

---

consigne: <http://prodageo.insa-rouen.fr/casimono/sujetprojmd/consignes.html>

---

# Monographie - Exploitation des journaux (logs) d'application

---

## Tableau de suivi des versions

---

Numéro de version	Date
0.00	30/10/2020
0.01	21/11/2020

## Cartouche 2

---

- Code du sujet : 40Z
- Code de l'alliance : 2020A40Z
- Liste des équipes
  - E16 - CASIment l'UML
    - Larcher, théo (aerodynamic-saucepan)
    - Billard, baptiste (BaptBillard)
    - Lebeaud, simon (LeboSi)
  - E04 - Equipe B
    - Lacroix, Daphné (dalacroix)
    - Lardière, Nina (nlardiere)
    - Delaunay, Julien (jdelahunay)
  - E07 - Equipe C
    - Benkassou, Ali (khazit)
    - Leclerc, Tanguy (LeclercTanguy)
    - Marie, Joran (Joran MARIE)
  - E08 - Equipe D
    - De Laroque, Florent (FlorentDL)
    - Marie, Leo (Imarie01)
    - Toutain, Aurélien

## Partie A

---

### A0 : Introduction.

Le stockage, la collecte et l'analyse de grandes quantités de journaux (logs) sont une partie indispensable dans la gestion d'une architecture logicielle complexe. L'objectif de ce sujet est de trouver des techniques permettant d'analyser efficacement des journaux (logs) d'une application destinée à monter en charge.

Ce projet permettra d'apporter des éléments de solution à un besoin réel et actuel d'entreprises devant manipuler des bases de données volumineuses (Big Data) et marchera de paire avec une application, choisie par l'équipe, qui permettra de générer les données (les logs).

Afin d'étudier les différentes solutions permettant d'exploiter efficacement la richesse des informations contenues dans les logs

l'application choisie par l'équipe sera stressée avec un outil comme Gatling.

Ce projet n'a donc pas pour but d'apporter une amélioration à un processus de traitement de données mais bien à étudier différents outils d'analyse.

L'application choisie par l'équipe est : 60B : annuaire contacts ASI .

Les questions d'amorces suivantes permettent de mieux appréhender le sujet de ce document.

## Quel est le rôle respectif de chacun des 3 outils ELK ?

ELK est un outil d'analyse de logs composé de 3 logiciels open source. Elasticsearch, Logstash et Kibana. Les rôles respectifs des trois outils sont détaillés ci-dessous.

- Elasticsearch permet l'indexation et le stockage des informations sur une base de données NoSQL. Il propose également des requêtes avancées pour effectuer une analyse détaillée et stocke toutes les données de manière centralisée.
  - Logstash est un outil d'intégration de données open source qui permet de collecter des données à partir d'une variété de sources, de les filtrer, les transformer et de les envoyer à la destination souhaitée (ex: Elasticsearch). Son but principal est de rassembler et normaliser tous les types de données provenant de différentes sources et de les rendre disponibles pour une utilisation ultérieure.
- Kibana est un outil de visualisation de données qui complète la pile ELK , c'est une couche de visualisation qui fonctionne au-dessus d'Elasticsearch, offrant aux utilisateurs la possibilité d'analyser et de visualiser les données récupérées Elasticsearch.

## Qu'est-ce que le mouvement DevOps ? En quoi ELK contribue au mouvement DevOps ?

DevOps est la contraction des mots "Developpers" et "Ops". La culture DevOps vise à casser la séparation entre les développeurs qui produisent de nouvelles fonctionnalités, et les Ops qui ont pour objectif la fiabilité du site. La caractéristique la plus importante de DevOps est la création d'un cycle de production et de lancement soutenu par l'automatisation. À chaque étape, des processus scénarisés vérifient s'il y a des erreurs.

C'est dans cette optique qu'ELK contribue au mouvement DevOps, en effet en tant que solution d'analyse de logs performante elle permet de mettre en place des systèmes de monitoring et d'alerting. Le monitoring et l'alerting permettent d'avoir une vue d'ensemble sur son infrastructure, de résoudre les problèmes qui surviennent et d'améliorer les performances.

## A quel type de question sur l'application stressée peut permettre de répondre une suite telle que ELK ?

Dans un contexte de stress d'application, ELK permet de mesurer les performances de l'application et détecter les éventuelles failles lors de scénarios particuliers, à l'aide de syst\_mes de monitoring et d'alerting.

## Quelles solutions alternatives à ELK existe-t-il sur le marché (open-source, saas ou cots) ?

Apache Solr constitue avec ELK les 2 principaux moteurs de recherche open source. Mais Elasticsearch est mieux adapté et plus fréquemment utilisé dans le cas d'utilisation d'analyse de journaux. Plusieurs solutions SaaS existent comme Algolia, Searchify ou Oracle Log Analytics.

## A quoi sert rsyslogd ? Qu'est-ce qu'un fichier RRD ?

## Quelle est la différence entre une base de données et un moteur d'indexation ?

## A1. Glossaire et Mots-clés.

- **Big Data** : Domaine technologique dédié à l'analyse de très grands volumes de données informatiques (petaoctets), issus d'une grande variété de sources, tels les moteurs de recherche et les réseaux sociaux ; ces grands volumes de données. (Source Larousse)

- **Base de données** : Ensemble structuré de fichiers regroupant des informations ayant certains caractères en commun ; logiciel permettant de constituer et de gérer ces fichiers. (Source Larousse)
- **Logs** : Fichier regroupant l'ensemble des événements survenus sur un logiciel, une application, un serveur ou tout autre système informatique. (Source dicodunet)
- **Analyse** : Étude minutieuse, précise faite pour dégager les éléments qui constituent un ensemble, pour l'expliquer, l'éclairer. (Source Larousse)
- **ELK (Elasticsearch, Logstash, Kibana)** : ELK est un outil d'analyse de logs composé de 3 logiciels open source, développés par la société Elastic : Elasticsearch, Logstash et Kibana. (Source Oracle)
- **Rsyslog** : Logiciel libre utilisé sur des systèmes d'exploitation de type Unix transférant les messages des journaux d'événements sur un réseau IP. (Source Wikipedia)
- **Fichiers RRD** : Solution de stockage des structures pyramidales pour un jeu de données raster. (Source ArcGIS)
- **Gatling** : Outil open-source de test de charge et de performance pour applications web. (Source Wikipedia)
- **Stress** : Il s'agit d'un test au cours duquel on va simuler l'activité maximale attendue tous scénarios fonctionnels confondus en heure de pointe de l'application, pour voir comment le système réagit au maximum de l'activité attendue des utilisateurs. (Source Wikipedia)
- **DevOps** : Mouvement en ingénierie informatique et une pratique technique visant à l'unification du développement logiciel (dev) et de l'administration des infrastructures informatiques (ops), notamment l'administration système. (Source Wikipedia)

