

Pourquoi choisir les projets open source

Un document eFORCE FRANCE
Mars 2003

Pourquoi choisir les projets open source...

La crédibilité des principaux projets Open Source s'est accrue au point que se sont ces logiciels comme Apache qui sont désormais considérés comme les références de leurs domaines, voire comme des leaders de leurs champs d'applications.

Pourquoi un tel engouement pour les logiciels à code source ouvert ?

Les arguments techniques en faveur des OSS (Open Source Software) sont nombreux et solides. Voici rassemblées ici quelques-unes des questions les plus habituelles comme « les OSS sont-ils performants ? », « les OSS sont-ils fiables ? », « les OSS sont-ils supportés ? »...

Les OSS sont performants

En se penchant sur le fonctionnement de Linux, on obtient une réponse globale sur la qualité d'ensemble des projets OSS car ce système qui est devenu très populaire ces derniers temps peut être vu comme un « concentré » d'OSS !

En effet, le noyau de Linux est complété par toute une compilation des principaux projets OSS : Apache, SendMail, Perl, Xfree, etc. Or, un bench « TPC-D » d'avril 1999 (déjà un peu ancien mais toujours valable) mettait justement Linux face à Windows NT avec Oracle8 comme base de données commune aux deux systèmes. Les résultats publiés sur l'Internet démontraient un avantage de 2 à 20 fois en faveur de Linux...

Les conditions de ces tests étaient volontairement simplifiées : configurations courantes des serveurs employés¹ et réglages par défaut des configurations logicielles. Le but affiché était de mesurer le potentiel de l'un et de l'autre quand ils sont mis en œuvre par des non-spécialistes (typiquement le contexte PME qui est une des cibles de ces environnements). On peut toujours arguer que Windows NT (ainsi qu'Oracle8) est très sensible au tuning et que les résultats pourraient être radicalement différents avec une phase d'optimisation approfondie... Certes, mais c'est un argument qui vaut aussi pour Linux !

Un portage réalisé dans de meilleures conditions grâce à l'ouverture

Comment expliquer que Linux se soit montré comme plus rapide que NT pour faire fonctionner Oracle8 ?

Tout d'abord, les développeurs d'Oracle ont eu accès à plus d'informations sur les caractéristiques internes du système Linux qu'ils n'en ont eu dans le cas de NT ; le portage a pu être réalisé dans de meilleures conditions et le fonctionnement s'en ressent.

1 - Des serveurs Intel Dell PowerEdge

Ensuite, il est patent que Linux consomme moins de ressources que Windows NT à machine égale. Tout simplement parce que Linux est réellement modulaire car doté d'une architecture propre et qui n'impose pas de fonctionnement incontournable.

Samba a surpris les équipes de Microsoft

Ainsi, l'interface graphique est au choix et basée sur X-Window² mais, encore plus important, cette dernière ne se charge pas automatiquement au démarrage et évite donc ainsi de consommer de la mémoire inutilement dans le cadre d'un fonctionnement en serveur dédié. A l'inverse, Windows NT lance automatiquement et obligatoirement son interface graphique (module GDI) dès le boot et il n'est pas possible de l'éviter...

Enfin, ce sont les équipes de Microsoft qui ont mis à l'épreuve Samba³ et qui ont été surpris par son niveau de performance. Ils l'ont rédigé dans un document interne qui a été diffusé en 1999 (le fameux document « halloween »). Donc, les OSS sont performants ; la bonne tenue de Linux face à Windows NT ou de PHP face à Java le prouve.

Les OSS sont fiables

Le Web foisonne de témoignages multiples et diversifiés sur la fiabilité constatée des logiciels OSS. Ces derniers sont souvent employés pour des applications critiques et ils affichent un fonctionnement de type « *boot and forget* » à la grande satisfaction de leurs administrateurs !

On retrouve ainsi souvent Apache, Perl, PHP, SendMail, Linux, FreeBSD et d'autres dans les configurations des plus grands sites commerciaux du Web (Yahoo, Hotmail, Ebay, etc.) où la disponibilité est un point clé.

Autre domaine où la haute fiabilité n'est pas une option : les équipements qui reposent sur de « l'informatique embarquée ». Ici aussi, on retrouve souvent Linux. Les spécialistes de la sécurité informatique font également souvent confiance à Linux pour mettre en œuvre des *firewalls*. Principalement parce que ce dernier offre un code ouvert, très « revu et corrigé » ainsi que des correctifs fréquents et rapidement disponibles dès qu'une faille est détectée.

