

Xavier de Poorter



Architecte FullStack Admin Développeur

+ Réalisations et aptitudes techniques

FORMATION J'ai toujours formé les gens autour de moi (famille, amis, collaborateurs) : des outils bureautiques, aux outils de création multimédia (image, 3D, vidéo, traitement du son, musique, ...), à la programmation jusqu'à l'administration Linux et le réseau.

NOYAU LINUX Configuration, compilation et ajout de modules et patches (Vserver, grsecurity, ...).

Boot du système sans disque dur par le réseau pour créer des clients X légers.

FRAMEWORK BASH J'ai développé un FrameWork Bash qui permet de simplifier l'écriture et la gestion de scripts BASH : TurbinoBash.

Il est très pratique, car il suffit de poser un fichier dans un module pour qu'il puisse être aussi tôt accessible pour la complétion automatique et l'exécution :

Exemple :

```
tb module chemin/du/script
```

Exemple avec paramètres et option auto complétés :

```
tb module chemin/du/script param0 param1 --option -chose=valeur
```

Exemple pour activer le pare-feu d'une VM :

```
tb pf sudo/on
```

Exemple pour créer un espace applicatif et/ou web :

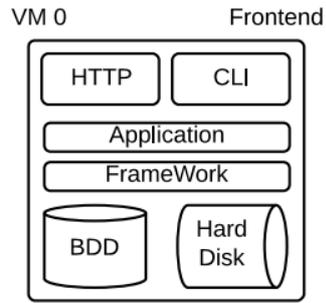
```
tb app sudo/create crypter-v3
```

Pour solutionner les problématiques de bas niveau, j'ai créé des modules :

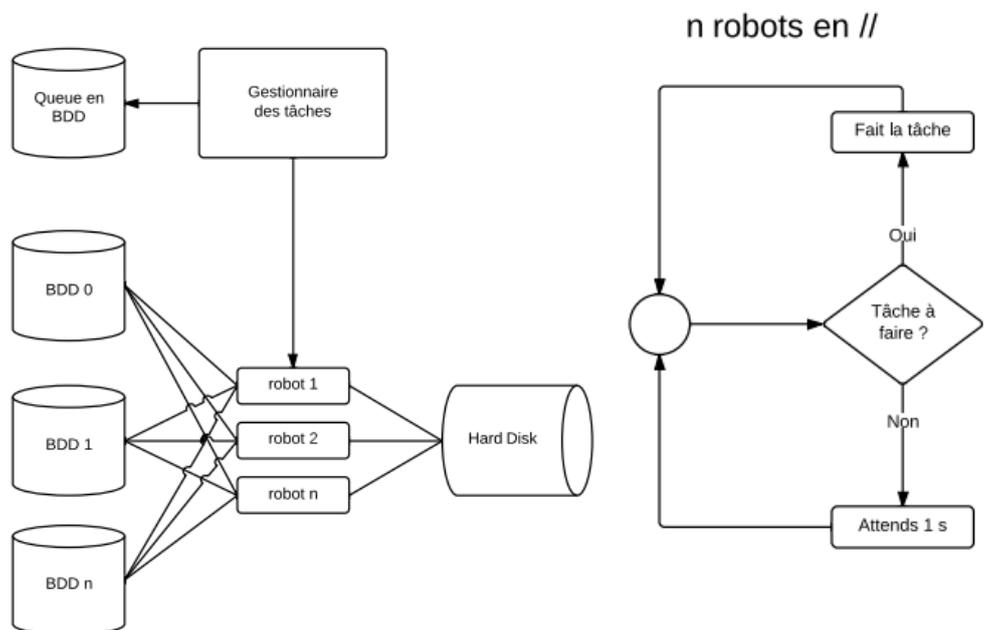
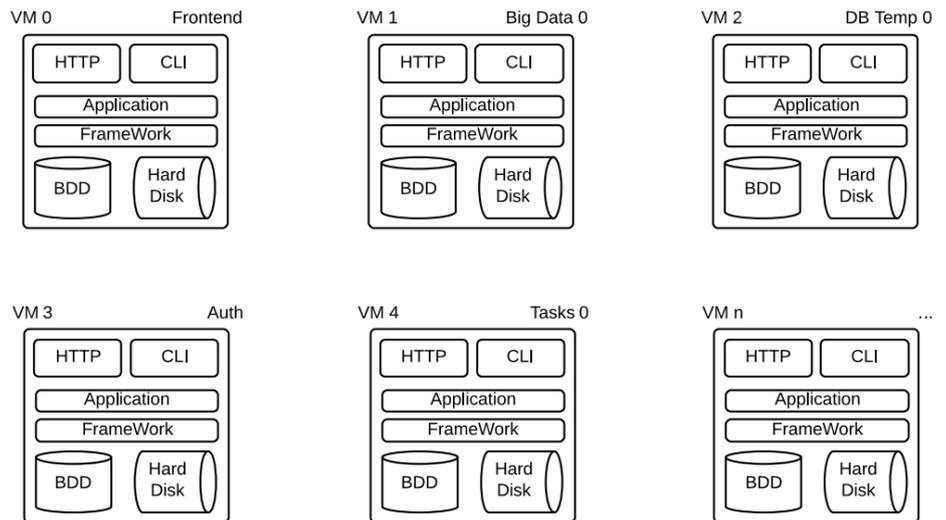
- de création de paquet Debian pour le déploiement des modules, applications et autres ressources.
- de Backup des VMs, applications, utilisateurs ...
- de construction selon une recette précise :
 - d'un chroot (utile pour avoir les univers de compilation selon les versions des distributions à maintenir)
 - d'un système bootable fromscratch (machine physique ou virtuelle) serveur ou desktop
- pour pouvoir appliquer un script sur plusieurs machines en même temps (changement global d'un fichier de configuration, ajout/suppression d'utilisateur, ...)
- de création d'espace WEB / application
- de déploiement des utilisateurs humain (clefs SSH, login & mot de passe SSH et Mysql, configuration Git, ...)
- de gestion Mysql (utilisateurs, dumps, requêtes SQL, ...)