## A2. Webographie

- [Wikipédia](#)
- [Simplilearn](#)
- [Datamation](#)
- [Sodifrance \(Gatling\)](#)
- [ElasticSearch](#)

## A3. Bibliographie

Anurag Srivastava et Douglas Miller, 24 octobre 2019, **Elasticsearch 7 Quick Start Guide: Get up and running with the distributed search and analytics capabilities of Elasticsearch**

Sections :

- Use cases of ElastisStack
- Anatomy of an Analyzer
- Aggregating datasets

*Critère de choix* : Guide exhaustif sur Elasticsearch.

James Turnbull, 2014, **The Logstash Book**, Kindle edition

Sections :

- Add structured logging to your applications and parse your application logs
- Use Logstash's awesome web interface Kibana
- Install and deploy Logstash.

*Critère de choix* : Guide exhaustif sur Logstash.

Saurabh Chhajed, 2015, **Learning ELK Stack**, eBook

Sections :

- Introduction to ELK Stack
- Building Your First Data Pipeline with ELK
- Collect
- Parse and transform Data with Logstash

*Critère de choix* : Présentation de ELK (Elasticsearch, Logstash, and Kibana), présentation des méthodes et de la mise en place à la configuration en passant par l'installation des outils. Parle également de pipeline de données.

## A4. Acteurs

### Apache Solr

Apache Software Foundation est une organisation à but non lucratif créée en juin 1999 dans le Delaware aux États-Unis par Brian Behlendorf, Roy Fielding et Ben Laurie. Elle développe ses logiciels en open-source, en mode collaboratif.

Sa plateforme logicielle de moteur de recherche Solr (prononcée "Solar"), basée sur le moteur d'indexation et de recherche de texte **Lucene**.

Le développement de Solr débute chez CNET, mais ces derniers décident très vite de rendre le code public : en janvier 2006, la première version de la plateforme est publiée, et son développement est confié à Apache. Ces derniers améliorent alors le moteur, notamment en 2008 : ils améliorent largement ses performances et le dotent d'une recherche distribuée.

Les projets **Lucene** et **Solr** sont fusionnés en 2011 : le développement est commun, les distributions sont séparées mais un tronc commun permettant de ne disposer que d'une seule version est vite créé. La recherche distribuée est nettement améliorée ; **Solr** est aujourd'hui utilisée notamment par **Adobe, Instagram, Netflix, eBay et Goldman Sachs**.

**Solr** permet l'intégration des bases de données, le surlignage des termes trouvés, l'auto-suggestion et la recherche full-text.

### ElasticSearch

Elasticsearch inc. est une entreprise centrée sur la recherche fondée par Steven Schuurman, Uri Boness et Simon Willnauer.

L'entreprise est basée sur le succès de l'outil **Elasticsearch**, un logiciel open source, fonctionnant sur Lucene pour l'indexation et la recherche de données. Suite au succès d'**Elasticsearch** deux autres projets sont créés ; Logstash et Kibana. Logstash est un outil informatique de collecte, analyse et stockage de logs, c'est un outil d'ingestion open source. Il est généralement associé avec **Elasticsearch**, moteur de recherche distribué, et Kibana, interface d'Elasticsearch. En effet Kibana est une interface utilisateur permettant de visualiser les logs. Ces trois outils ; **Elasticsearch**, Kibana et Logstash forment la solution EKL.

L'entreprise met toute la puissance de la recherche au service de ses clients pour les aider à explorer et analyser leurs données, que ce soit le cours des actions, les flux twitter ou les logs Apache.

### Algolia

Algolia est une start up française. Lors de sa création en 2012 par Nicolas Dessaigne et Julien Lemoine, ils ont commencé à travailler sur la recherche hors ligne pour les téléphones mobiles.

Aujourd'hui Algolia propose une plateforme de recherche en tant que serveur « search as a service » qui est comme un moteur de recherche hébergé en externe pour faire des recherches sur site avec des résultats rapides et concrets.

**Google** ou **Microsoft** proposent déjà cela, mais ils explorent et indexent le web de manière générale. Algolia propose, quant à lui, de n'indexer que les sites de leurs clients ce qui rend les recherches plus simples. Ils utilisent **RESTful JSON API** pour que les données du client soient push vers Algolia. La zone de recherche est ensuite ajoutée aux pages web du client.

Comme la recherche se fait sur un site précis et non sur le web, Algolia peut proposer des recherches intelligentes et plus spécifiques au site car ils en connaissent la structure et la sémantique.

Ils sont également très attentifs sur leur capacité à fournir des fonctionnalités instantanées, multi-plateformes et tolérantes aux fautes de frappe.

Leur solution n'est toutefois pas open-source.

Algolia fournit son service via diverses API. Il prend en charge plusieurs langages tels que Python, Ruby, PHP, JavaScript, Java, Go, C #, Scala et les systèmes d'exploitation Android et iOS. Leur solution est également intégrable sur Ruby on Rails, Symfony, Django et Laravel.

## A5. Facteurs qualité

- Consistency
- Accuracy
- Error Tolerance

## A6. Indicateurs qualité

- Consistency
  - Task effectiveness
  - Task Time
- Accuracy
  - Number of failures
  - Number of faults
- Error tolerance
  - Number of memory related errors
  - Number of inaccurate computations encountered by users

## A7. Références théoriques

Nous allons dans cette partie faire le point sur les patterns appliqués au DevOps issus de <http://blog.devopsguys.com/2030/02/20/twelve-devops-anti-patterns/> et <https://www.slideshare.net/realgenekim/2012-velocity-london-devops-patterns-distilled> ainsi que du management des logs en DevOps selon <https://medium.com/logsense/devops-and-log-management-in-2019-b05d9dd08b65>.

Ces patterns indiquent la marche à suivre quand on réalise du DevOps.