Pourquoi les OSS bénéficient-ils d'une réputation de qualité aussi flatteuse ?

Sûrement grâce à l'approche « Darwinienne » des développements qui caractérise (entre autres) les projets OSS !

2 - X-Window est un système de fenêtrage développé par le MIT. Il s'agit désormais d'un standard du marché Unix éprouvé depuis de nombreuses années.

3 - Samba est un projet Open Source permettant d'offrir à Linux la compatibilité avec les réseaux Microsoft dans les deux sens : un logiciel client utilisant le protocole SMB pour s'interfacer avec les serveurs de fichiers Windows NT et Windows 2000, un logiciel serveur utilisant le protocole SMB pour partager les fichiers d'un serveur Linux avec des postes clients sous Windows...

Ainsi, les multiples développements parallèles sont encouragés (ce que ne fera jamais un éditeur traditionnel, d'abord par manque de ressources, ensuite pour éviter les conflits entre les équipes) et, au final, seules les meilleures contributions sont retenues par le groupe « noyau ». Pourtant, au départ ce mode de développement « collaboratif » heurte nos convictions ancrées dans les habitudes et les idées-reçues...

L'incontournable « loi de Brooks »

Dans « Le mythe du mois-homme », Fred Brooks observa qu'on ne peut pas diviser et répartir le temps du programmeur; si un projet a du retard, lui ajouter des développeurs ne fera qu'accroître son retard. Il explique que les coûts de communication et de complexité d'un projet augmentent de manière quadratique avec le nombre de développeurs, alors que le travail réalisé n'augmente que linéairement. Depuis, cela est connu sous le nom de « loi de Brooks », et on la considère en général comme un truisme. Mais si seule la loi de Brooks comptait, le développement de Linux comme il s'est déroulé aurait été impossible.

La programmation non-égoïste contredit la loi de Brooks

Le classique de Gerald Weinberg « La psychologie de la programmation sur ordinateur » (*The Psychology of Computer Programming*) apporta ce qu'on pourrait considérer après coup comme une correction vitale à Brooks. Dans sa discussion sur la « programmation non-égoïste », Weinberg observa que dans les sociétés où les programmeurs ne marquent pas le territoire de leur code, et encouragent les autres à y chercher les bugs et les améliorations potentielles, les améliorations sont drastiquement plus rapides qu'ailleurs.

Mais, encore plus important que le développement initial, c'est surtout au stade de la mise au point que l'effet d'échelle (d'un très grand nombre de participants) que peuvent faire valoir les projets OSS a une influence déterminante sur la qualité résultante...

La loi du plus grand nombre

Car le debug est particulièrement efficace grâce à « la loi du plus grand nombre » : Etant donné un ensemble de bêta-testeurs et de co-développeurs suffisamment grand, chaque problème sera rapidement isolé, et sa solution semblera évidente à quelqu'un.

Ou, moins formellement, « Étant donnés suffisamment d'observateurs, tous les bugs sautent aux yeux ». Les testeurs étant largement plus nombreux que les développeurs et cette grande exposition débouche sur une profondeur de correction inconnue dans le cadre de projets commerciaux où la date de lancement est souvent un élément inamovible qui prime sur tout le reste.

Le cercle fermé est inefficace

Dans la programmation menée de façon traditionnelle (petit groupe travaillant en cercle fermé), les bugs et les problèmes de développement représentent des phénomènes difficiles, ennuyeux, insidieux, profonds. Il faut à une poignée de passionnés des mois d'observations minutieuses avant de bien vouloir se laisser

convaincre que tous les bugs ont été éliminés. D'où les longs intervalles séparant les mises à jour, et l'inévitable déception quand on se rend compte que la mise à jour tant attendue n'est pas parfaite.

Dans un cadre de projet OSS, vous supposez qu'en général, les bugs sautent rapidement aux yeux lorsque de nombreux contributeurs enthousiastes se précipitent sur toute nouvelle mise à jour. C'est pourquoi vous mettez à jour souvent afin de disposer de plus de corrections.

Si ce principe était faux, alors tout système aussi complexe que le noyau Linux aurait dû s'effondrer depuis longtemps sous la masse des interactions néfastes et imprévues et à cause des bugs « profonds » non découverts.

