



Institut Supérieur de l'Electronique et du Numérique
Deuxième année de Cycle Informatique et Réseaux

Rapport de stage en entreprise

Stage de 7 semaines

Maître de stage : Philippe Le Goff

Pierre Zemb



Analyse et mise en oeuvre de métriques techniques sur des
environnements virtualisés VMWare et serveurs Windows

Relecq-Kerhuon, de Juin à Juillet 2013

Remerciements

Je tiens tout particulièrement à remercier :

Mon maître et tuteur de stage **Philippe Le Goff**, responsable de l'équipe système pour son accueil et sa gentillesse de m'avoir accueilli au sein de son équipe, et de m'offrir la possibilité d'élargir mes compétences par un contrat de professionnalisation dès le mois d'Octobre prochain.

Steven Le Roux, ancien intervenant Cisco à l'ISEN, pour m'avoir mis en contact avec Monsieur Le Goff.

Yann Morice et **Mathieu Bernard**, pour leur patience et leur aide face à mes nombreuses questions.

Lorelei Simon, actuelle alternante au Crédit Mutuel Arkea, pour m'avoir proposé du covoiturage avec elle.

Ainsi que toute l'équipe Système pour leur accueil.

Table des matières

I)	Présentation de l'entreprise	4
a)	Historique	4
b)	Quelques chiffres	4
c)	Secteurs d'activité	5
d)	Produits et services	5
e)	Organisation du département Technologies	6
f)	L'équipe Système	6
II)	La vie en entreprise	8
a)	La localisation	8
b)	Les horaires	8
c)	Déroulement du stage	8
III)	Infrastructure du système d'information	10
a)	Vue générale	10
b)	Présentation de Artimon	10
c)	Présentation de Flume	11
d)	Présentation de Kafka	11
e)	Présentation de Zookeeper	12
f)	Présentation des VHMS	12
g)	Présentation de Hadoop	13
h)	Présentation de l'infrastructure virtuelle VMware	14
IV)	Contexte du stage	16
a)	Contexte	16
b)	Missions	16
V)	Détails des missions	17
a)	Portage de Flume	17
b)	Création de métrique Windows	18
c)	Créations de métriques sur plateforme virtualisée VMWare	19
VI)	Bilan	20
VII)	Annexe	21
a)	Schéma de fonctionnement Artimon	21
b)	Artimongraph	22
c)	Photo de rack	23
d)	Exemple de variable Artimon Windows	24
e)	Exemple de variables Artimon issues de VM	25
f)	Exemple de variables Artimon issues d'host	26

I) Présentation de l'entreprise

a) Historique

Le Groupe Arkea a été formé en 2002, mais son histoire commence en 1979.

- 1979 : Cinq caisses bretonnes du Crédit Mutuel se regroupent et forment le Crédit Mutuel de Bretagne.
- 1980 : La création de filiales spécialisées démarre.
- 1996 : Le CMB et le CMSO se rassemblent et créent la CICM.
- 2002 : Le CMMC rejoint la CICM, qui devient le Crédit Mutuel Arkéa, et absorbe la Compagnie Financière du Crédit Mutuel (CFCM).

b) Quelques chiffres

Le groupe Arkéa c'est :

- 3,2 millions de sociétaires et clients
- 9 000 salariés
- 3 700 administrateurs
- 4,7 milliards d'euros de fonds propres
- 91 milliards d'euros de total de bilan
- 2200 serveurs
- 2 serveurs mainframe qui sont des ordinateurs centraux IBM
- 500 switches et une centaine de routeurs
- 40 To de données primaires¹
- 1526 automates...

1. données liées aux comptes bancaires

c) Secteurs d'activité

Arkéa est un groupe coopératif et mutualiste, réunissant les trois fédérations historiques, de Crédit Mutuel de Bretagne (CMB), du Sud- Ouest (CMSO) et du Massif Central (CMMC) autour de la caisse de Crédit Mutuel.

Cet ensemble contrôle la Compagnie Financière du Crédit Mutuel, banque de marché et société-holding rassemblant une vingtaine de filiales spécialisées couvrant les domaines de l'immobilier, du tourisme et du voyage d'affaires pour les particuliers et les professionnels.

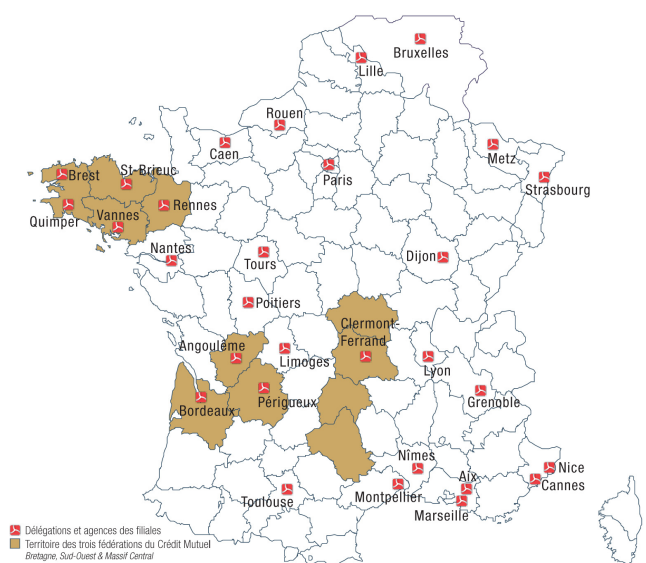


FIGURE 1 – Carte représentant les emplacements des filiales

d) Produits et services

Ouvert à tous, le Groupe Arkea est, par les Fédérations de Crédit Mutuel qui le composent et par ses filiales spécialisées, une banque qui se veut universelle.

De l'épargne la plus modeste à la gestion des hauts patrimoines, des crédits aux familles au financement des entreprises et de l'économie, il propose une gamme de produits et de services couvrant l'éventail des besoins de toutes les catégories de population, des particuliers jusqu'aux principaux acteurs économiques et institutionnels.

Afin de disposer d'une offre commerciale complète et évolutive, le Groupe Arkea s'appuie sur l'expertise et les savoir-faire complémentaires d'une vingtaine de filiales spécialisées qui optimisent les synergies dans divers domaines : les prestations aux entreprises, la gestion d'actifs, l'assurance, l'immobilier, etc.



FIGURE 2 – Logos des différentes filiales du groupe

e) Organisation du département Technologies

Lors de ce stage j'ai intégré le département architecture et technologie et plus précisément, l'équipe Système, sous la tutelle de Philippe Le Goff, responsable du service.

