration of challenging, large-scale experiments. With the aid of Business Workflow Management, the experiment admits a semi-formal representation in the language of workflows. A special software is then responsible for the execution of the experiment. This new approach is a promising idea that needs further research.

References

[1] D. C. Ince, L. Hatton, and J. Graham-Cumming, "The case for open computer programs," *Nature*, vol. 482, no. 7386, pp. 485–488, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1038/nature10836

[2] B. Videau and O. Richard, "Expo : un moteur de conduite d'expériences pour plates-forme dédiées," in *Conférence Française en Systèmes d'Exploitation (CFSE)*, 2008.

[3] R. Prodan and T. Fahringer, "ZENTURIO: An Experiment Management System for Cluster and Grid Computing," in *Proceedings of the 4th International Conference on Cluster Computing (CLUSTER 2002)*. IEEE Computer Society Press, 2002, pp. 9–18. [Online]. Available: http://www.par.univie.ac.at/project/zenturio

[4] B. N. Chun, "DART: Distributed Automated Regression Testing for Large-Scale Network Applications," in *Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS)*, 2004.

[5] J. Albrecht, C. Tuttle, R. Braud, D. Dao, N. Topilski, A. C. Snoeren, and A. Vahdat, "Distributed Application Configuration, Management, and Visualization with Plush," *ACM Transactions on Internet Technology*, vol. 11, pp. 6:1–6:41, Dec. 2011. [Online]. Available: http://doi.acm.org/10.1145/2049656.2049658

[6] Y. Wang, M. J. Rutherford, A. Carzaniga, and A. L. Wolf, "Automating Experimentation on Distributed Testbeds," in *Proceedings of the 20th IEEE/ACM International Conference On Automated Software Engineering (ASE)*, ser. ASE '05. New York, NY, USA: ACM, 2005, pp. 164–173. [Online]. Available: http://doi.acm.org/10.1145/1101908.1101934

[7] E. Eide, L. Stoller, and J. Lepreau, "An Experimentation Workbench for Replayable Networking Research," in *Proceedings of the 4th Symposium on Networked System Design and Implementation (NSDI)*, 2007.

[8] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow Patterns," *Distrib. Parallel Databases*, vol. 14, no. 1, pp. 5–51, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1022883727209