L'idée reçue de la motivation par l'utopie

Certains diront « le développement des logiciels Open Source reposent sur la notion de communauté agissant par idéal, n'est-ce pas dangereux pour le futur de ce mouvement et l'intérêt qu'il représente en matière de qualité ? ». Non, l'Open Source ne repose plus seulement sur le côté utopie désintéressée. Si les développeurs contribuent, c'est d'abord qu'ils y trouvent leur intérêt !

Celui qui corrige un bug fait remonter sa correction afin que le bug disparaisse pour de bon et ne risque pas de revenir avec la version N+1, c'est ça la vraie logique vertueuse de l'Open Source...

Le modèle du développement collaboratif à code source ouvert est le seul qui ait produit des résultats tangibles en matière de qualité et de rapidité d'évolution. C'est la notion d'ouverture totale qui est à l'origine de ce changement d'échelle dans la production de logiciel.

Le danger du fractionnement

Les détracteurs affirment aussi que les logiciels reposant sur le développement collaboratif vont se fractionner en de multiples versions incompatibles les unes avec les autres. Le danger de fractionnement, s'il était réel, aurait déjà produit ses effets. Or, c'est plutôt au contraire qu'on assiste : des projets qui gagnent en sérieux au fur et à mesure qu'ils étendent leur audience (documentations, spécifications standards, support d'acteurs établis, etc.).

Distribuez tôt, mettez à jour souvent

Donc, on peut « paralléliser le debug ». Même si le debug requiert que les testeurs communiquent avec un développeur chargé de la coordination, une réelle coordination entre testeurs n'est pas indispensable. Ainsi, le debug n'est pas la proie de cette envolée de la complexité et des coûts d'organisation qui rend problématique l'ajout de nouveaux développeurs.

En pratique, le monde des OSS n'est quasiment pas affecté par la perte théorique d'efficacité qui découle du fait que plusieurs testeurs travaillent sur la même chose au même moment. L'une des conséquences de la politique du « distribuez tôt, mettez à jour souvent » est de minimiser les pertes de ce type en propageant au plus vite les corrections qui sont revenues au coordinateur.

C'est grâce à ces mécanismes qu'on constate qu'effectivement, la fiabilité des logiciels Open Source (due à la qualité du code) est largement meilleure que celle constatée dans les logiciels développés en circuit fermé...

Les OSS sont supportés

Le problème du support technique peut, à lui seul, freiner bien des ardeurs en faveur de logiciels libres. L'opinion générale qui prévaut est qu'il n'y a tout simplement pas de support technique pour ce type de logiciels !

Cette perception est très éloignée de la réalité. InfoWorld a même décerné le prix du meilleur support technique à la communauté Linux en 1998... Les communautés d'utilisateurs et de développeurs qui entourent les OSS sont connues pour être très réactives : lorsqu'un bug est identifié, le correctif associé est mis en ligne dans des délais très courts, le plus souvent. Mais les décideurs ne peuvent se contenter de ce type de support, pour efficace qu'il soit, ne serait-ce que pour des conditions psychologiques !

Une offre commerciale de services autour des OSS

Pour commencer, ce sont les éditeurs qui diffusent Linux et les autres projets open sources majeurs sous forme de distributions qui offrent également des services de formation, hot line, etc.

En dehors de ce premier cercle assez naturel, on constate que des sociétés se sont créées pour répondre à la demande. Autour de SendMail ou d'Apache, des sociétés spécialisées proposent des contrats de support centrés sur ces logiciels.

Encore plus significatif, ce sont des géants comme HP ou IBM qui offrent désormais le support technique de Linux au niveau mondial et disponible en permanence (7j/7 et 24h/24).

Avec une telle diversité de l'offre, la question du support technique se résume désormais au choix d'une démarche.

Quelle évolution pour les OSS ?

Derrière cette question, on peut sentir toute l'incertitude (voire même l'inquiétude) que peut générer le modèle de développement collaboratif dans l'esprit des décideurs !

Elle révèle aussi l'incompréhension profonde de la motivation des contributeurs aux projets OSS... Effectivement, si les contributeurs de tel ou tel projet OSS ne faisait cela que pour la beauté du geste, il aurait de quoi douter de la pérennité du résultat. En fait, la plupart des développeurs qui contribuent activement à l'évolution d'Apache, PHP ou Linux sont directement intéressés et bénéficient de la popularité du logiciel sur lequel ils travaillent « bénévolement ». Un seul exemple : l'éditeur Red Hat a dédié une équipe de ses salariés pour participer exclusivement au développement du noyau Linux.