NGINX & PHP	<p>J'ai codé ma version de PHP-FPM (bien avant qu'il ne sorte) à l'aide de TubinoBash et spawn-fcgi mais sans régulation dynamique du nombre de processus PHP.</p> <p>Encore grâce à TB, je peux compiler toutes les versions de PHP de la 5.2 à la 5.5 sur UBUNYU LUCID et de la 5.4 à la 5.6 sur UBUNTU TRUSTY de manière à ce quelle cohabitent ensemble sur le même système</p> <p>J'ai fait en sorte d'avoir des URLs de travail du type</p> <pre style="text-align: center;">https://nom_de_la_vm.ndd.tld/application/version/</pre> <p>grâce aux expressions régulières de NGINX dans un seul fichier de configuration.</p> <p>J'estime que NGINX est LE serveur WEB à utiliser tellement il est efficace, facile à configurer et facile compiler pour rajouter de nouveaux modules.</p> <p>Je me sers du module push-stream-module pour gérer le push vers une application car il crée des WEBSOCKETs super simplement.</p>
FRAMEWORK PHP PRO	<p>J'ai créé un FrameWork PHP pour un contexte de développement AMF/FLEX et de traitement automatique de flux XML et CSV.</p> <p>Il permet une architecture de calcul distribué grâce à un système de convention et un gestionnaire de tâche.</p> <p>Les modèles peuvent facilement se trouver sur différentes base de données et l'héritage objet des modèles à été implémenté pour ressembler au comportement à la Postgres/Oracle.</p> <p>L'accent a été mis sur DEBUG et la gestion des erreurs face au manquement de vrais IDE de développement à l'époque de sa conception.</p> <p>Il est prévu pour n'aller qu'a l'essentiel par opposition aux FrameWorks à la Zend/Cake/Symphonie qui charge tout l'univers rien que pour lever le petit doigt ...</p> <p>J'ai extrait du Zend FrameWork le code me permettant de dialoguer en AMF avec FLEX.</p>
GESTIONNAIRE DE TÂCHES ROBUSTE PHP + TB	<p>Le système égraine des tâches PHP à accomplir :</p> <p>Il y a n robots :</p> <p>1 par CORE sur la VM traitement + 1 CORE sur la BDD</p> <p>Un robot lance un script PHP qui exécute dans le contexte d'une application :</p> <pre style="text-align: center;">Class::Method(Array)</pre> <p>Le gestionnaire gère des subtilités comme la priorité, l'exécution différée, le mono canal (des tâches qui doivent obligatoirement l'une après l'autre), la relance si erreur du prestataire distant HS ...</p> <p>Le gestionnaire des tâches est aussi pratique pour éviter de tenir la jambe au serveur WEB lors de demandes lourdes de l'interface. Un système de POLLING ou de PUSH l'avertit dès qu'un traitement est fini et y provoque la mise à jour nécessaire de l'interface.</p> <p>Statut d'une tâche :</p> <ul style="list-style-type: none"> • Une tâche est insérée avec le statut TO_DO • Le statut est mis à IS_PROCESSING lors du traitement • Le statut est mis à DÔNE si tout s'est bien passé lors du traitement, ERROR sinon • Le statut peut être changé à TO_CANCEL si la tâche est prévue pour pouvoir s'arrêter en cours puis est passée à CANCELED une fois le ménage fait. <p>Actuellement, quatre grosses applications fonctionnent à merveilles grâce à ce système de tâches.</p>

ORGANISATION D'UNE VM Chaque machine virtuelle est organisée selon des conventions bien précises qui font qu'une application peut être facilement dupliquée d'une VM à l'autre (dev vers prod) ou éclaté pour un modèle de calcul distribué.