### Les 3 principes du DevOps

#### Systems thinking (Dev to Ops)

Le premier principe consiste à comprendre le flux de travail et à toujours chercher à l'augmenter. Il ne faut jamais autoriser des optimisations locales qui peuvent causer des dégradations globales et il faut parvenir à une compréhension approfondie du système.

#### Amplify feedback loops (Ops to Dev)

Ce principe consiste à comprendre et répondre aux besoins des clients internes et externes. Pour cela il faut de nombreux feedbacks qui doivent être conscits. Il faudra créer et intégrer les connaissances où elles sont nécessaires.

#### Culture of continual experimentation and learning

Le troisième principe consiste à appliquer les deux premiers en continu, afin de créer un environnement qui permet de l'apprentissage et de la compréhension en continu.

### DevOps Areas

Nous allons maintenant voir les 4 thématiques du DevOps.

#### Area 1: Extend delivery to production

Cette thématique a pour but principal de se concentrer sur l'aspect "entreprise" avec une vue de bout en bout du projet. Cela permet d'engager la conversation entre les différents acteurs pour avancer.

Les différentes étapes sont:

- Établir la confiance entre les équipes avec des meetings.
- Comprendre le goulot d'étranglement, la limite du processus.
- L'intégration continue.

- La livraison continue.
- Intégrer les autres rôles dans le processus (management, assurance qualité ...).

## **Area 2: Extend operations feedback to project**

Cette thématique a pour but de produire un feedback et de donner de la visibilité sur le travail réalisé. Cela afin de montrer les améliorations de l'organisation.

Pour cela il faut:

- Des objectifs et des instructions de fonctionnement clairs.
- Une prise de conscience commune de la situation.

## **Area 3: Embed Project knowledge into Operations**

Cette thématique a pour but de raccourcir et d'amplifier les feedback, de créer de la connaissance et des capacités où il y en a besoin et s'assurer de l'optimisation du système.

## **Area 4: Embed Operations knowledge into Project**

Cette thématique a pour but d'apprendre aux développeurs à réfléchir comme les opérateurs, de gérer la dette technique et d'avoir une vision de bout-en-bout du projet.

## **Lien entre DevOps et l'analyse des logs**

La mise en pratique du DevOps vise à détecter rapidement les problèmes, afin que le code publié soit fiable. Le monitoring et l'analyse des logs sont importants pour obtenir une publication fiable du code. Mais l'analyse des logs est rendu de plus en plus compliqué avec la multiplication des instances et des conteneurs. On peut se retrouver avec des données brutes de plusieurs gigaoctets, ce qui rend impossible l'intervention humaine pour trouver les informations.

Pour pallier à ce problème, il faut donc rendre l'analyse des logs automatiques, hors l'automatisation est au coeur du DevOps. Il faut donc élargir l'automatisation à l'analyse des logs pour permettre un travail approfondi de détection des problèmes. Cela peut se faire au moyen de scripts automatisés qui vont identifier plusieurs types de problèmes.

Les non-fatal errors, qui ne causent pas de problèmes évidents pour l'utilisateur comme des données non valides ou des exceptions qui sont capturées et traitées. Ces problèmes peuvent dégrader les performances ou encore consommer de la mémoire. L'analyse des logs permet d'identifier la cause de ces problèmes en indiquant ce qui se passe avant le message d'erreur ou le crash du système.

Une nouvelle release pourrait affecter les performances du système. L'analyse des logs permet de savoir où le temps nécessaire à l'exécution d'une tâche ou les ressources requises ont changé. De plus le changement dans l'usage de certains patterns peut mettre à rude épreuve des aspects du code qui n'avaient pas beaucoup d'impact auparavant. Encore une fois ces problèmes pourraient être identifiés avec une analyse automatique des logs. Faire une analyse statistique de ces analyses de logs peuvent permettre d'identifier certaines tendances et donc d'accélérer la production en évitant de futurs problèmes.

# **Partie B**

---

## **B1. Approche technique**

La tendance actuelle quant à l'observabilité des donnée nous amène à proceder comme suit : on collecte nos indicateurs dans une base de données temporelle ou un service SaaS pour monitorer les ressources, ensuite on collecte les logs (avec la suite ELK), et enfin on utilise un troisième outil pour effectuer les traçage au niveau des requêtes.

Cependant cette approche a plusieurs inconvénients: la mise en corrélation manuelle de différentes sources de données et la redondance des données font perdre beaucoup de temps et les frais engendrés par la gestion de 3 datastores opérationnels sont élevés.

Il est donc important de pouvoir disposer de toutes ses informations dans un magasin opérationnel unique, permettant de faire corréler ses données automatiquement via une interface utilisateur intuitive.

Nous nous basons sur le fonctionnement de la suite ELK qui permet d'exploiter toutes les données.

La suite ELK est aujourd'hui composée de 3 logiciels:

- Elasticsearch,
- Kibana,
- Logstash

Commençons par la collecte des données avec Logstash. Logstash est considéré comme un ETL(Extract-Transform-Load), il s'agit d'un intergiciel(middleware) permettant de faire de la synchronisation massive entre plusieurs sources de données. Il est capable de gérer quasiment tous les types de logs et permet donc de filter, normaliser les informations liées au logs issus de différentes sources.

Une fois que ce pre-process de la donnée a été effectué, ces informations sont envoyées au système de stockage, Elasticsearch, qui lui a pour but l'indexation des données et les recherches sur celles-ci.

C'est enfin Kibana qui va récupérer les données issues d'Elasticsearch afin de fournir un système de visualisation graphique des données pour l'utilisateur.



## Logstash

Comme expliqué précédemment, Logstash est une pipeline côté serveur permettant le traitement des données. Cet outil permet de récolter, analyser, filtrer et sauvegarder des logs pour les transformer en documents formatés pouvant être utilisés ensuite par Elasticsearch. Il accepte quasiment tous les types de logs tels que les logs système, les logs de serveur web, les logs d'erreurs et les logs d'applications. L'objectif est d'obtenir un format uniformisé pour booster l'analyse.

### Configuration flexible de pipeline

Les possibilités d'entrées différentes sont immenses (54 plugins en entrée sont listés sur la documentation). Les logs émanant de différentes sources sont traités simultanément. Voici maintenant plus de détails sur les transformations de différents formats de données.