Red Hat préserve ses intérêts

En agissant ainsi, Red Hat ne fait pas preuve d'altruisme mais préserve ses intérêts. En effet, plus le noyau est évolué, plus Linux se diffusera et plus les produits et services de Red Hat pourront toucher une clientèle qui va en s'élargissant...

Les sociétés qui font du développement Web et qui ont construit leurs réputations sur leurs maîtrises d'Apache, PERL ou PHP agissent de la même façon et pour les mêmes raisons.

Les OSS sont une des illustrations du principe mutualiste où les participants retirent plus qu'ils n'apportent (l'autre cas célèbre est l'Internet lui-même).

SendMail, un exemple de cohabitation mutuellement profitable

Un exemple pour illustrer cet aspect d'entraînement du marché : le célèbre logiciel de messagerie SendMail (un OSS donc disponible gratuitement avec son code source) est également commercialisé dans une version plus "accompagnée" par la société créée par le développeur (Eric Allman) à l'origine de SendMail. Cette version (SendMail Pro) n'est pas une divergence de SendMail qui continue d'évoluer et d'être la base de la version "Pro", c'est en fait une application d'administration (reposant sur une interface graphique) de SendMail afin d'en faciliter la mise en oeuvre. D'autres exemples comme celui de SendMail illustrent la cohabitation bénéfique entre développement basé sur un code source ouvert et commercialisation avec valeur ajoutée.

Un projet qui reste indépendant de son créateur

Ce qu'il est important de comprendre (dans le cas de SendMail mais c'est pareil pour tous les autres cas similaires) c'est que même Eric Allman (le développeur/créateur originel de SendMail) n'a pu tirer profit de sa création sur le plan commercial qu'en créant une société en marge du mouvement sans pouvoir prétendre à aucun moment contrôler, diriger ou réclamer des droits sur sa création. Au contraire, il est même toujours de son intérêt de voir la version libre de SendMail se diffuser largement (et aussi d'évoluer régulièrement) afin d'élargir le marché de ses additifs...

Une autre raison objective milite en faveur d'une pérennité supérieure des OSS face aux logiciels commerciaux : comme les OSS ne sont pas (et ne peuvent pas) être sous le contrôle d'une entité privée ou publique, il n'y a pas de danger de voir les projets arrêtés pour des raisons marketing ou politiques. Au contraire, les projets OSS naissent souvent pour assurer la pérennité d'un logiciel, l'histoire d'Apache en est une illustration.

Arrêté pour raisons « politiques »

En revanche, dans la sphère commerciale ce type de coup d'arrêt est chose courante (par exemple quand Sun décida de faire passer ses clients de SunOS à Solaris ou quand le même Sun racheta trois serveurs d'applications –Net Dynamics, Forté et Netscape ex-Kiva- sans être capable de choisir ensuite, les tuant finalement tous les trois !) au gré des changements de stratégies des éditeurs et constructeurs. Les clients sont très regardants sur la question de la pérennité des produits sur lesquels ils font reposer leurs développements et on les comprend !

Or, il est de plus en plus clair qu'il n'y a pas de pérennité quand les acteurs (éditeurs, constructeurs) s'en mêlent, les seuls projets vraiment pérennes, ce sont les projets Open Source.

Nous venons de le voir, les logiciels Open Source sont performants, fiables, supportés et pérennes.

Je préfère choisir du sûr

Toutefois, il y a quand même un côté rassurant à acheter la licence d'un logiciel commercial, car l'éditeur a des obligations. Le décideur cherche souvent à se rassurer avec un nom connu et sa direction générale ne lui reprochera jamais d'acheter de l'IBM, de Microsoft, de l'Oracle ou de SAP... Que devient cette « protection » avec les logiciels Open Source ?

C'est tout à fait exact, cet obstacle existe et le réflexe "je préfère choisir du sûr" est bien présent dans les grandes organisations, il serait absurde de le nier. Cependant, la croyance dans les "obligations" de l'éditeur et dans sa capacité à y faire face est tout à fait illusoire !

La vérité est que l'industrie des logiciels propriétaires est en crise comme est en crise l'industrie du PC. Il s'agit d'une crise latente mais qui va en s'aggravant.

