

Université Claude Bernard Lyon 1
Master Informatique
2015 - 2016

MIF-TI5

Projet PERF

Conception d'une plateforme pour l'alignement
d'entités par flux

Document Post-sprint 2

Alexandre Millot
Corentin Lonjarret
Antoine Hintzy
Murat Halat
Felician Raphael Matinya

Université Claude Bernard  Lyon 1

Table des matières

Projet PERF	1
1 Préambule	3
2 Pré-sprint	3
2.1 Travail effectué pré-sprint	3
2.2 Décomposition en tâches	3
2.2.1 Tâches et attentes sprint 2	3
Côté métier	3
Côté interface	3
Côté déploiement	4
Côté gestion de projet	4
2.2.2 Planning du sprint à la demi-journée	4
2.3 Évaluation des charges et répartition des tâches	5
2.4 Livrables techniques	5
3 Post-sprint	6
3.1 État d'avancement des tâches & écarts par rapport aux prévisions	6
3.2 Résultats du sprint et modifications apportées au projet	7
3.3 Objectifs de la prochaine itération	7

1 Préambule

Ce document présente la synthèse des activités réalisées ainsi que des méthodes employées pour mener à bien le sprint 2 du projet PERF.

Celui-ci décrit tout d'abord le travail effectué avant le sprint, les livrables techniques à produire pour la recette ainsi que la décomposition, la répartition et l'évaluation de la charge des tâches devant être réalisées au cours du sprint. Ensuite, une comparaison post-sprint est présentée de façon à évaluer les écarts entre les prévisions faites avant le sprint et les résultats réels fournis en sortie de ce dernier. Enfin, nous discutons des résultats obtenus et des modifications apportées au projet ainsi que des objectifs de la prochaine itération.

2 Pré-sprint

2.1 Travail effectué pré-sprint

Plusieurs tâches ont été réalisées pré-sprint de façon à être au point dès le début du sprint et à assurer le bon avancement du projet.

Parmi les tâches effectuées, on retrouve notamment :

- Création d'un nouveau dépôt sur Bitbucket (utilisation de git flow).
- Préparation des tâches du sprint
- Séparation du code en *master/worker* pour faciliter la mise en place du futur protocole d'échange *client-master-worker*.

2.2 Décomposition en tâches

2.2.1 Tâches et attentes sprint 2

Côté métier

- Gestion de l'envoi de fichiers sur le master lors de l'ajout d'un *dataset*.
- Mise en place d'un *worker* & du *master* (au départ, un *worker* et un *master*).
- Envoi d'un message du *worker* concernant la date de début de travail du client.
- Calcul du temps avec la réception des résultats du client par le master.
- Implémentation de plusieurs *workers* (deux dans un premier temps).
- Gestion d'une file d'attente de clients avec plusieurs *workers* et plusieurs clients.
- Calculer et ajouter en base les résultats des clients.

Côté interface

- Prototype des interfaces.
- Implémentation des interfaces.
- Création des tutoriels.

Côté déploiement

- Version du sprint 1 fonctionnelle sur la VM (en mode production).
- Commencer à utiliser Docker.

Côté gestion de projet

- Document post-sprint 2.
- Document d'Architecture Logiciel version 2.

2.2.2 Planning du sprint à la demi-journée

Lundi Matin

- Gestion de l'envoi de fichiers sur le *master*.
- Mise en place du *master* et d'un *worker*.
- Version du sprint 1 fonctionnelle sur la VM.

Lundi Après-midi

- Gestion de l'envoi de fichiers sur le *master*.
- Calculer et mettre en base les résultats.
- Mise en place du *master* et d'un *worker*.
- Mise en place d'une seconde VM.

Mardi Après-midi

- Prototypes des interfaces.
- Mise en place du *master* et d'un *worker*.
- Calculer et mettre en base les résultats.

Mercredi Matin

- Envoi d'un message du *worker* concernant la date de début de travail du client.
- Mise en place du *master* et d'un *worker*.
- Réunion du code des différentes parties et tests.
- Implémentation des interfaces.

Mercredi Après-midi

- Mise en place des tutoriels administrateurs et utilisateurs.
- Réunion du code des différentes parties et tests.
- Implémentation de plusieurs *workers*.

Jeudi Matin

- Début de rédaction du Document d'Architecture Logiciel.
- Implémentation des interfaces.
- Implémentation de plusieurs *workers*.

Vendredi Matin

- Implémentation des interfaces.
- Rédaction du document post-sprint 2.
- Gestion d'une file d'attente de clients pour plusieurs clients avec plusieurs *workers*.
- Bonus : « Dockerisation ».

2.3 Évaluation des charges et répartition des tâches

La charge des tâches est exprimée en demi-journée.