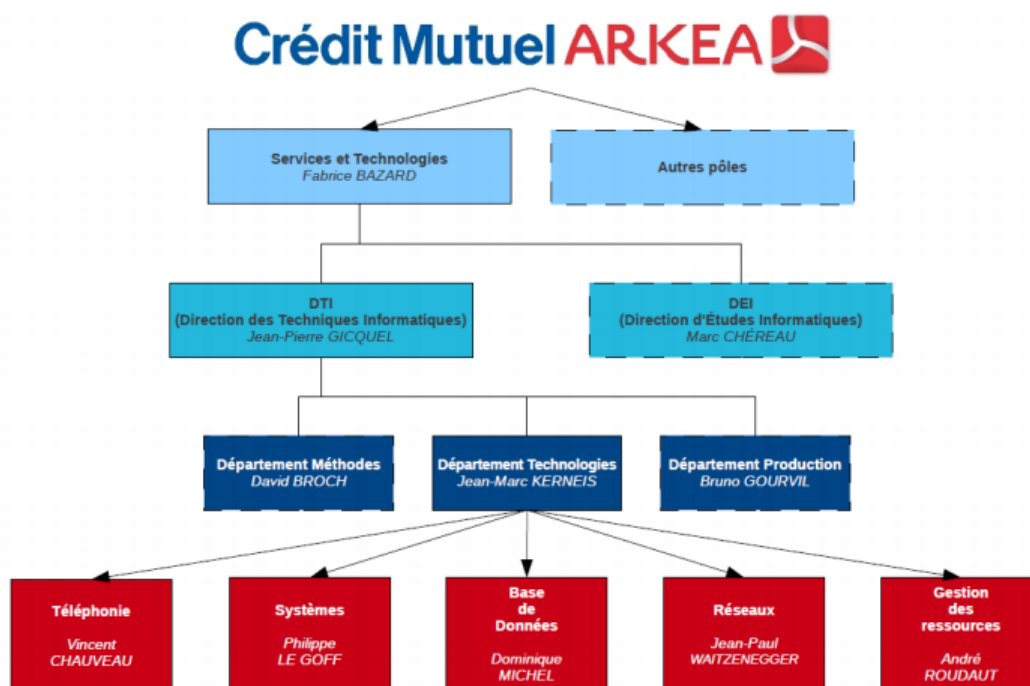


FIGURE 3 – Organigramme simplifié

f) L'équipe Système

L'équipe Systèmes est composée uniquement de personnes de niveau ingénieur et comprend 3 équipes réparties dans un grand bureau en open-space :

Equipe Windows

- Gestion des masters des postes de travail
- Mise en place de solution de virtualisation des postes
- Gestion de l'architecture de l'Active Directory 1

Equipe Unix

- Gestion des masters d'OS Linux/Solaris/AIX
- Baies de stockage
- Développement d'outils de supervision

Equipe MainFrame

- Gestion de l'OS Mainframe (zOS) et de ses composants

Il faut savoir que l'équipe s'occupe du support de niveau 2, il y a toujours une personne d'astreinte ou de garde. En effet, si un serveur plante, c'est tout une partie des activités de l'entreprise qui pourrait être bloquée, d'où la nécessité d'agir rapidement.

J'étais membre de l'équipe Windows pendant la totalité de mon stage.



FIGURE 4 – Photo d'un Z10, un des 2 serveurs principaux

II) La vie en entreprise

a) La localisation

J'ai effectué mon stage au siège social du Crédit Mutuel Arkéa, situé au Relecq-Kerhuon près de Brest. N'ayant pas encore le permis de conduire, j'ai eu la chance de trouver du covoiturage avec une collègue alternante de mon service.

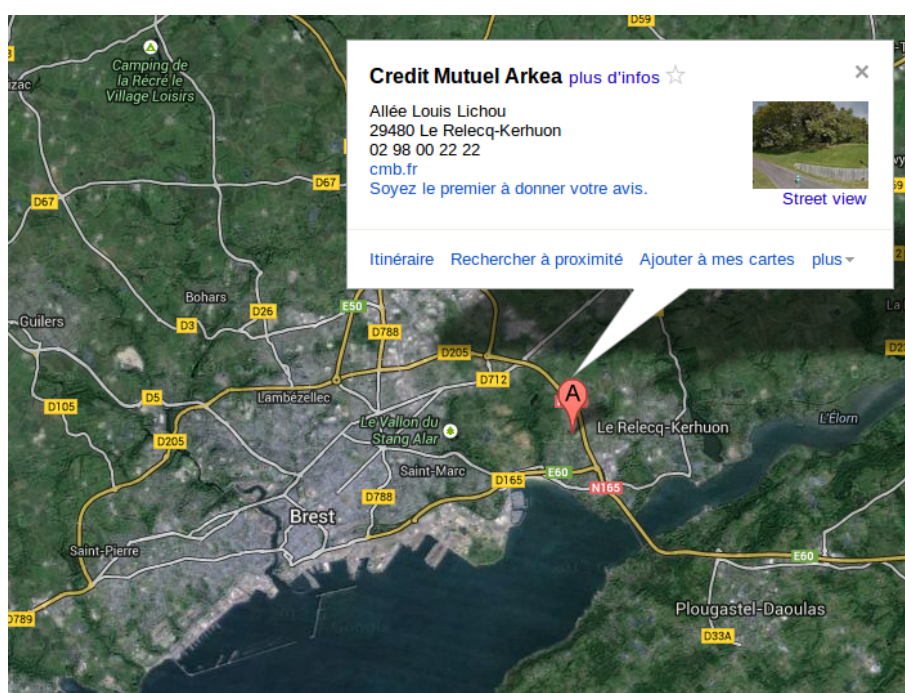


FIGURE 5 – Emplacement du siège d'Arkéa

b) Les horaires

Je n'avais pas réellement d'horaires, dans le sens où ma seule contrainte était de faire 35h entre le lundi matin et le vendredi midi car j'étais libéré l'après-midi. Généralement je démarrais à 8h30 le matin, je m'accordais une petite demi-heure pour manger avec mon service le midi au restaurant d'entreprises interne à Arkéa, et je terminais entre 17h et 18h.

c) Déroulement du stage

J'ai commencé mon stage le lundi 10 juin 2013 à 9h. J'ai rapidement fait connaissance avec mon tuteur, Monsieur Le Goff qui m'a présenté l'entreprise, puis l'infrastructure d'Arkéa, et enfin le rôle de mon stage. J'avoue avoir eu des difficultés à tout comprendre, dans le sens où le nombre de technologie utilisé par Arkéa est assez important. Il m'a également présenté Maïa, l'intranet de l'entreprise, qui est une véritable mine d'or.

