Université de Lille

OPL projet : Green-Computing

Comparaison consommation CPU JEE vs NodeJS

Authors:
Benhammou A.

Source github : WebserverConsomationAnalyse

January 18, 2016

Contents

1 Introduction

1.1 Contexte

La réduction de la consommation des machines informatiques est un des nouveaux enjeux du 21e siècle, le nombre croissant de datacenter[1] impose aux grandes entreprises et aux gouvernements de réfléchir à la minimisation de la consommation de ces centres de données. Il peut être intéressant d'analyser plus particulièrement la consommation CPU[2] des serveurs Web, car elles représentent une grande partie des consommations au niveau mondial.

1.2 problème

La consommation CPU est directement liée aux nombres d'instructions qu'il exécute, les serveurs Web[3] ont donc des consommations variables selon l'application qu'elle héberge et la technologie utilisée . Une question à se poser est quel serveur Web adopté d'un point de vue économie d'énergie.

1.3 objectif

Pour répondre à cette question, l'étude sera faite sur deux technologies bien distinctes d'une part Java EE[4]/TomCat[5] qui est une technologie historiquement fiable et robuste basée sur le langage Java et d'autre part NodeJs[6] qui est une technologie montante basée sur le traitement asynchrone d'évènement. L'article présentera les consommations détaillées des deux technologies.

2 Travail technique

2.1 Vue d'ensemble

L'étude analyse la consommation électrique en milliwatt à l'aide de l'application POWERAPI (https://github.com/Spirals-Team/powerapi). Cet outil calcul la consommation CPU d'une application donnée à une fréquence de capture paramétrable. Afin d'obtenir des valeurs de référence et sur des échelles variables et réalistes, l'étude a capturé l'énergie utilisée sur les deux technologies sur 3 périodes avec des sollicitations du serveur différentes:

- Première période : Une requête émis au serveur afin de connaitre la valeur unitaire de chaque technologie.
- deuxième périodes : 10 requêtes pour connaitre la tendance des serveurs à plusieurs sollicitations.
- Troisième périodes : 1000 requêtes afin d'avoir la consommation dans des conditions réelles.

2.2 Choix de l'algorithme

Il est important pour ne pas fausser l'étude, de solliciter les serveurs sur des algorithmes de complexité similaire. De plus, l'algorithme doit être assez complexe pour que l'application POWERAPI est le temps de capturer la consommation, car il lui faut un minimum de 50ms de calcul.
Algorithme choisi:

Data: Integer i,Integer j, Integer result
Result: Integer : result
result = 1;
for *i = 0 to 10 000 step 1* **do**
  for *j = 0 to 100 step 1* **do**
    result = result * j;
  end
end

Algorithm 1: 10 000 fois factorielle 100

Complexité : O(n!)*10000 = 100! * 10000

2.3 Fréquence

Le choix de la fréquence est également important, car plus la fréquence est faible,plus POWERAPI à du mal à capturé la consommation et au contraire, plus la fréquence est élevée, plus il y a de bruit dans la consommation ce qui signifie qu'elle est moins représentative de la réalité.

Exemple concret de fréquence et consommation capturée pour un même programme:

- 250ms capture 171 mW
- 100ms capture 5837 mW
- 50ms capture 21623 mW

Un bon compromis est 50ms, car il est le plus proche de la réalité sans poser problème à la récolte des comsommations.

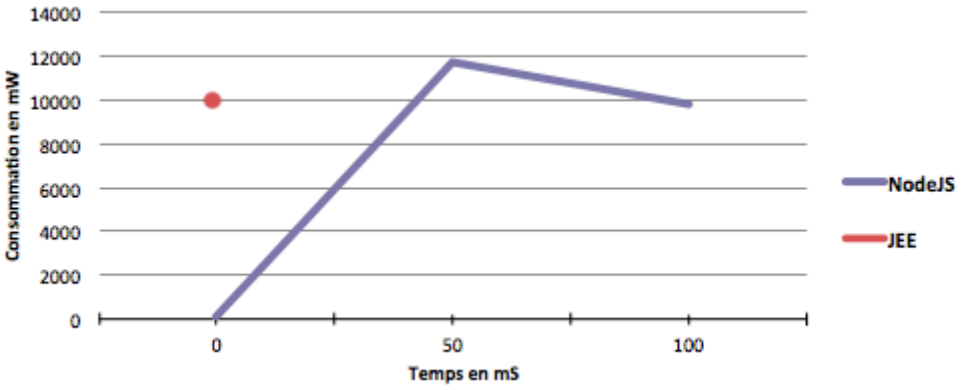
2.4 Architecture

Afin de faciliter le lancement des requêtes sur les serveurs, il a été choisi de les lancer par requête AJAX via un script js. Ce script lance en moyenne 1000 requêtes en 5 secondes.