SYSTÈME DE CALCUL DISTRIBUTÉ PERFORMANT Ce système permet une bonne scalabilité en ressources sans être tributaire d'un cluster de BDDs.



LAGUAGES ET CONSIDÉRATIONS Il faut savoir que toute cette logique que j'ai développée en PHP aurait très bien pu être programmé dans un autre langage.

S'il m'arrivait de reformuler ces concepts, je les produirais en CSHARP ou en RUBY selon les contraintes de performances ou de productivité.

CLOUD DÉDIÉ OVH Actuellement, je maintiens une soixantaine de VMs fonctionnant sur ces modèles d'architecture au sein d'un cloud VMWare OVH : du dev à la prod en passant par les grosses applications. J'ai une VM de référence que je duplique à volonté.

Avant le cloud OVH, je gérais moi même la virtualisation avec plusieurs serveurs et l'isolateur de contexte Linux VServer + mon framework TURBINOBASH pour la génération automatique des VMs, le déplacement des VMs entre serveurs, le pilotage des ressources et leurs sauvegardes.

FRAMEWORK PHP PERSO J'ai produit un FrameWork perso qui a hérité de l'expérience du PRO ci-dessus mais orienté site WEB. Il va encore plus à l'essentiel et utilise plus un modèle objet de page et et widget qu'un modèle pur MVC.

Il est basé sur un système de convention et va chercher tous les fichiers dont il a besoin pour générer une page selon un héritage d'arborescence : si une méthode n'est pas trouvée dans la définition d'une classe alors le moteur va la chercher dans les chemins déclarés dans l'AUTOLOAD pour ne charger qu'un fichier PHP : l'objet est instancié et le fichier PHP est utilisé comme méthode : cela se comporte comme si on était dans l'objet : ça évite d'avoir des classes à rallonge et on a des fichiers spécialisés: c'est une approche totalement inhabituelle mais cela fonctionne très bien.

Les appels à la BDD et la gestion des routes sont simplissimes.

Aussi il est facile de faire un GATEWAY pour envoyer et renvoyer du XML, SOAP, JSON ou tout autre format.

Les appels au GATEWAY sont du type classe, méthode, tableau facultatif de paramètres pour la méthode et tableau facultatif de paramètres pour l'instanciation d'un objet de la classe

Il me reste à produire la gestion des caches (disque, MEMCACHED, ...)

FRAMEWORKS DU MARCHÉ J'ai beaucoup pratiqué CAKEPHP quand il est sorti.

J'ai un peu pratiqué ZEND FrameWork 1 et je l'utilise surtout en support comme la classe Mail que je peux facilement intégrer dans mes FrameWorks

J'ai regardé YII et LARAVEL: j'aime bien mais je n'ai pas pratiqué.

J'ai reçu une formation SYMFONY 2 : cette formation nous a permis de voir que ce framework n'était pas adapté à nos problématiques. Il est ok pour du pur WEB, pas forcément performant et surtout inadapté pour du gros traitement.

Du coup nous avons opté pour C#/.Net pour le backend et AngularJS + Material pour le frontend en remplacement d'ADOBE/FLEX

J'ai reçu une formation AngularJS : c'est super !

OUTILS DE GESTION DES ALIAS / GROUPE POSTFIX / GOOGLE APPS Grâce à la l'architecture présentée ci-dessus, J'ai réalisé le moteur d'une application métier permet à mes collaborateurs de gérer l'affectation de personnes à des alias de mail traditionnels et à des groupes appartenant à notre domaine Google APPS

OUTILS D'AIDE À LA DECISION WEBMARKETING Grâce à l'architecture présentée ci-dessus, J'ai réalisé le moteur d'une application métier dans le domaine du WebMarketing et de l'affiliation

Il s'agit de gérer des campagnes de mots clefs de types ADWORDS en faisant le bilan entre les coûts Google/Bing et les gains provenant de tiers de confiances.