J'ai passé les premiers jours de mon stage sur Maïa, afin d'être un peu moins perdu et de pouvoir comprendre les différentes technologies que je pourrais rencontrer. On me parlait de VHMS, Artimon, Zookeeper, de Hadoop et d'autres technologies que je ne connaissais pas. Curieux de nature, j'ai passé beaucoup de temps à me documenter sur des choses qui ne me concernaient pas directement, mais dont j'entendais parler.

J'ai également dû me mettre à niveau pour la manipulation de Linux, bien que j'eusse quelques connaissances, il me manquait certaines bases telles que "ssh", "patch" et autres commandes légèrement plus avancées que "ls" ou "cd". J'ai vraiment apprécié travailler dans un environnement Linux même si je développais pour Windows, cela m'a permis de gagner une certaine aisance dans un environnement que je voudrais savoir manier et dont j'ai l'intention d'utiliser au quotidien. Cette aisance me sert actuellement durant les cours d'administration Linux de cette année.

Une fois prêt, j'ai attaqué mes missions. J'ai d'abord porté un logiciel appelé Flume de Linux vers Windows, j'ai géré sa mise en production également. Puis j'ai créé différentes métriques venant de différents environnements. Le détail de ces missions se trouvent page 17. Dans l'ensemble, je n'ai pas eu de problème majeur durant mes missions, mis-à-part que je ne connaissais pas les technologies utilisées, je devais donc apprendre à les manier avant de pouvoir réaliser quelques choses.

Je me suis vraiment bien entendu avec toute l'équipe, notamment avec des anciens iséniens Yann Morice et Julien Girard mais aussi avec mon tuteur Philippe Le Goff, avec qui je discutais de nombreuses choses comme par exemple de développement, ou d'autres technologies utilisées par Arkéa. J'ai également rencontré des personnes en dehors de mon service, comme Horacio Gonzalez, avec qui j'ai négocié un partenariat entre le Club Electronique/Informatique de l'ISEN et le FinistGDG et le le FinistJUG, 2 groupes de développement situés à Brest.

III) Infrastructure du système d'information

a) Vue générale

Toutes les données sont réparties entre 4 dataCenters :

- Brest
- Rennes
- Angers
- Paris

Le site de Rennes est le plus impressionnant, car il contient ce qu'on appelle le Mainframe qui sont les 2 serveurs gérant les comptes des clients. Situés dans 2 salles différentes (Junon et Vulcain), ils sont fabriqués par IBM. Le groupe loue l'utilisation de ces machines avec la Bred.

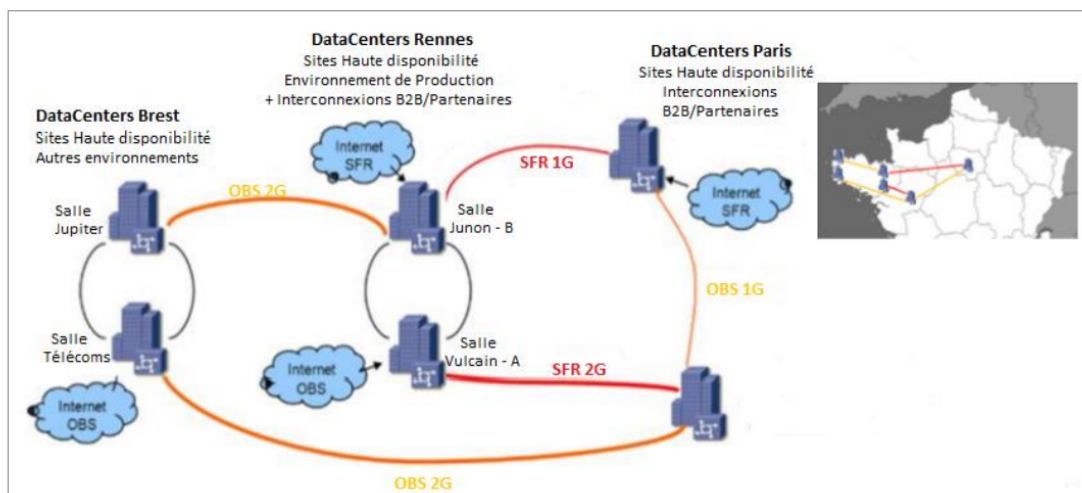


FIGURE 6 – Emplacement des data-centers du groupe

b) Présentation de Artimon

Artimon (Arkéa Real Time Information MONitoring) est un framework de monitoring destiné à remonter des métriques. La détection, le diagnostic et la prévention des incidents sont primordiales pour une société comme le groupe Arkea.

Ces métriques peuvent ensuite être analysées, individuellement ou en masse, afin de dégager des tendances, analyser des comportements, voire calculer d'autres métriques.

Elle repose sur quatres infrastructures :

- Zookeeper
- Kafka
- Flume
- VHMS
- Hadoop

Le framework Artimon manipule des variables labellisées et typées. Il se présente sous la forme suivante :

`variable{label1=valeur1,label2=valeur2,...} valeur`

La labellisation des variables consiste en l'association à une variable donnée d'un ensemble de clefs/valeurs destiné à la qualifier. Par exemple, il n'est pas nécessaire de mettre l'adresse IP de la machine en label, Artimon générera directement le label.

Vous pouvez retrouver des exemples de métriques Artimon issus de Windows que j'ai généré en Annexe page 24.

Un schéma résumant l'agencement des briques est disponible en Annexe page 21

Je vais maintenant présenter chaque technologie.

c) Présentation de Flume



FIGURE 7 – Logo de Flume

Flume est une solution logicielle en Java permettant l'acheminement d'évènements d'un point A à un point B. Flume est une solution Open Source dont le développement est conduit par Cloudera. Au sein du groupe Arkéa, Flume est utilisé pour l'acheminement des métriques Artimon et des logs Syslog et log4J vers des serveurs de collecte centralisés de type Kafka.

Localement à chaque machine, un daemon Flume s'exécute en permanence. Des extensions propres à Arkéa sont installées, elles permettent notamment la génération de métriques systèmes, le relai des messages Syslog et log4j et la collecte de métriques Artimon. Flume s'articule autour de nœuds. Chaque nœud possède :

- une source (d'où les données proviennent)
- un sink (où les données sont envoyées)

La version Made In Arkea patchée par Mathias Herberts permet d'avoir plusieurs nœuds correspondant aux différents fonctionnements propres à Arkea sur une même machine et désactive l'enregistrement auprès de Zookeeper.

Il envoie toutes les minutes les données qu'il a collecté vers les serveurs Kafka.

d) Présentation de Kafka

Kafka est un composant logiciel développé par LinkedIn et passé en incubation chez Apache.

Il réalise l'écriture des messages dans des fichiers de logs, appelés aussi partitions, où chaque ligne contiendra un en-tête assez réduit (essentiellement le checksum) ainsi que le message sous forme

de string ou encodé d'une certaine manière.