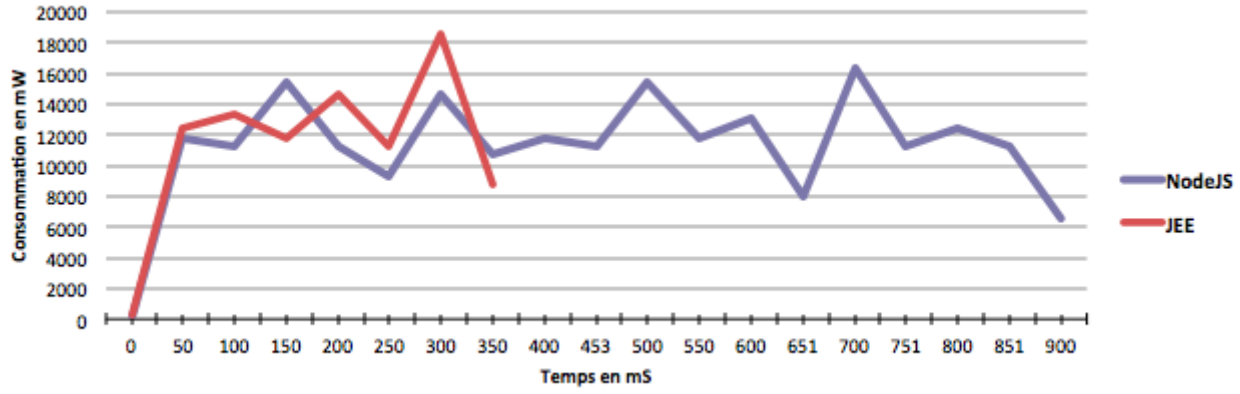
3 Evaluation

3.1 Résultat

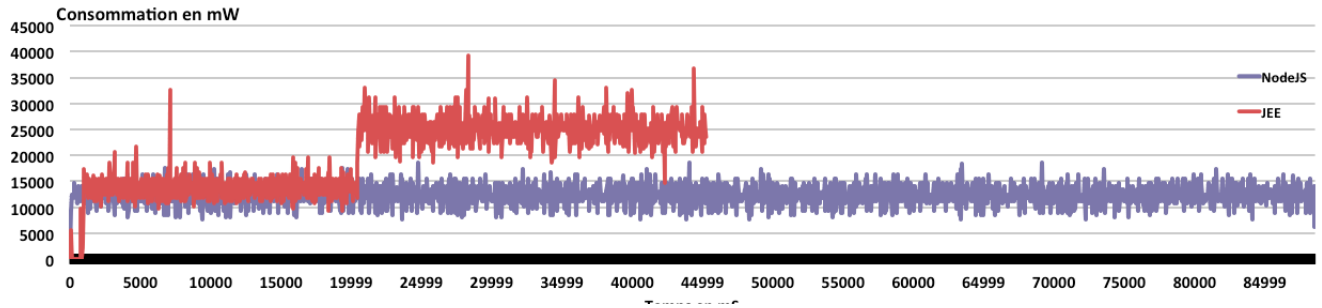
3.1.1 Consommations des deux applications soumit à une requête



3.1.2 Consommations des deux applications soumit 10 requête



3.1.3 Consommations des deux applications soumit 1000 requête



3.1.4 Consommations d'énergie

Technologie	JEE	NodeJs
Consommation Kwh	**0,2927**	0,5349

3.2 Analyse

Il apparait rapidement que la durée de traitement des requêtes est très différente sur les deux technologies qu'un coté NodeJs qui a un temps de traitement plus lent 8,5 seconde, mais des consommations très constante et faible et d'un autre coté JEE qui traite les requêtes rapidement 4,5s mais à des amplitudes de consommation énergetique beaucoup plus élevé. Malgré cette consommation plus élevée et variable, JEE reste la technologie la plus économe, car avec son temps de réponse faible, elle consomme deux fois moins de Kwh que NodeJs.