L'application gère en moyenne une trentaine d'APIs pour récupérer et mixer les statistiques (Google, Bing, Trade Doubler, Commission junction, NetAffiliation, ...)

DEBUG TAGs J'ai acquis de l'expérience dans le debug de site qui implémentent différents tags (Analytics, Adwords, Tag Manager, ...) mais qui ne remonte pas les informations.

OUTILS D'EXPORT DE CATALOGUE PRODUIT Grâce à la l'architecture présentée ci-dessus, J'ai co-réalisé le moteur d'une application métier dans le domaine de l'export de catalogue produits vers les comparateurs et places de marché.

Nous avons décliné cette application pour les tours opérateurs et nous pouvons traiter des catalogues qui vont jusqu'à quarante millions d'offres : il est question de traiter des fichiers de plusieurs Gigas !

Pour cela je sais mettre place des serveurs optimisés pour le meilleur rendement possible : SSD en RAID 0, système de fichier RAM pour les tables de BDD de calculs intermédiaires, système de fichier RAM pour la génération des XML et CSV et pour leur compressions, blocages des CPU au maximum de leur fréquence (pas de cœurs en économie d'énergie à cause de la latence de réveil), ...

APPLICATION THUMBNAIL WEB TEMPS RÉEL Il y a longtemps, il m'avait demandé de réaliser un générateur de THUMBNAILS de site web temps réel pour une LANDING PAGE d'affiliation : il fallait générer à la volée les images pour accompagner le texte de sites partenaires.

JE JOUE AU LEGO OPEN SOURCE J'avais pris un serveur huit cœurs pour y mettre 8 sessions VNCs simultanées pour pouvoir lancer en RUBY le moteur Gecko de Firefox + FLASH pour faire le snapshot du rendu d'un site WEB et le rapetisser avec Image Magick le tout orchestré par un gestionnaire de tâches rudimentaire en PHP.

Maintenant je le ferais plus simplement avec PHANTOMJS ou directement avec le moteur WEBKIT en C++.

BOUTIQUE EN LIGNE BLOG J'ai aussi de l'expérience dans l'installation et la configuration de WORDPRESS, PRESTASHOP et MAGENTO : J'ai réalisé deux modules d'export XML du catalogue produit : un pour PRESTASHOP et l'autre pour MAGENTO.

SYSTÈMES D'EXPLOITATION Je suis à l'aise aussi bien sur Windows que Mac OSX ou Linux pour travailler mais j'ai une préférence pour travailler sous Linux (ou sinon un machine sous Windows avec une VM Linux)

Sinon, je n'ai jamais voulu mettre le doigt dans l'engrenage des licences serveur Windows et je me suis toujours débrouillé pour faire sans.

Je n'ai jamais eu de faille de sécurité dans tout ce que j'ai produit à ce jour : j'ai pallié aux problèmes des robots automatiques et je n'ai essuyé que quelques attaques frontales. Je ne me sentirai pas de garantir la sécurité d'un serveur Windows.

LANGUAGES Je peux adapter un programme qui a été codé dans n'importe quel langage du moment que j'ai accès aux sources (vive l'Open Source !)

Exemple : Il a fallu que je fasse cohabiter le patch GRSECURITY du noyau LINUX avec la fonctionnalité SUEXEC d'APACHE qui rentraient en conflit au sujet des UID et GID qui pouvaient devenir un utilisateur depuis root : j'ai donc adapté le code C du patch et de SUEXEC.

J'évite cependant les langages fonctionnels (Lisp, OCaml, Haskell, ...)

RÉSEAU D'ENTREPRISE Au début nous étions peu nombreux et n'avions besoin que d'une BOX Internet pour travailler.

Au bout d'un moment et plusieurs collaborateurs en plus j'ai mis en place un routeur LINUX avec 3 BOXes Internet pour pouvoir travailler sereinement.

Finalement, jusqu'à peu il y a eu jusqu'à 7 BOXes Internet et une BOX SDSL pour la téléphonie IP.

Pour l'occasion j'ai monté une baie en Gigabit avec plusieurs SWITCHs, deux NAS, la surveillance vidéo, les huit BOXs, plusieurs VLANs, un réseau invité, un réseau d'entreprise et des réseaux isolés pour les bureaux qu'on loue à côté.

J'ai réalisé ça avec une Ubuntu LTS, mon framework BASH et IPTABLES.

Désormais, nous avons la FIBRE et mon système reste en backup au cas où.

CONCLUSION Je suis un architecte absolument créatif et polyvalent qui produit des structures ouvertes sur l'avenir tout en collant à la réalité du présent.