Kafka fonctionne en mode distribué. La distribution est en grande partie assurée grâce à Zookeeper. Les différents serveurs Kafka communiquent avec Zookeeper pour signaler la topologie du cluster, que ce soit le nombre de partitions qu'ils gèrent ou leurs tailles, ou l'état de chaque nœud.

Il existe deux types de messages :

- Producteur
- Consommateur

Les messages producteurs sont donc écrits séquentiellement et sont gardés pour une durée déterminée par l'utilisateur.

Les messages consommateurs, eux, visent à récupérer les données pour les stocker dans le cluster Hadoop d'Arkéa ou dans les VHMS.

e) Présentation de Zookeeper



FIGURE 8 – Logo de Zookeeper

ZooKeeper est un logiciel de gestion de configuration pour systèmes distribués, basé sur le logiciel Chubby qui a été développé par Google. ZooKeeper est utilisé entre autres pour l'implémentation de HBase.²

Chaque système s'enregistre auprès de Zookeeper et renvoie son état. Zookeeper permet donc de notifier du bon fonctionnement d'un service à ceux dont ils sont dépendants.

f) Présentation des VHMS

Les VHMS sont des serveurs de type Consommateurs Kafka. Il en existe 4 qui diffèrent suivant leur granularité³ et leur historique.

Ce sont les serveurs que l'on interroge pour récupérer les métriques Artimon en temps réel avec des outils comme Artimongraph (voir Annexe page 22).

2. HBase est une base de données distribuée disposant d'un stockage structuré pour les grandes tables.

3. Leur précision

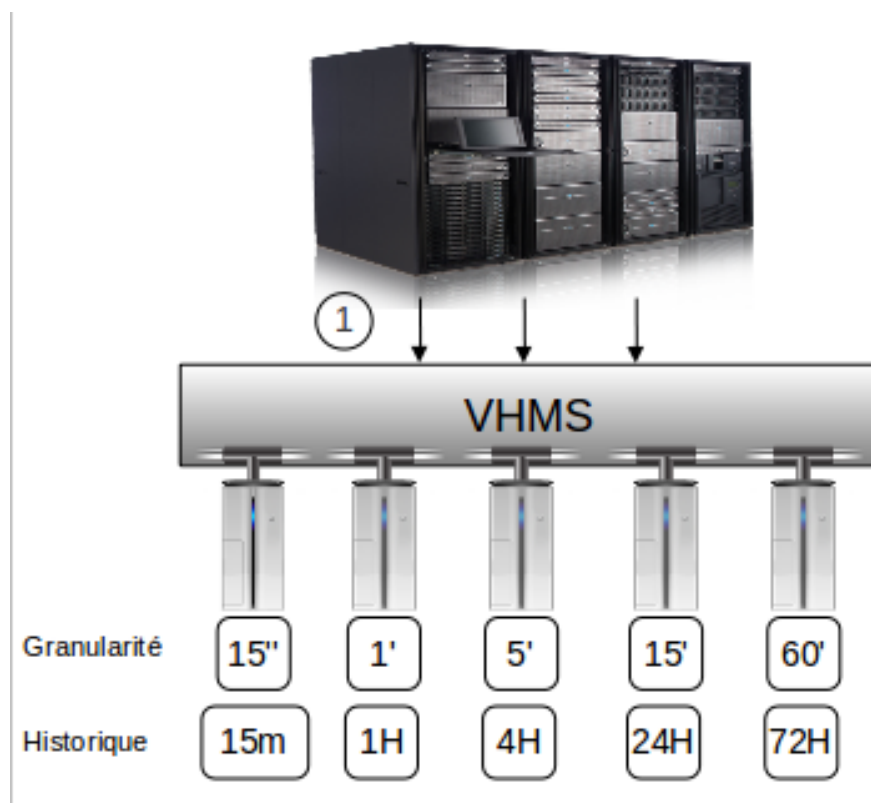


FIGURE 9 – fonctionnement des VHMS

g) Présentation de Hadoop

Hadoop est un projet issu des travaux du projet Google File System (GFS) créé en 2003 par Google. À ses débuts, il n'existait rien sur le marché pour permettre au moteur de recherche de Google de donner un indice de popularité sur l'ensemble des informations collectées.

Ils ont donc créé leur propre solution afin de subvenir à leurs besoins. Le projet fut rapidement mis en open-source et fut repris un peu plus tard par Yahoo en 2006, qui lui ajouta quelques fonctionnalités. C'est en 2007 que le projet Hadoop voit vraiment le jour et Yahoo confia le projet à l'incubateur d'Apache.

En 2012, le projet Hadoop est entièrement sous la tutelle de la fondation Apache, ses sources sont disponibles et libres d'utilisations. La plate-forme Hadoop a été conçue pour résoudre des problèmes lorsque l'on souhaite faire des traitements avec beaucoup de données.

HDFS est un système de fichiers distribués calqué sur le modèle de Google Files System qui gère le stockage, la répartition et l'accès aux données de la plate-forme.

Un cluster HDFS se compose d'un NameNode (serveur maître) qui gère les métadonnées du système de fichiers et de DataNodes qui stockent les données réelles. Quand l'application cliente a besoin d'accéder aux informations, elle interroge le NameNode qui lui indique les DataNodes sur lesquelles se trouvent ces données. L'application cliente va ensuite directement interroger les DataNodes.

Il permet de faire des calculs et des analyses en beaucoup moins de temps qu'un système de données classique. Les données sont réparties et sont répliquées sur plusieurs machines. De plus, Hadoop peut être installé sur n'importe quel matériel de base, impliquant un coût d'utilisation

faible.

La mise en place de cette plate-forme s'inscrit dans une démarche propre au service Systèmes : proposer les solutions les plus performantes possible avec un budget restreint. Le Crédit Mutuel Arkéa a été l'une des premières entreprises françaises à se munir de serveur Linux et d'un cluster Hadoop.

Hadoop permet de stocker de façon permanente sur un an les métriques. Pour l'instant l'infrastructure Artimon génère environ 11TO de métriques par mois.