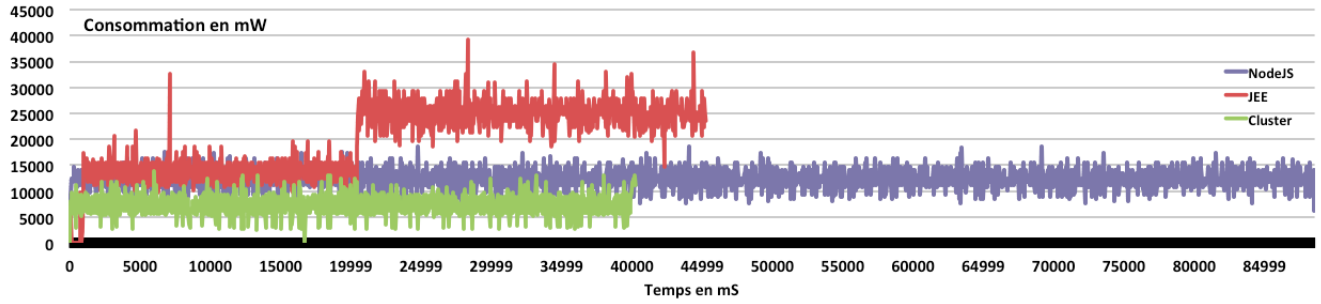
3.3 Confirmation des résultats

La consommation linaire de Nodejs vient du fait que l'application est mono-thread ce qui l'oblige à traiter les requêtes les une à la suite des autres et de garder une sollicitation du CPU constante. Alors qu'au contraire Tomcat étant une technologie multi-Thread, il traite les requêtes en parallèle et sollicite fortement le CPU dans un intervalle plus cours.

3.4 Pour aller plus loin

Node Js est depuis peu capable d'être multi-thread (dans la limite d'un thread par cœur CPU). Il est donc intéressant d'étudier Node avec cette configuration. Il en ressort en appliquant la même étude, que Node Js(multi-thread) est plus performant et économe en énergie quand mono-thread, mais également plus économe que JEE.
** Etude faite avec avec 4 thread*

3.4.1 Consommations des trois applications soumisent à 1000 requêtes



3.4.2 Consommations d'énergie

Technologie	JEE	NodeJs(mono-thread)	NodeJs(multi-thread)
Consommation Kwh	0,2927	0,5349	**0,1605**

En effet, NodeJs acquiere un temps de réponse beaucoup plus faible tous en gardant sa consommation énergetique plus faible.

4 Conclusion

NodeJS Muti-thread n'est pas une technologie réellement utilisé malgré, c'est réel avantage de temps d'exécution et utilisation CPU. La technologie Jee est donc le vainceur de cette étude, car il consomme deux fois moins que nodeJS et pendant un laps de temps beaucoup plus faible.
Une étude approfondie de nodeJs Multi-thread serai intéressante afin de déterminer pourquoi il consomme si peu d'énergie.

5 Lexique

**source Wikipédia*

1. **DataCenter** : Un centre de données ou data center est un site physique sur lequel se trouvent regroupés des équipements constituants du système d'information de l'entreprise (ordinateurs centraux, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.).
2. **CPU** :Un processeur (ou unité centrale de traitement, UCT, en anglais central processing unit, CPU) est un composant présent dans de nombreux dispositifs électroniques qui exécute les instructions machine des programmes informatiques.
3. **serveur Web** : Un serveur Web est un serveur informatique utilisé pour publier des sites web sur Internet ou un intranet. L'expression « serveur Web » désigne également le logiciel utilisé sur le serveur pour exécuter les requêtes HTTP.
4. **JAVA EE** : Java Enterprise Edition, ou Java EE (anciennement J2EE), est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux.
5. **TomCat** : Apache Tomcat est un conteneur web libre de servlets et JSP Java EE.
6. **NodeJS** : Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.