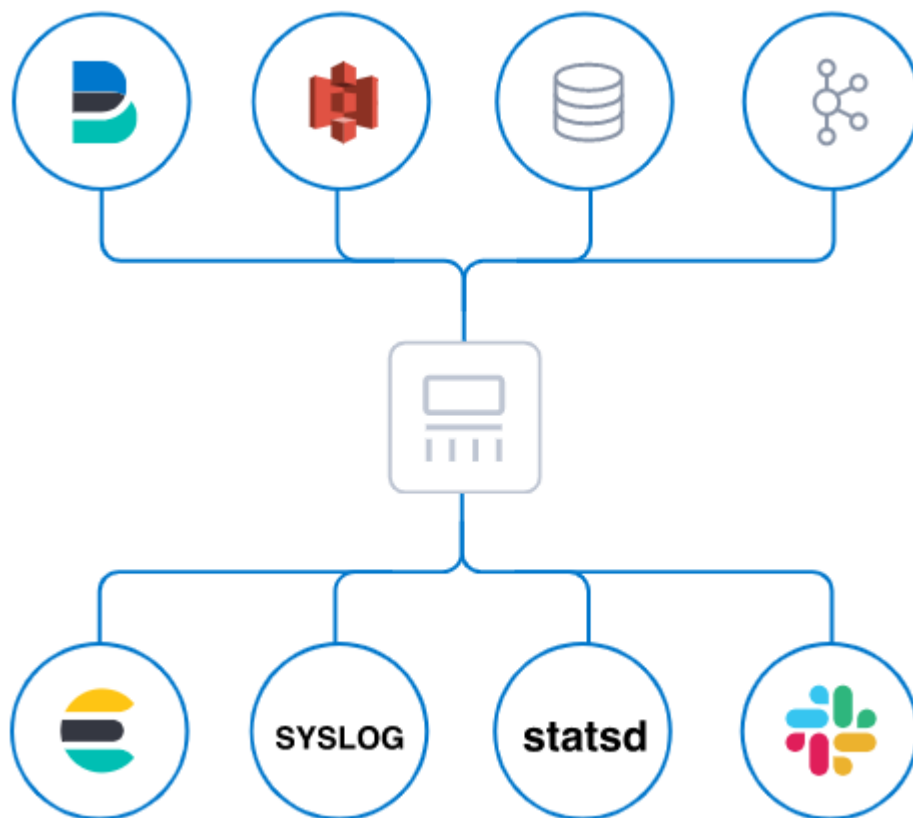
Logstash utilise le processeur Grok pour transformer des données non-structurées en données structurées. Grok extrait des champs structurés d'un seul champ de texte dans un document: on choisit le champ dont on veut extraire les champs correspondants, ainsi que le grok pattern qui devrait correspondre. Un grok pattern est comme une expression régulière qui supporte des expressions aliasées qui peuvent être réutilisées.

Cela ne s'arrête pas là: Logstash permet également de déchiffrer des coordonnées géographiques à partir d'adresses IP ou encore d'anonymiser des données personnelles et d'exclure entièrement des champs confidentiels. Logstash compte plusieurs

dizaines de plugins de filtre permettant de s'adapter au type d'entrée.

Après le traitement des données, bien que le plus utilisé soit Elasticsearch, Logstash offre un large choix de protocoles de sortie tels que csv, email, mongodb, tcp et des dizaines d'autres.

La force de Logstash repose donc dans ses possibilités d'adaptation et de personnalisation à tout type d'architecture. En effet, en plus des dizaines de plugins disponibles, il est possible de créer soi-même un plugin Logstash assez facilement grâce à leur API afin de rendre la configuration de sa pipeline encore plus flexible.



### Facteurs qualité

Logstash utilise des files d'attente persistantes pour garantir la livraison d'événements sans pertes. Les événements non traités sont envoyés vers une file d'attente de lettres mortes pour investigation et relecture.

Le service proposé par Logstash permet d'assurer la sécurité totale des pipelines ainsi que de pouvoir avoir un aperçu détaillé des fonctions et des flux de données.

### ElasticSearch

ElasticSearch est un moteur de recherche basé sur la bibliothèque Apache Lucene qui est gratuite et open-source.

Elasticsearch peut être utilisé pour rechercher toutes sortes de documents. Cela permet une recherche évolutive, en temps quasi réel.

Elasticsearch est distribué, ce qui signifie que les indices peuvent être divisés en shards (partition horizontale des données dans une base de données ou un moteur de recherche) pouvant avoir zéro ou plusieurs répliques. Chaque nœud héberge une ou plusieurs shards et joue le rôle de coordinateur pour déléguer les opérations aux shards correctes. Le rééquilibrage et le routage se font automatiquement.

Les données connexes sont souvent stockées dans le même index, qui se compose d'une ou plusieurs shards primaires et de zéro ou plusieurs shards répliquées. Une fois qu'un index a été créé, le nombre de shards primaires ne peut plus être modifié.



Elasticsearch utilise Lucene et tente de rendre toutes ses fonctionnalités disponibles par le biais du JSON et de l'API Java. Il prend en charge le facettage (application de filtres) et la recherche inverse (retourne les recherches qui correspondent à un document), ce qui peut être utile pour signaler si de nouveaux documents correspondent à des requêtes enregistrées.

Une autre fonctionnalité appelée "gateway" gère la persistance à long terme des index; par exemple, un index peut être récupéré à partir de la gateway en cas de panne du serveur. Elasticsearch prend en charge les requêtes GET en temps réel, ce qui le rend adapté à un stockage de données NoSQL. Malheureusement les transactions distribuées manquent à Elasticsearch.

Elasticsearch est développé en parallèle avec un moteur de collecte et d'analyse de données appelé Logstash, une plateforme d'analyse et de visualisation appelée Kibana, et Beats, un ensemble de transporteurs de données légers. Les quatre produits sont conçus pour être utilisés comme une solution intégrée, appelée "Elastic Stack" (anciennement "ELK stack").

- **Scalabilité**

L'utilisation de JSON permet une scalabilité horizontale simplifiée, ce qui permet d'adapter le hardware très facilement et ainsi traiter un nombre impressionnant de requêtes par seconde.