Je n'ai pas réellement travaillé avec Hadoop, mais c'est une technologie que je trouve très intéressante, et que j'aimerais apprendre à manipuler dans le futur sur un projet personnel peut-être.

h) Présentation de l'infrastructure virtuelle VMware



FIGURE 10 – Logo de VMware

VMware a produit une multitude de solutions en entreprises qui permettent de créer une multitude de VM⁴. Arkea possède plus de 600VM. VMware vCenter Server est une infrastructure virtuelle avec gestion centralisée comme l'explique le schéma ci-dessous :

4. Machine virtuelle

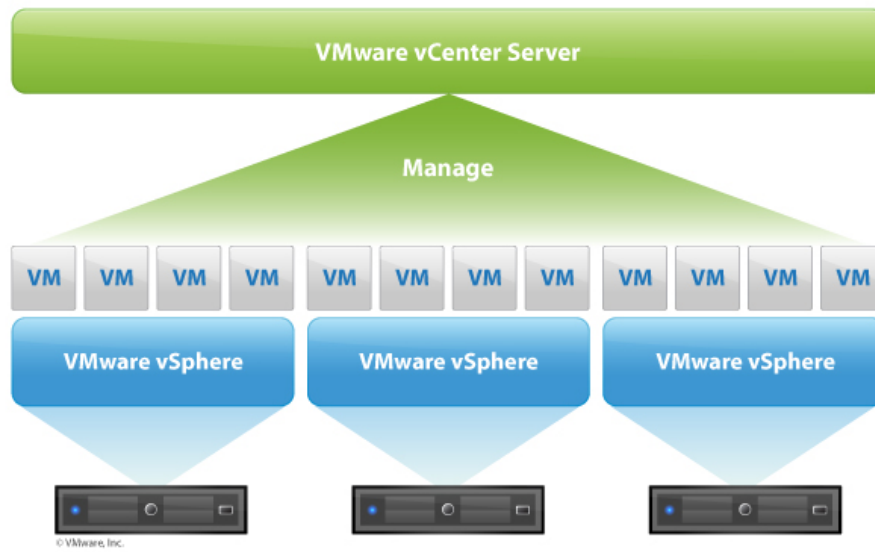


FIGURE 11 – Organisation d'une infrastructure vCenter

Les vSphere, véritables serveurs supportant de nombreuses VM, possèdent déjà tout un système propriétaire de création et de gestion de métriques d'eux-mêmes et de leurs VM dans le vCenter.

Une photo des serveurs Vsphere est disponible en Annexe page 23.

IV) Contexte du stage

a) Contexte

Comme dit précédemment, le groupe possède plus de 2000 serveurs disséminés sur plusieurs sites. L'équipe Système étant basée à Brest, il est nécessaire de pouvoir surveiller à distance la bonne santé de n'importe quel serveur. L'infrastructure Artimon (Arkéa RealTime Monitoring) répond à ce besoin de surveillance.

Malheureusement, cette infrastructure n'était pas compatible avec l'environnement Windows, ce qui représente environ 800 serveurs.

b) Missions

Le sujet du stage était le suivant :

Analyse et la mise en oeuvre de métriques techniques sur des environnements virtualisés VMWare et serveurs Windows.

Le sujet étant assez généraliste, et dans l'idée de découvrir l'infrastructure Artimon, j'ai effectué plusieurs missions durant le stage :

- Portage de Flume sur Windows
- Création de métriques Windows
- Création de métrique venant de l'infrastructure de virtualisation VMWare
L'ensemble de ces missions m'ont permis de découvrir tous les outils que j'utiliserai lors de mon alternance durant l'année scolaire 2013-2014.

La répartition des missions durant mon stage s'est faite de la manière suivante :

- Découverte de l'entreprise et des technologies, mise à niveau en Linux : 1 semaine
- Portage de Flume sur Windows : 4 semaines et demie
- Création de métriques Windows : 1 semaine
- Création de métriques liés à l'infrastructure de virtualisation VMWare : 1/2 semaine

Le reste du temps m'a permis de rédiger le début du présent rapport.

V) Détails des missions

a) Portage de Flume

La première mission de mon stage était le portage de Flume sur Windows et de le faire fonctionner en Service⁵. La version originale de Flume fonctionnait sur Windows et non la version patchée "Made In Arkéa".

La première étape était de réussir à compiler Flume avec Ant en appliquant les différents patches Made in Arkea. Cette tâche a été assez compliquée du fait de plusieurs problèmes techniques (manque de connaissances au début du projet, le réseau qui bloquait les téléchargements des paquets nécessaires à la compilation). Une fois le paquet correctement recompile, il a fallu modifier de nombreuses variables d'environnement dans les fichiers de configuration afin que les chemins soient corrects.

L'étape la plus compliquée était de faire fonctionner à la mode Arkéa Flume. En effet, normalement Flume lance un unique Node sur une machine, donc il a besoin d'un simple paramètre de lancement qui est "start".

A Arkéa, Flume démarre avec plusieurs nœuds actifs, avec chacun leurs paramètres.

Il a donc fallu modifier le code source de la classe Java gérant le lancement des nœuds pour qu'il prennent en compte les différents paramètres des nœuds.

Le fichier de lancement de Flume étant un fichier bash, il a fallu réécrire en entier un fichier de lancement compatible Windows. Arkea possédant une multitude de versions de serveurs, j'ai dû utiliser le Batch, qui est une liste de commande DOS afin qu'il puisse démarrer sur n'importe quelle version de Windows.

Ne connaissant pas le DOS, j'ai mis un peu de temps à obtenir un fichier batch qui fonctionne, surtout que le DOS accuse son âge face à des langages comme le Powershell.

Le fichier Batch permet de passer les bons paramètres de jvm et de nœuds à Procrun, un exécutable Windows d'Apache permettant de créer un Service à partir d'un programme Java.

L'avantage de ProcuRun est qu'il est livré avec un autre logiciel appelé Prunmgr qui permet de gérer le service en temps réel avec une icône dans la barre des tâches.