L'industrie du logiciel est en crise

Un symptôme significatif qui indique que le point de rupture est proche : le site Web BugNet (spécialisé dans l'information sur les bugs des logiciels, comme son nom l'indique) a renoncé en 1999 à attribuer son prix annuel de la meilleure correction ! En effet, BugNet estimait alors que la qualité globale des logiciels commerciaux n'a jamais été aussi basse et que les éditeurs sont de plus en plus désinvoltes dans leur façon de traiter le support technique. La situation ne s'est pas améliorée depuis et cela commence à se savoir.

Les éditeurs sont de moins en moins considérés comme un "recours" solide. Donc, l'argument du parapluie sera de moins en moins "politiquement correct". Comment justifier, face à une direction générale, d'avoir retenu un produit coûteux s'il n'est pas généralement considéré comme le meilleur de son domaine ?

Or, dans le cas des serveurs Web (par exemple), le meilleur, c'est Apache, un projet Open Source !

Idem en matière de proxy, de cache et de multiples autres exemples.

Conclusion

Non, la vague open source n'est pas un phénomène de mode.

Mais, ce qui va accélérer encore l'adoption définitive des logiciels Open Source comme composants de base des projets Ebusiness, c'est le ralliement massif des prestataires de services... En effet, pour tous les intégrateurs développant pour les clients, il y a une grande différence entre utiliser une boîte noire (ce que sont en fait les logiciels propriétaires) et utiliser les projets Open Source qui sont eux, complètement transparents et ouverts.

Vis-à-vis des clients, le développeur/intégrateur doit être en mesure de garantir la bonne fin du projet. Cette garantie réside essentiellement dans le niveau de compétence et la maîtrise technique réelle affichée sur telle ou telle solution.

Or, cette maîtrise sera toujours plus facile à acquérir sur les logiciels Open Source et toujours plus difficile sur les logiciels propriétaires et fermés. Pour une raison simple : les logiciels propriétaires gardent leurs codes internes inaccessibles et cela restera ainsi. Avec les outils Open Source, la latitude d'intervention des intégrateurs est bien plus grande et aussi, par la même, la capacité d'aboutir avec succès.

Sans accès au code source, pas de cercle vertueux découlant de la " très grande exposition " et du *peer review*. La qualité reste donc, au mieux, limitée et les évolutions, lentes.

On se retrouve tributaires de services de support débordés et faiblement compétents des éditeurs, juste capable d'enregistrer nos plaintes et de répondre " attendez le prochain service pack, cela devrait aller mieux "...

Notre époque ne peut plus se contenter d'un traitement aussi désinvolte c'est pourquoi les clients sont de plus en plus intéressés par les projets open source et nous ne pouvons que les soutenir et les encourager dans cette évolution.

Alain Lefebvre - directeur de la stratégie technique d'eFORCE FRANCE. Mars 2003.

À propos du Groupe eFORCE :

Créé en Mars 1999, le Groupe eFORCE est aujourd'hui un acteur majeur aux Etats-Unis dans la mise en place de solutions informatiques stratégiques qui contribuent à l'enrichissement de la chaîne de valeur de l'entreprise – eBusiness, CRM, EAI, Portails et Business Intelligence. Les fondateurs, issus du secteur des services informatiques et du conseil (Siebel Systems, EDS, KPMG, Booz Allen, Cap Gemini), ont créé le Groupe eFORCE avec l'objectif de mettre en place des applications informatiques stratégiques à forte valeur ajoutée pour les grands comptes avec 100 % de clients satisfaits.

Pour répondre aux besoins de ses principaux clients : Alcatel, American Express, DHL, Hilton, King Fisher, Mitsubishi, Nortel, Novartis, Visa, notamment sur des projets internationaux, le Groupe eForce met en place des équipes projet multi-pays qui permettent de coordonner et de déployer les applications. Avec 250 collaborateurs dans le monde, le Groupe eForce est aujourd'hui présent aux Etats-Unis (Austin, Boston, Dallas, Houston, Los Angeles, New-York, Silicon Valley) en Europe (Grande-Bretagne et France) et en Inde.

Prenez contact avec nous : Thierry Filho (directeur général) à tfilho@eforceglobal.com et Alain Lefebvre (directeur de la stratégie technique) à alefevre@eforceglobal.com.

Technologies**Services proposé par eFORCE France**

Base de données fédérées avec middleware XML

- Séminaires de formation
- Conseil
- Développement
- Intégration

Bus de services basé sur les web services et les MOM

- Séminaires de formation
- Conseil
- Développement
- Intégration

Externalisation des applications par ASP avec intégration par web services

- Séminaires de formation
- Conseil
- Développement
- Intégration