- **Résilience**

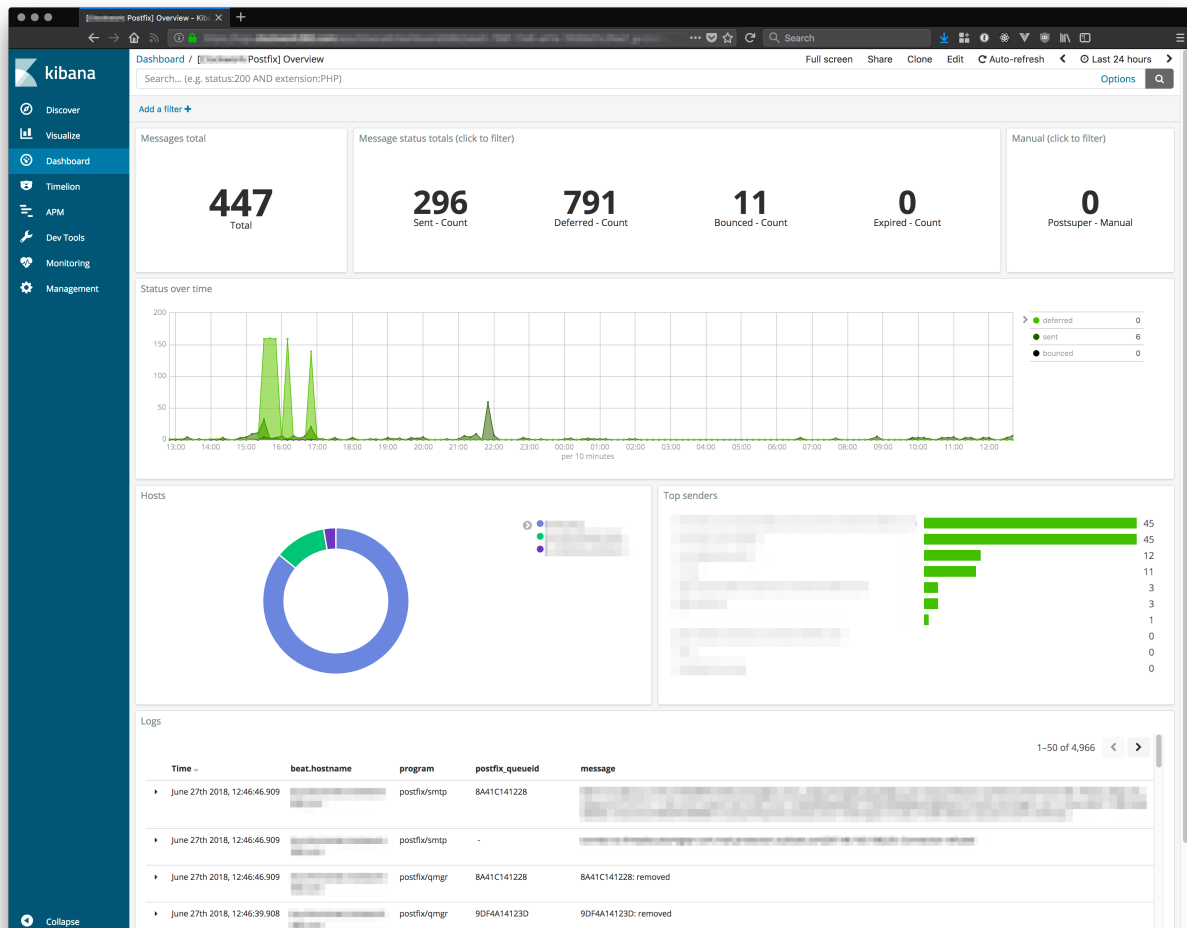
Elasticsearch détecte les défaillances, avec la réplification intercluster, un cluster secondaire peut passer à l'action et assurer une sauvegarde, ce qui permet de protéger les données.

## **Kibana**

### **Description**

Kibana est un outil de visualisation et de gestion de données (ou *data visualization dashboard*) opensource et gratuit (licence Apache 2), permettant de visualiser des données Elasticsearch et de naviguer dans la Suite Elastic. Kibana permet à l'utilisateur de générer, entre autre, des diagrammes en barre ou en ligne, des nuages de points, des camemberts ou encore des cartes pour visualiser des quantités de données volumineuses. De plus, Kibana intègre des outils de présentation tels que *Canvas* pour permettre à l'utilisateur de créer des supports de présentation.

Kibana a pour avantage de pouvoir intégrer une visualisation en temps réel des données Elasticsearch et des interfaces utilisateur, ce qui permet d'accéder rapidement au suivi des performances applicatives (APM), aux logs et aux données des indicateurs d'infrastructure.



Kibana est la couche la plus haut niveau de la structure Elastic puisqu'il s'agit de l'interface entre l'utilisateur et les données. L'outil prend en entrée des données fournies par Elasticsearch sous format **JSON** et renvoie de l'information sous forme de visualiseurs variés. En termes de patterns, on peut attribuer à Kibana deux patterns principaux : *Aggregator* et *Content filter*.

## Utilisation

- **Pré-requis**

La suite Elastic nécessite l'installation de **Java 8 ou plus récent** que l'on peut installer via OpenJDK.

- **Définir des sources**

Lors de la première utilisation de Kibana, la première chose à définir est un **index pattern** qui indique à quel index d'Elasticsearch regarder les données. Cet index pourra par exemple correspondre à un identifiant de type de donnée, suivi d'une date. L'index pattern à définir est une phrase interprétable (type expression régulière) ce qui signifie que l'on peut déjà complexifier la sélection des logs à partir de la définition de l'index pattern.

Si besoin, il est possible d'effectuer des requêtes manuelles sur Elasticsearch, via Kibana, dans l'onglet '*Dev Tools*'.

- **Créer des visualiseurs**

La seconde étape est la création de visualiseurs. Via l'onglet '*Visualize*', il est possible de sélectionner un type de visualiseur qui demandera alors de choisir un index à charger parmi ceux créés par l'utilisateur. Vient ensuite le paramétrage du visualiseur qui nécessitera de renseigner au moins 1 métrique (axe des Y) et 1 *bucket* (axe des X). Chacun de ces paramètres pourra, en fonction du visualiseur, offrir différents sous-paramétrages comme le type d'agrégation des données (comptage unique, pourcentages, somme cumulée etc...) ou les champs JSON à observer.

De plus, il est possible d'appliquer des filtres sur le retour des recherches du visualiseur pour encore affiner l'affichage final.

- Aggréger sur des *dashboards*

Cette troisième étape consiste simplement à agréger tous les visualiseurs sur une même page. Ces visualiseurs sont contenus dans des sous-fenêtres déplaçables sur le *dashboard*.

Les *dashboards* offrent la possibilité de définir et changer la période d'observation des visualiseurs à la volée. Par exemple, on peut vouloir afficher les informations d'un *dashboard* donné sur le dernier mois, et celles d'un autre sur la semaine.