5. équivalent d'un daemon sur Windows

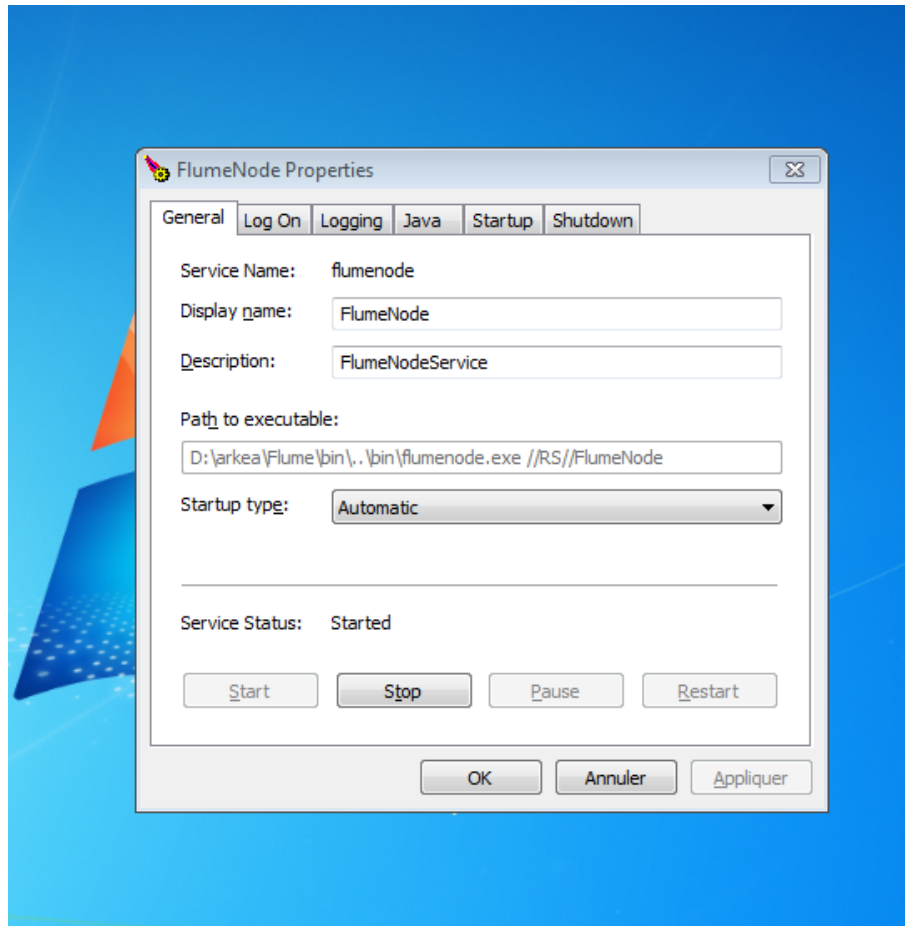


FIGURE 12 – Partie graphique de Prunmgr

Ayant terminé cette mission assez rapidement, j'ai attaqué les soucis de mise en production. En effet, il faut pouvoir l'installer sur des serveurs Windows loin d'être mis à jour, et pouvant avoir une JVM différente de celle que j'utilise.

J'ai donc ajouté à mon package une JVM portable afin de régler les problèmes de variables globales avec la JVM. J'ai également créé une installation silencieuse grâce à Inno Setup⁶ afin qu'il puisse être installé dans le banc générateur de serveurs Windows de l'entreprise.

La création de Flume, et de son installation silencieuse, m'a pris environ 4 semaines et demie sur les 7 de mon stage.

A ce jour, ma version Windows de Flume est prête au déploiement, et sera installée sur tout futur serveur Windows et sur les anciens au fil du temps.

La seule amélioration possible serait d'unifier la version Linux et Windows de Flume ou d'utiliser une version plus récente de Flume.

b) Création de métrique Windows

Après que Flume fonctionnait sur Windows, il fallait l'alimenter avec des métriques système. Le meilleur moyen de récupérer les métriques est d'utiliser ce qu'on appelle des requêtes WMI. Semblable à de la SQL, ces requêtes renvoient la valeur de l'objet demandé.

6. Inno Setup est un logiciel permettant de créer des installateurs Windows

En utilisant toutes les classes WMI, l'ordinateur génère plus de 69 000 valeurs. Pour ne pas exploser l'infrastructure Artimon, nous nous sommes limités à quelques informations vitales comme le processeur, disques dur ou encore les connexions réseaux.

Afin de pouvoir exécuter des requêtes WMI, j'ai choisi le VBScript car c'est un langage de script simple à utiliser avec des requêtes WMI, compatible nativement avec de nombreuses versions de Windows. De plus, j'avais manipulé du VBScript l'année dernière lors de mon stage à Thomson. Pour pouvoir appeler le script indéfiniment, j'ai ajouté à Flume un thread Java qui exécute mon script de manière cyclique. Grâce aux VBScript, je récupère le résultat des requêtes WMI qui m'intéresse, puis je formate les variables pour qu'elles correspondent au format Artimon, puis le script génère un fichier dans le dossier que Flume surveille pour qu'il remonte les données.

Il m'a fallu environ 1 semaine à écrire le script, car j'ai eu des difficultés à trouver les requêtes qui étaient pertinentes parmi les 69 000 variables.

Des exemples de variables sont disponibles en Annexe page 24.

c) Créations de métriques sur plateforme virtualisée VMWare

Comme dit précédemment, les vSpheres possèdent déjà un système de création et de gestion de métriques. L'objectif de la mission était de récupérer les données en temps réel pour les intégrer à Flume.

Pour cela, VMware a créé une API pour Powershell appelé PowerCLI, qui permet de dialoguer avec les VSphere pour administrer le tout.

Ne connaissant ni le Powershell ni l'API, j'ai appris ce langage même si c'est très proche d'un bash Linux. L'API est très simple à utiliser, tout est géré sous la forme d'objet, il suffit de récupérer les attributs des bons objets, de les mettre au format Artimon et de boucler le tout pour chaque VM présent sur la VSphere, le script s'exécutant en tâche planifiée sur une VM dédiée.

J'ai mis 3 jours à écrire le script, sachant que je devais découvrir PowerShell.

Des exemples de métriques pour les VSphere sont disponibles en Annexe page 25 et page 26 pour les métriques de VM.

VI) Bilan

Ce stage a permis d'améliorer la qualité de l'infrastructure Windows en unifiant la partie Monitoring aux 2 systèmes d'exploitations principaux qui sont Linux et Windows. Le portage de Flume sur Windows permet de pouvoir envoyer toute sorte de données vers les serveurs de centralisation de données.

A partir de mon travail et de l'infrastructure existante, on peut imaginer la création d'un système de détection automatique d'erreurs ou de problèmes qui fonctionneraient directement sur l'infrastructure Artimon.

Ma version de Flume est déjà en service. En effet, elle se trouve sur les Banc de génération de serveurs. Ainsi, tous les futurs serveurs d'Arkéa contiendront Flume et seront donc monitorés de la même manière que sur Linux. L'installation sur les serveurs actuels sera faite au fur et à mesure.

Je suis très content de mon stage pour de nombreuses raisons. Tout d'abord j'ai rejoint une équipe dynamique et extrêmement sympathique. Malgré le fait que j'étais tout seul sur le projet, j'ai dû beaucoup communiquer avec les personnes qui s'occupaient de moi, car j'avais de nombreuses questions sur les technologies. J'ai également beaucoup aimé le fait de pouvoir travailler en autonomie sur mon projet. D'une manière générale c'est moi qui devait trouver la technologie la plus adaptée à la situation, et je la présentais à mes tuteurs.

J'ai été confronté à quelques difficultés, notamment au niveau des langages, la découverte et l'apprentissage de nouveaux langages au fur et à mesure était nécessaire. Curieux de nature, c'est cette partie là de mon stage que j'ai le plus apprécié, puisque j'ai beaucoup appris sur le plan technique en java et j'ai découvert le Bash, le batch, le VBScript et plus généralement de nombreuses choses sur Windows et Linux...

