

Evaluation et mise en place d'un serveur de message pour Chamilo

Etudiant: Daniel Stanculescu

Coordonateurs: Laurent Opprecht, Patrick Roth

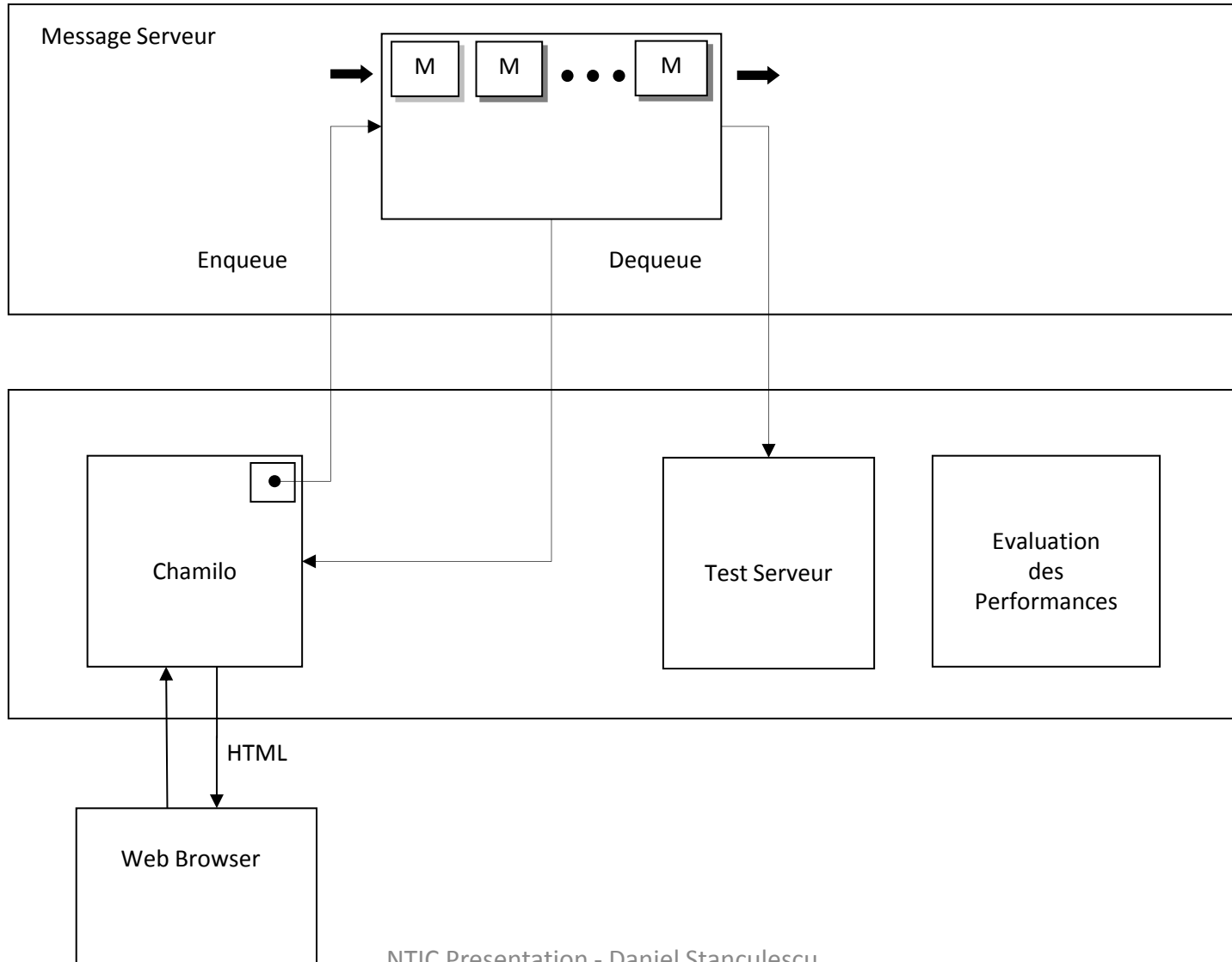
Plan

- Introduction
- Schéma et fonctionnement
- L'interface Message Serveur
- Les approches de stockage
- Evaluation et performances
- Remarques

Introduction

- Le but est d'améliorer la montée en charge et les performances pour la partie « event » de Chamilo/Dokeos.
- Le sous-système « statistique » de Dokeos consomme beaucoup de ressources.
- Pour améliorer cette situation on implémente un serveur de message pour la partie statistique de son successeur : Chamilo.
- On peut stocker les données dans une queue et les traiter.

Schéma



Fonctionnement

- On crée des événements, en appelant une méthode.
- Ces événements sont détectés par Chamilo qui appelle le Message Server pour enqueue le message.
- Message server fait appel à la factory pour utiliser la méthode qu'on désire.
- Les messages sont enregistrés, en respectant la forme FIFO.
- On utilise dequeue pour récupérer les messages.
- Les testes de performance ont été faites pour toutes les méthodes, en utilisant 10, 100 et 1000 événements.

L'interface Message Serveur

Le serveur de message est une classe abstraite qui a les fonctions suivantes (**message_server.class.php**):

- get_instance
- enqueue(message)
- dequeue
- factory(method)

Pour vérifier le code on utilise la classe:

test_serveur.php

Pour évaluer les performances on utilise la classe:

test_performances.php

Les Approches/Méthodes

- Empty
- Memoire
 - array_push, array_shift.
- Fichiers
 - stockage de la forme suivante:
message_server_file_<MICROTIME>_<# EVENT>_<USER>.txt
- Fichiers buffer
 - buffer longueur fixe, après on écrit dans les fichiers.
- Fichiers APC
 - la mise en cache et l'optimisation de code intermédiaire PHP.
- Existante dans Chamilo.

Performance et evaluation

	Empty	Mémoire	Fichiers	Fichiers Buffer	Fichiers APC	Default
10 messages	0.001214	0.001234	0.015067	0.001362	0.002115	0.397381
100 messages	0.011837	0.013200	0.165964	0.082123	0.082476	1.525664
1000 messages	0.130979	0.136307	1.410019	0.783486	0.807286	14.197292

	Empty	Mémoire	Fichiers	Fichiers Buffer	Fichiers APC	Default
Empty	1	1.04067	12.411037	6.937822	6.967643	128.889414
Mémoire	0.983792	1	12.573030	6.221439	6.248181	115.580606
Fichiers	0.071322	0.079535	1	0.494824	0.496951	9.192740
Fichiers Buffer	0.144137	0.160734	2.020919	1	1.004298	18.577791
Fichiers APC	0.143520	0.160046	2.012270	0.995719	1	18.498278
Default	0.007758	0.008651	0.924200	0.513537	0.529137	1

Remarques

- Une solution FIFO peut être implémenté avec `posix_mkfifo`, mais cette fonction appartient à une librairie UNIX.
- L'utilisation de la factory.
- Portfolio c'est une solution très bonne pour garder le projet et avoir une trace de tout que j'ai travaillé.
 - <https://portfolio.unige.ch/view/view.php?id=241>

Questions?