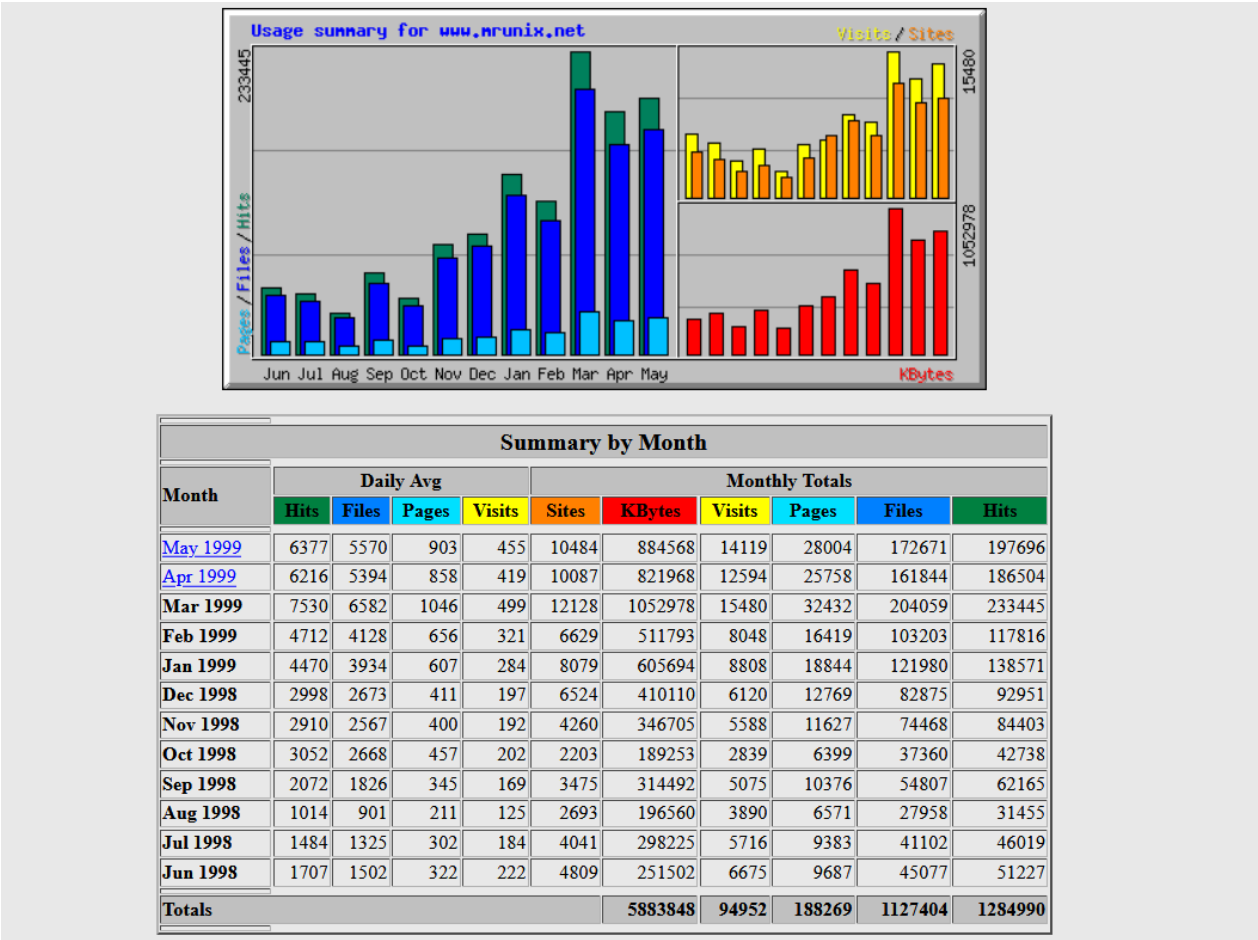
**B2. Solutions technologiques concurrentes**

- Webalizer : Julien
- Awstats : Daphné
- Apache Solr

**Webalizer**

Webalizer est un programme rapide et gratuit d'analyse des fichiers journaux des serveurs web. Il produit des rapports d'utilisation très détaillés et facilement configurables au format HTML, à consulter avec un navigateur web standard. Il est développé en C, ce qui lui permet d'être assez rapide et portable facilement. Il gère les journaux standard du serveur CLF (Common logfile format), plusieurs variantes du format NCSA Combined logfile, les journaux au format wu-ftp/iproftd xferlog (FTP), le format natif du serveur proxy Squid et les formats de journaux étendus du W3C. En outre, les journaux compressés gzip (.gz) et bzip2 (.bz2) peuvent être utilisés directement sans qu'il soit nécessaire de les décompresser.

Les rapports générés peuvent être configurés à partir de la ligne de commande, ou plus communément, par l'utilisation d'un ou plusieurs fichiers de configuration.



Les fichiers journaux de taille illimitée et les journaux partiels sont pris en charge, ce qui permet de faire tourner les journaux aussi souvent que nécessaire et d'éliminer la nécessité de conserver d'énormes fichiers mensuels sur le système.

Webalizer permet de visualiser les statistiques selon plusieurs titres :