Ce stage confirme mon projet professionnel de devenir ingénieur développeur. je suis impatient de pouvoir retourner dans cette équipe lors de mon contrat de professionnalisation durant l'année 2013/2014, et de continuer à découvrir de nouvelles technologies.

VII) Annexe

a) Schéma de fonctionnement Artimon

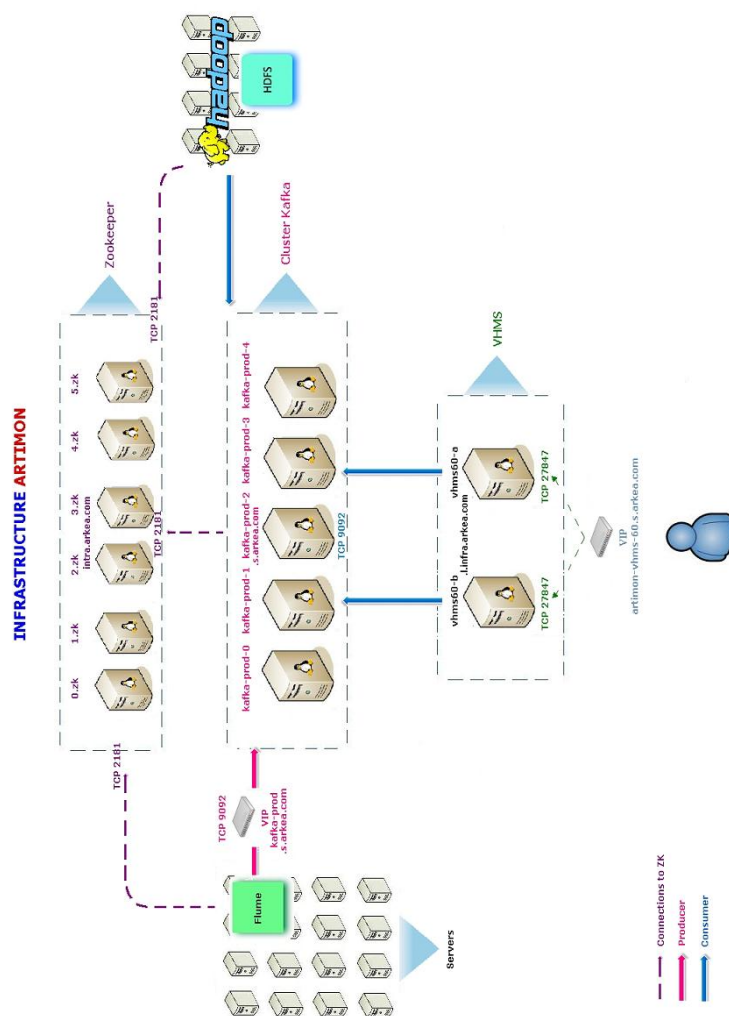


FIGURE 13 – Schéma des communications des briques de Artimon

b) Artimongraph

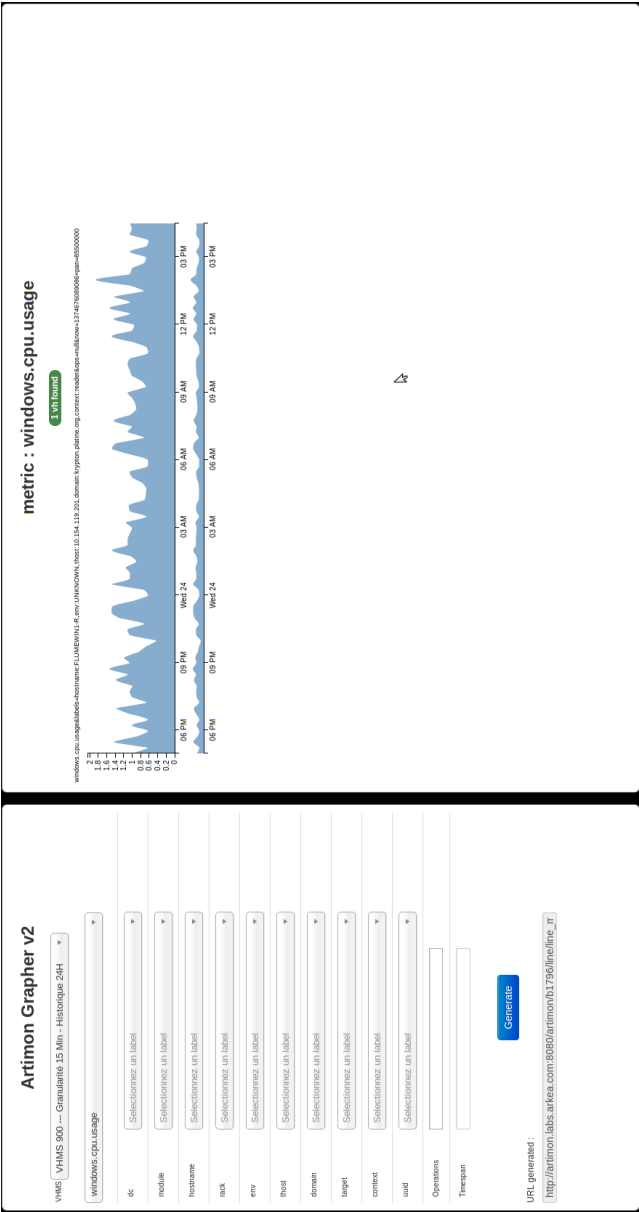


FIGURE 14 – Copie d’écran d’artimongraph montrant une métrique cpu

c) Photo de rack



FIGURE 15 – Photo de rack de serveurs de type VSphere

d) Exemple de variable Artimon Windows

```
windows.ram.totalphysicalmemory{hostname=0000094336-WIN, domain=cmb.platine.org} 3220692992
windows.cpu.physicalcore{hostname=0000094336-WIN, domain=cmb.platine.org} 2
windows.cpu.logicalcore{hostname=0000094336-WIN, domain=cmb.platine.org} 2
windows.cpu.maxclockspeed{hostname=0000094336-WIN, domain=cmb.platine.org} 3292
windows.cpu.currentclockspeed{hostname=0000094336-WIN, domain=cmb.platine.org} 3292
windows.cpu.usage{hostname=0000094336-WIN, domain=cmb.platine.org} 58.04
windows.hdd.freespace{hostname=0000094336-WIN, domain=cmb.platine.org, Drive=C} 29765
windows.hdd.size{hostname=0000094336-WIN, domain=cmb.platine.org, Drive=C } 65533
windows.hdd.freespace{hostname=0000094336-WIN, domain=cmb.platine.org, Drive=D} 11246
windows.hdd.size{hostname=0000094336-WIN, domain=cmb.platine.org, Drive=D } 12000
windows.hdd.percentdiskreadtime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdisktime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdiskwritetime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdiskreadtime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdisktime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdiskwritetime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdiskreadtime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdisktime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.percentdiskwritetime{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.hdd.diskvarbytespersec{hostname=0000094336-WIN, domain=cmb.platine.org, drive=-C-D} 197040128
windows.hdd.diskreadpersec{hostname=0000094336-WIN, domain=cmb.platine.org, drive=-C-D} 11495
windows.hdd.diskWritespersec{hostname=0000094336-WIN, domain=cmb.platine.org, drive=-C-D} 4574
windows.hdd.diskvarbytespersec{hostname=0000094336-WIN, domain=cmb.platine.org, drive=Total} 197040128
windows.hdd.diskreadpersec{hostname=0000094336-WIN, domain=cmb.platine.org, drive=Total} 11495
windows.hdd.diskWritespersec{hostname=0000094336-WIN, domain=cmb.platine.org, drive=Total} 4574
windows.tcp.connectionfailures{hostname=0000094336-WIN, domain=cmb.platine.org} 4
windows.tcp.connectionsactive{hostname=0000094336-WIN, domain=cmb.platine.org} 106
windows.tcp.connectionsestablished{hostname=0000094336-WIN, domain=cmb.platine.org} 12
windows.tcp.connectionspassive{hostname=0000094336-WIN, domain=cmb.platine.org} 16
windows.tcp.connectionsreset{hostname=0000094336-WIN, domain=cmb.platine.org} 9
windows.network.packetsPersec{hostname=0000094336-WIN, domain=cmb.platine.org} 4491
windows.network.bytestotalpersec{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.network.utilization{hostname=0000094336-WIN, domain=cmb.platine.org} 0
windows.network.bytestotalpersec{hostname=0000094336-WIN, domain=cmb.platine.org} 0
```


e) Exemple de variables Artimon issues de VM

```
vmware.vm.cpu.average{name=A06034-00} 1.89
