

Premier rendu :
BIBLIOGRAPHIE & ÉTUDE DE L'EXISTANT

Salmane Bah & Tristan Braquelaire & Wassim Romdan & Timothée Sollaud

UNIVERSITÉ DE BORDEAUX

5 juillet 2014

Commentaire bibliographique

Dans le cadre de l'Unité d'Enseignement "Projet de Programmation", suivie durant la formation de Master 1 informatique de l'UNIVERSITÉ DE BORDEAUX, il nous a été proposé de réaliser une plate-forme de compilation à distance. Celle-ci doit être basée sur docker afin de sécuriser la compilation et l'exécution de code émis par un client distant. Ce type d'outil nécessite l'application d'un certain nombre de concepts liés à la sécurité et aux protocoles de communication.

Sécurité

La mise en place d'une telle plate-forme nécessite une sécurité de haut niveau afin de protéger le serveur de code à risque. Le principe de *sandbox* détaillé dans [GWTB96] & [Gal] permet d'isoler puis d'exécuter des programmes dans un espace virtuel clos, indépendant du système hôte.

L'isolement des processus peut-être assuré par l'utilisation des conteneurs (LXC Container) tel que décrit dans [Hel09]. L'avantage des conteneurs est qu'ils permettent d'utiliser directement le noyau du serveur hôte à la différence d'une machine virtuelle qui charge intégralement un noyau pour chaque client.

Docker [Inc] facilite la création et l'utilisation des conteneurs linux pour l'exécution sécurisée d'applications. La vidéo [SH] propose un exemple d'utilisation de l'outil.

La communication

Afin de réaliser les appels de procédures depuis un client distant, la mise en place d'une architecture RPC (**R**emote **P**rocedure **C**all) [Mic] est nécessaire.

Il existe une architecture client / serveur orientée objet appelée RMI (**R**emote **M**ethod **I**nvocation) [Fro12a].

Une des technologies web qui permet l'échange de données dans un environnement distribué est le web service [Kre01]. Un tel service peut être implémenté à l'aide d'une architecture REST [Sun] ou de requêtes SOAP [Sud03], [Fro12b].

Étude de l'existant

Plusieurs plate-formes de compilation et d'exécution similaires sont disponibles en ligne. Cependant, aucune ne répond entièrement à nos exigences. Voici une liste non exhaustive des plus populaires :

Ideone

<http://ideone.com/>

Avantages

- Traite un grand nombre de langage.
- Propose une API basée sur le protocole SOAP [Ide].
- Possibilité de télécharger le code source.
- Possibilité de rendre le code privé (pour les abonnés).

Inconvénients

- Nombre limité de requêtes gratuites.
- Traitement d'un unique fichier à la fois.
- Temps de compilation limité lors d'une utilisation gratuite.
- Espace mémoire relativement restreint lors d'une utilisation gratuite.
- Tout code compilé gratuitement est public.

Compileonline

<http://compileonline.com/>

Avantages

- Reconnaît un grand nombre de langage et de métalangage.
- Permet de traiter plusieurs fichiers selon le langage.
- Entièrement gratuit.
- Possibilité de télécharger le code source.
- Configuration du mode d'édition (Emacs / Vim).

Inconvénients

- Temps d'exécution limité à un nombre inconnu de cycle.
- Architecture des programmes à compiler limitée (induit par la restriction sur le nombre de fichiers).
- Ne propose pas d'API.

Complr

<https://compilr.com/>

Avantages

- Intégration d'un IDE complet.

Inconvénients

- Inscription obligatoire.

Bibliographie

- [Fro12a] Annick Fron. *Architectures réparties en Java*. DUNOD, 2e edition edition, 2012.
- [Fro12b] Annick Fron. *Architectures réparties en Java*. DUNOD, 2e edition edition, 2012.
- [Gal] Sarah Fister Gale. Creating a secure 'sandbox' on employee devices. <http://www.workforce.com/articles/creating-a-secure-sandbox-on-employee-devices>. Accédée : 17/01/2014.
- [GWTB96] Ian Goldberg, David Wagner, Randi Thomas, and Eric Brewer. A secure environment for untrusted helper applications confining the wily hacker. In *Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography - Volume 6*, SSYM'96, pages 1–1, Berkeley, CA, USA, 1996. USENIX Association.
- [Hel09] Matt Helsley. Lxc : Linux container tools. Technical report, IBM, 2009.
- [Ide] Ideone. ideone api specification. <https://ideone.com/files/ideone-api.pdf>. Accédée : 29/01/2014.
- [Inc] Docker Inc. Learn what docker is all about. http://www.docker.io/the_whole_story/. Accédée : 17/01/2014.
- [Kre01] Heather Kreger. Web services conceptual architecture (wsca 1.0). Technical report, IBM, 2001.
- [Mic] Microsoft. How rpc works. <http://technet.microsoft.com/en-us/library/cc738291> Accédée : 17/01/2014.
- [SH] dotcloud Solomon Hykes. The future of linux containers. <http://youtu.be/wW9CAH9nSLs>. Accédée : 17/01/2014.
- [Sud03] Brian Suda. Soap web services. Master's thesis, University of Edinburgh, 2003.
- [Sun] Bruce Sun. A multi-tier architecture for building restful web services. <http://www.ibm.com/developerworks/library/wa-aj-multitier/>. Accédée : 17/01/2014.