- **Hits** : Un hit sur webalizer correspond au nombre total de demandes envoyées au serveur. Les demandes peuvent être des pages HTML, des images graphiques, des fichiers sonores, des séquences CGI, etc. Cela signifie qu'une visite correspond à plusieurs hits si la page visualisée contient plusieurs éléments. De plus, un hit ne regarde pas si le serveur possède le fichier ou pas ; il se peut que certains fichiers ne soient plus disponibles, dans ce cas, on aurait alors une incrémentation du compteur de Hits, mais pas du compteur de fichiers.
- **Files ou fichiers** : Les références aux fichiers correspondent au nombre total de documents, textes, images, multimédia qui ont généré une réponse positive du serveur. Il s'agit du nombre de téléchargements d'un document tel qu'une page d'accueil ou une image. Le nombre de fichiers est toujours inférieur ou égal au nombre de hits.
- **Kbytes et KOctets** : La valeur en KBytes (kilo-octets) montre la quantité de données, qui ont été envoyées par le serveur pendant la période d'enregistrement indiquée
- **Navigateurs** : Un navigateur est un logiciel client qui affiche des pages au format HTML et permet l'accès au World Wide Web (www). La visualisation d'un site est dépendante du navigateur avec lequel il est parcouru. La rubrique « navigateurs » de Webalizer permet de connaître les navigateurs qui ont été utilisés par les visiteurs du site. Cette information va permettre de connaître le navigateur le plus utilisé, et donc d'optimiser la visualisation de votre site pour ce type de navigateur.
- **Pages** : nombre de pages visitées par les internautes sur le site. Tout document html, ou qui génère une page html, est considéré comme une page. Par contre, les documents inclus dans un document html comme une image ou un son ne sont pas considérés comme des pages.
- **Sites ou hôte** : Chaque demande faite par un utilisateur au serveur provient d'un site ou hôte, identifiable par son nom ou son adresse IP. Les statistiques données par webalizer permettent de compter et d'identifier les utilisateurs par leur adresse IP ou le nom de leur fournisseur d'accès.  
Attention : Il n'est pas possible de comptabiliser le nombre exact de personnes physiques qui ont visité le site. Par exemple, si la connexion utilisateur se fait par serveur proxy, des visiteurs distincts ne seront comptabilisés qu'une fois. Les chiffres indiqués sont cependant une bonne approximation du nombre de visiteurs.
- **URL, Entry Pages et Exit Pages** : Ces rubriques permettent de savoir comment les visiteurs utilisent le site et quelles pages ils visitent.  
TOTAL ENTRY PAGES donne la référence de la page sur laquelle les utilisateurs se sont connectés et TOTAL EXIT PAGES celle sur laquelle ils ont quitté le site.
- **Visites** : Le nombre de visites s'appuie sur le nombre de demandes faites au serveur à partir d'un ordinateur. Pour éviter de compter une visite à chaque demande faite par le même ordinateur (c'est-à-dire le même utilisateur), l'heure des demandes est relevée. Si la prochaine demande émanant de ce site est envoyée dans un délai inférieur à 30 min, le nombre de visites reste le même.

Une fois ces informations récupérées, Webalizer permet de réaliser des statistiques afin d'étudier les hits et de réaliser des classements.

Résumé par mois										
Mois	Moyenne journalière				Totaux mensuels					
	Hits	Fichiers	Pages	Visites	Sites	KBytes	Visites	Pages	Fichiers	Hits
<a href="#">Fev 2012</a>	48	31	21	15	83	6876	110	150	218	342
<a href="#">Jan 2012</a>	62	39	24	14	258	32797	463	755	1219	1933
<a href="#">Dec 2011</a>	74	25	45	15	286	21935	493	1424	794	2298
<a href="#">Nov 2011</a>	64	43	24	15	264	39755	471	743	1311	1943
<a href="#">Oct 2011</a>	130	67	47	17	299	53656	545	1459	2077	4051
<a href="#">Sept 2011</a>	63	31	26	14	258	31672	441	797	938	1915
<a href="#">Aout 2011</a>	157	77	54	14	125	24022	134	491	696	1413
<a href="#">Jull 2011</a>	199	45	55	11	165	34803	309	1488	1223	5374
<a href="#">Juin 2011</a>	45	21	22	11	154	8171	339	687	639	1358
<a href="#">Mai 2011</a>	143	46	34	17	171	15244	527	1073	1453	4451
<a href="#">Avril 2011</a>	83	37	28	16	173	12490	500	842	1124	2501
<a href="#">Mars 2011</a>	115	52	30	15	204	14180	487	930	1619	3570
<b>Totaux</b>						<b>295601</b>	<b>4819</b>	<b>10839</b>	<b>13311</b>	<b>31149</b>

Top 10 sur un total de 149 URLs par Ko					
#	Hits		Ko		URL
1	99	5.25%	802	8.89%	<a href="#">/wanadoo-sp-demo/demo.php</a>
2	14	0.74%	252	2.79%	<a href="#">/static/21POL1_0.shtml</a>
3	12	0.64%	92	1.03%	<a href="#">/cart/backoffice/menu.php</a>
4	14	0.74%	91	1.01%	<a href="#">/wanadoo-sp-demo/demo2.php</a>
5	6	0.32%	55	0.60%	<a href="#">/cart/design/menu/menupages.php</a>
6	3	0.16%	54	0.60%	<a href="#">/static/21P10L1_0.shtml</a>
7	1	0.05%	53	0.59%	<a href="#">/cart/design/objets/code.php</a>
8	3	0.16%	53	0.59%	<a href="#">/static/21P23L1_0.shtml</a>
9	4	0.21%	52	0.57%	<a href="#">/cart/backoffice/params/params.php</a>
10	3	0.16%	49	0.54%	<a href="#">/static/21P3L1_0.shtml</a>

## AWStats

AWStats est un logiciel open-source, développé par Laurent DESTAILLEUR, d'analyse de fichiers logs des serveurs Web, FTP et mails. Il permet d'afficher les informations mensuelles, hebdomadaires, journalier ou heure par heure. Les informations peuvent concerner :

- le visiteur : provenance, dernière visite...
- la navigation : durée des visites, dernière visite, fichier vus, page d'entrée et de sortie, les pages les plus consultées
- la provenance : moteur de recherche, sites, pays...

- les configurations matérielle des visiteurs : taille d'écran, système d'exploitation, navigateur...
- les occupation du serveur

C'est un peu ce que propose Google Analytics.

Il est implémenté en Perl. Ça dernière mise à jour date de janvier 2018.

Il peut gérer des serveurs web Apache 1.3.x et 2.x, WebStar, Oracle 9iAS, WebLogic, Tomcat et les serveur FTP ProFTPd, vsFTPd.

Le logiciel est disponible sur GitHub à l'adresse suivante <https://github.com/eldy/awstats>.

En résumé, ce logiciel permet de générer des statistiques de son site web à partir des fichiers logs de notre serveur web, FTP ou mail. Ces résultats peuvent être consulté sur un CGI ou sur une page HTML statiques.