vmware.vm.mem.vmmemctl.average{name=A06034-00} 0
vmware.vm.mem.usage.average{name=A06034-00} 18.99
vmware.vm.mem.active.average{name=A06034-00} 99612
vmware.vm.mem.granted.average{name=A06034-00} 524168
vmware.vm.net.transmitted.average{name=A06034-00,connection=4000} 0
vmware.vm.net.received.average{name=A06034-00,connection=4000} 0
vmware.vm.cpu.average{name=A06034-06-rec} 7.25
vmware.vm.mem.vmmemctl.average{name=A06034-06-rec} 0
vmware.vm.mem.usage.average{name=A06034-06-rec} 5.99
vmware.vm.mem.active.average{name=A06034-06-rec} 125828
vmware.vm.mem.granted.average{name=A06034-06-rec} 2097152
vmware.vm.net.transmitted.average{name=A06034-06-rec,connection=4000} 0
vmware.vm.net.received.average{name=A06034-06-rec,connection=4000} 0
vmware.vm.cpu.average{name=A06034-07-rec} 3.67
vmware.vm.mem.vmmemctl.average{name=A06034-07-rec} 0
vmware.vm.mem.usage.average{name=A06034-07-rec} 5.99
vmware.vm.mem.active.average{name=A06034-07-rec} 125828
vmware.vm.mem.granted.average{name=A06034-07-rec} 2097152
vmware.vm.net.transmitted.average{name=A06034-07-rec,connection=4000} 0
vmware.vm.net.received.average{name=A06034-07-rec,connection=4000} 0
vmware.vm.cpu.average{name=A06034-08-rec} 4.03
vmware.vm.mem.vmmemctl.average{name=A06034-08-rec} 0
vmware.vm.mem.usage.average{name=A06034-08-rec} 5.99
vmware.vm.mem.active.average{name=A06034-08-rec} 125828
vmware.vm.mem.granted.average{name=A06034-08-rec} 2097152
vmware.vm.net.transmitted.average{name=A06034-08-rec,connection=4000} 0
vmware.vm.net.received.average{name=A06034-08-rec,connection=4000} 0
```

f) Exemple de variables Artimon issues d'host

```
vmware.host.cpu.average{name=10.154.102.229} 78
vmware.host.cpu.average{name=10.154.102.229,core=0} 0.58
vmware.host.cpu.average{name=10.154.102.229,core=1} 0.08
vmware.host.cpu.average{name=10.154.102.229,core=2} 0.53
vmware.host.cpu.average{name=10.154.102.229,core=3} 0.39
vmware.host.cpu.average{name=10.154.102.229,core=4} 0.12
vmware.host.cpu.average{name=10.154.102.229,core=5} 0.1
vmware.host.cpu.average{name=10.154.102.229,core=6} 0.1
vmware.host.cpu.average{name=10.154.102.229,core=7} 0.06
vmware.host.cpu.average{name=10.154.102.229,core=8} 0.26
vmware.host.cpu.average{name=10.154.102.229,core=9} 0.33
vmware.host.cpu.average{name=10.154.102.229} 0.43
vmware.host.cpu.average{name=10.154.102.229,core=10} 0.43
vmware.host.cpu.average{name=10.154.102.229,core=11} 0.08
vmware.host.cpu.average{name=10.154.102.229,core=12} 0.05
vmware.host.cpu.average{name=10.154.102.229,core=13} 0.09
vmware.host.cpu.average{name=10.154.102.229,core=14} 0.13
vmware.host.cpu.average{name=10.154.102.229,core=15} 0.08
vmware.host.mem.active.average{name=10.154.102.229} 97572
vmware.host.mem.usage.average{name=10.154.102.229} 3.03
vmware.host.mem.vmmemctl.average{name=10.154.102.229} 0
vmware.host.mem.granted.average{name=10.154.102.229} 97572
vmware.host.net.received.average{name=10.154.102.229,connection=vmnic0} 0
vmware.host.net.received.average{name=10.154.102.229,connection=vmnic1} 11
vmware.host.net.received.average{name=10.154.102.229,connection=vmnic2} 0
vmware.host.net.received.average{name=10.154.102.229,connection=vmnic3} 0
vmware.host.net.transmitted.average{name=10.154.102.229,connection=vmnic0} 0
vmware.host.net.transmitted.average{name=10.154.102.229,connection=vmnic1} 0
vmware.host.net.transmitted.average{name=10.154.102.229,connection=vmnic2} 0
vmware.host.net.transmitted.average{name=10.154.102.229,connection=vmnic3} 0
```