Intitulé de la tâche	Évaluation de la charge	Personnes affectées
Version du sprint 1 fonctionnelle sur la VM	1	Antoine
Mise en place d'une seconde VM	1	Antoine
« Dockerisation »	(1)	Antoine, Corentin
Gérer l'envoi de fichiers sur le <i>master</i>	1.5	Murat, Felician
Prototype des interfaces	1	Antoine
Implémentation des interfaces	3	Antoine
Mise en place d'un <i>worker</i> & du <i>master</i>	3.5	Corentin, Alexandre
Envoi d'un message du <i>worker</i> concernant la date de début de travail du client	0.5	Murat, Felician
Implémentation de plusieurs <i>workers</i>	2	Corentin, Murat, Felician
Gestion d'une file d'attente de clients pour plusieurs clients avec plusieurs <i>workers</i>	1	Corentin, Murat, Felician
Calculer et mettre en base les résultats	1.5	Murat, Felician
Mise en place tutoriels	1	Antoine, Alexandre
Réunion des différentes parties	0.5	Corentin, Murat, Felician, Alexandre
Document post-sprint 2	1	Alexandre
Document Architecture Logiciel v2	1	Alexandre

2.4 Livrables techniques

Parmi les livrables techniques que nous devons réaliser au cours du second sprint, on retrouve :

- Le produit logiciel : Version fonctionnelle avec plusieurs *workers* et capable de gérer plusieurs clients en parallèle.
- La seconde version du document d'architecture logiciel (DAL).

Ces deux livrables seront présentés aux porteurs du projet, c'est-à-dire M. DUCHATEAU et M. LUMINEAU lors d'une réunion le lundi 7 décembre.

3 Post-sprint

3.1 État d'avancement des tâches & écarts par rapport aux prévisions

Intitulé de la tâche	Évaluation charge	Charge réelle	État d'avancement
Version du sprint 1 fonctionnelle sur la VM	1	annulée	annulée
Mise en place d'une seconde VM	1	1	100 %
« Dockerisation »	1	3	100 %
Gérer l'envoi de fichiers sur le <i>master</i>	1.5	2	100 %
Prototype des interfaces	1	1	100 %
Implémentation des interfaces	3	1	50 %
Mise en place d'un <i>worker</i> & du <i>master</i>	3.5	1.5	100 %
Implémentation de plusieurs <i>workers</i>	2	0.5	100 %
Gestion d'une file d'attente	1	2	100 %
Calculer et mettre en base les résultats	1.5	0.5	100 %
Mise en place tutoriels	1	1	100 %
Première réunion et code et tests	0.5	0.5	100 %
Modification fichier <i>worker.txt</i> sur site	non prévue	0.5	100 %
Ajout fonctionnalités <i>datasets</i>	non prévue	1	100 %
Ajout date début et date fin dans les <i>datasets</i>	non prévue	1	100 %
Déploiement <i>master</i> sur VM	non prévue	1	90 %
Document post-sprint 2	1	1.5	100 %
Document architecture logicielle	1	0.5	10 %

3.2 Résultats du sprint et modifications apportées au projet

La majeure partie des tâches que nous avons prévu de réaliser a pu être implémenté. La seconde version de la plateforme est donc fonctionnelle.

Ainsi, en plus des fonctionnalités réalisées lors du sprint 1, la plateforme implémente maintenant d'autres fonctionnalités permettant le déploiement de plusieurs *workers* ainsi que la gestion de la prise en charge de plusieurs clients. Pour gérer cet ensemble de clients, un système de file d'attente a été mis en place au niveau des *workers* dans le cas où aucun des *workers* ne serait disponible au moment où un des clients enverrait une demande d'envoi d'entités. L'implémentation des interfaces est quasiment complète, seule la disposition et le CSS restent à retravailler lors de la prochaine itération.

En plus des tâches que nous avons prévu de réaliser, nous avons eu l'occasion de mettre en place certaines fonctionnalités supplémentaires que nous avons prévu d'implémenter en tant qu'améliorations :

- La modification du fichier *worker.txt* contenant la liste des *workers* actifs directement depuis l'application.
- La modification ainsi que la suppression d'un *dataset* directement depuis l'application.
- La simulation d'un mode « projet » avec l'ajout d'une date de début et d'une date de fin pour l'envoi de résultats de la part du client. Ainsi, un client ne peut soumettre des résultats au serveur que si la date à laquelle il retourne ces résultats se trouve entre la date de début du et la date de fin du projet.
- La « dockerisation » des machines virtuelles.

Concernant les modifications apportées durant le sprint, nous avons décidé de supprimer la tâche concernant la mise en place de la version 1 du sprint sur une VM pour nous concentrer sur la « dockerisation » des machines virtuelles et le déploiement des *masters* et *workers* sur ces dernières. De ce fait, nous avons décidé de passer moins de temps sur l'aspect visuel des interfaces pendant ce sprint et d'y revenir au prochain sprint.

3.3 Objectifs de la prochaine itération

On retrouve ci-dessous la liste des tâches prévues pour la prochaine itération du projet. Le but est d'obtenir la version finale et fonctionnelle du système facilement déployable sur les VMs fournies. Pour le prochain sprint, nous souhaitons donc mettre en place :

- Le déploiement facile du *master* et de plusieurs *workers* sur les VMs.
- Des tests de montée en charge. Le but est de tester la limite de charge que peut supporter un *worker* ainsi que de définir le nombre maximal de clients que le système est capable de prendre en charge.
- La modification de l'API Java. Actuellement, les clients doivent modifier de façon directe certaines classes de notre projet java dédiées au traitement des entités et à la communication avec le serveur. Le but est de fournir à la place un jar qui se présentera alors comme une librairie. Ainsi, les clients n'auront qu'à appeler les fonctions à partir de la librairie pour effectuer les traitements nécessaires.

- Finalisation de l'implémentation des interfaces
- Améliorations : Différentes petites améliorations à ajouter au projet, notamment d'exports des résultats (classements, listes de performances) et d'ajout de types pour les *datasets*.
- L'ensemble de la documentation.