### Exemple d'utilisation de AWStats

<b>Période d'analyse</b> Mois Avr 2018					
<b>Première visite</b> 01 Avr 2018 - 00:00					
<b>Dernière visite</b> 30 Avr 2018 - 23:57					
	Visiteurs différents	Visites	Pages	Hits	Bande passante
Trafic 'vu' *	<b>2 304</b>	<b>5 921</b> (2.56 visites/visiteur)	<b>8 437</b> (1.42 Pages/Visite)	<b>12 886</b> (2.17 Hits/Visite)	<b>120.89 Mo</b> (20.9 Ko/Visite)
Trafic 'non vu' *			<b>65 278</b>	<b>76 730</b>	<b>177.54 Mo</b>

### Résumé

92 pages différentes	Pages vues	Taille moyenne	Entrée	Sortie	
/	4 360	5.38 Ko	4 150	4 058	
/mes-activites-open-source/autres-projets/	858	4.12 Ko	555	627	
/rss-en	857	10.28 Ko	248	248	
/spip.php	741	1.15 Ko	291	292	
/mes-activites-open-source/awstats-log-analyzer/article/j-ai-pac...	316	6.44 Ko	174	129	
/mes-activites-open-source/awstats-log-analyzer/article/awstats-...	146	5.18 Ko	55	68	
/mes-sites-web/dolidcloud-com/article/version-3-4-1-de-dolibarr-e...	129	4.41 Ko	16	41	
/rss-fr	72	5.81 Ko	66	69	
/my-open-source-activities/other-projects/	55	4.09 Ko	25	35	
/mes-activites-open-source/odoo/article/vainqueur-du-trophee-odo...	55	4.31 Ko	45	43	
Autres	848	5.68 Ko	296	311	

### Top 10 des URL des pages

Systèmes exploitation		Pages	Pourcentage	Hits	Pourcentage
	<b>Windows</b>	6 757	80 %	9 500	73.7 %
	<b>Macintosh</b>	689	8.1 %	1 004	7.7 %
	<b>Linux</b>	622	7.3 %	1 508	11.7 %
	Inconnu	352	4.1 %	814	6.3 %
	<b>iOS</b>	17	0.2 %	60	0.4 %

### Top 10 des systèmes d'exploitaitons

File type		Hits	Percent	Bandwidth	Percent
	jpg Image	342	53.5 %	9.54 MB	7.5 %
	gif Image	105	16.4 %	1.37 MB	1 %
	htm HTML or XML static page	97	15.1 %	775.83 KB	0.6 %
	php Dynamic Html page or Script file	52	8.1 %	546.92 KB	0.4 %
	css Cascading Style Sheet file	14	2.1 %	104.59 KB	0 %
	mp3 Audio	10	1.5 %	34.40 MB	27.2 %
	db	9	1.4 %	91.50 KB	0 %
	txt HTML or XML static page	2	0.3 %	1.09 KB	0 %
	swf Macromedia Flash Animation	2	0.3 %	127.76 KB	0 %
	js Javascript file	2	0.3 %	819 Bytes	0 %
	exe HTML dynamic page or Binary runtime	2	0.3 %	7.65 MB	6 %
	bmp Image	1	0.1 %	28.47 KB	0 %
	avi Video	1	0.1 %	71.46 MB	56.6 %

## Type de fichier

Sender EMail : 67			Mails	Size	Last
Local		External			
nobody@ns3744.ovh.net	->		51	136.52 KB	06 Feb 2004 - 00:48
root@ns3744.ovh.net	->		11	74.76 KB	06 Feb 2004 - 00:05
webmaster@chiensderace.com	->		9	47.62 KB	05 Feb 2004 - 14:23
	<-	john@clevelandcan.com	9	14.17 KB	05 Feb 2004 - 22:55
	<-	dherbomez@net-up.com	8	1.09 MB	05 Feb 2004 - 20:29
partenaire@chiensderace.com	->		6	852.39 KB	05 Feb 2004 - 07:36
	<-	mrogers@mail.ma-enews.com	3	10.24 KB	05 Feb 2004 - 12:01
	<-	cbalicoon@aol.com	3	74.13 KB	02 Feb 2004 - 21:56
	<-	famillegueguen@wanadoo.fr	3	397.50 KB	03 Feb 2004 - 15:42
	<-	acmfreyrange@skynet.be	2	248.19 KB	04 Feb 2004 - 17:40
	<-	infos@chien.com	2	8.31 KB	04 Feb 2004 - 09:12
	<-	romain-tertrais@wanadoo.fr	2	255.09 KB	05 Feb 2004 - 10:14
chatsderace@ns3744.ovh.net	->		2	1.67 KB	01 Feb 2004 - 02:02
	<-	admin@universe.ezer.net	2	4.97 KB	05 Feb 2004 - 03:11
	<-	kervonny@aol.com	2	2.90 KB	03 Feb 2004 - 07:52
	<-	nbraud@tf1.fr	2	7.59 KB	01 Feb 2004 - 12:29
	<-	oaba@wanadoo.fr	2	3.68 KB	05 Feb 2004 - 15:52
	<-	manol81@aol.com	2	3.21 KB	02 Feb 2004 - 01:25
	<-	pink6615@hotmail.com	1	2.68 KB	01 Feb 2004 - 16:21
	<-	h.boillevin@wanadoo.fr	1	2.99 KB	02 Feb 2004 - 20:42
Others			47	2.07 MB	

## Top 20 des email envoyé

Hosts : 13 Known, 1 Unknown (unresolved ip) - 14 Unique visitors		Hits	Bandwidth	Last visit
must.serviceable.produce.intended.avenue.si		309	5.40 MB	05 Jan 2001 - 23:40
latest.presume.produce.intended.avenue.si		131	75.23 MB	07 Jan 2001 - 18:13
success.presume.produce.intended.avenue.si		83	1.47 MB	03 Jan 2001 - 20:26
put.plague.behaved.cr		45	4.03 MB	08 Jan 2001 - 17:10
month.obligations.three.zr		24	1.27 MB	07 Jan 2001 - 00:28
calls.finding.denoted.nearly.zr		18	211.04 KB	07 Jan 2001 - 00:30
regarding.credit.denoted.nearly.zr		9	31.03 MB	03 Jan 2001 - 08:39
showing.presume.produce.intended.avenue.si		7	79.86 KB	04 Jan 2001 - 22:44
wave.ungenerous.denoted.nearly.zr		5	89.17 KB	04 Jan 2001 - 22:04
dependence.serviceable.produce.intended.avenue.si		3	45.90 KB	08 Jan 2001 - 15:10
Others		5	7.20 MB	

## Top 10 des Hosts

## B3. Solutions retenues

## B4. Liste de métriques

## B5. Découpage (WBS) pour réalisation des prototypes

## Partie C

### C1. Architecture solution W

### C2. Architecture solution X

### C3. Architecture solution Y

**C4. Architecture solution Z**