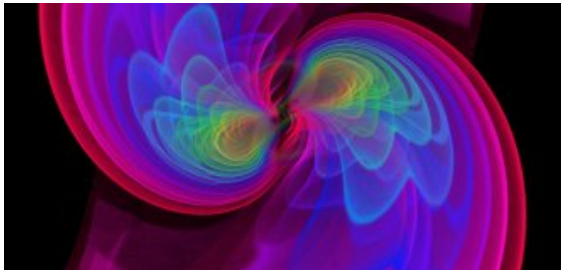




**A LA UNE**

## Les ondes gravitationnelles dans le cloud



Les calculs associés à la détection des ondes gravitationnelles vont engendrer des pics de charge espacés dans le temps et semblent d'excellents candidats à l'utilisation de l'élasticité offerte par le cloud computing. L'IN2P3 enquête.

La mise en évidence expérimentale récente des ondes gravitationnelles par les expériences LIGO/Virgo a ouvert un nouveau champ de l'astronomie. Les objets massifs comme les trous noirs (effondrement d'une étoile très massive à la fin

de sa vie) sont étudiés classiquement de manière indirecte (disque d'accrétion, jets...). L'observation des ondes gravitationnelles en provenance de ces objets permet d'obtenir des informations très précises sur leur nature comme leurs masses et leurs vitesses de rotation.

Les ondes gravitationnelles sont des déformations de l'espace-temps créées par l'accélération d'objets massifs. Elles sont des vecteurs d'information qui nous parviennent tout comme la lumière. Ces oscillations de l'espace-temps modifient la distance entre des objets en chute libre (massetest, Terre, étoiles à neutron de type pulsar...) et peuvent se détecter via plusieurs méthodes : chronométrie des pulsars (détection semi-directe), interférométrie spatiale (eLISA, evolved Laser Interferometer Space Antenna) et interférométrie au sol (LIGO, Virgo...). Le laboratoire APC est impliqué dans le projet eLISA à travers (...)

[lire la suite](#)

**Interview**  
**"L'évolution de notre infrastructure de calcul est un réel défi. Face au déluge de données issues de nos expériences, nous devons être proactifs"**



**Volker Beckmann, Directeur Adjoint Scientifique Calcul et Données à l'IN2P3**

[lire l'interview](#)

**■ Infrastructure**



### Gestion des journaux et métriques au CC-IN2P3

Contexte Le projet COLOSS, visant à moderniser la plateforme de gestion de journaux [1] du CC-IN2P3, est entré progressivement en production depuis le début de l'année 2016. La plateforme gère tous les jours plus d'un milliard d'événements, dont 10% sont stockés sur disque pendant un an. La provenance des informations est très variée : Journaux du système d'exploitation Messages applicatifs Événements d'appareillages d'infrastructure (climatiseurs, onduleurs, etc.) Mesures de performance des serveurs (cpu, mémoire, réseau, etc.) Comptabilité des "jobs" exécutés sur la ferme de calcul Le système permet aux administrateurs et exploitants de services du CC-IN2P3 d'injecter leurs données d'exploitation et de les consulter par la suite. Deux interfaces utilisateur sont mises à disposition à cet effet. La première permet l'analyse instantanée du flux d'événements via un système d'abonnement synchrone. Elle est utilisée notamment pour la remontée d'alertes et le suivi

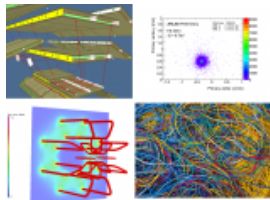
**■ Développement**



### Introduction à Go & Retours d'expérience

Il n'y a plus de "Free Lunch" possible : la loi de Moore [1] n'est plus aussi aisée à mettre en œuvre que par le passé. Les développeurs de logiciels, ainsi que les scientifiques, doivent maintenant se familiariser avec la loi d'Amdahl [2]. En effet, la fréquence d'horloge des processeurs n'augmente plus avec chaque nouvelle génération : les processeurs gagnent de plus en plus de cœurs, mais sans changer la fréquence par rapport à la génération précédente. L'augmentation du nombre de cœurs appelle à une utilisation plus répandue du parallélisme dans les logiciels développés par les communautés HEP et Astro. Traditionnellement, le parallélisme a été exploité au niveau du traitement des événements dans les logiciels de reconstruction et de simulation des expériences de physique des particules : il suffit de lancer N jobs sur les fermes de calcul, la Grille ou un cloud. Plusieurs événements peuvent être traités en parallèle, même si les algorithmes de

**■ Gestion de données**



### ATLAS, CMS et les « Conditions Data »

ATLAS et CMS, expériences phare au CERN, ont lancé un projet commun pour le développement d'une infrastructure de gestion de données de type Conditions de nouvelle génération. Au cœur de cette infrastructure, la « Conditions DB » est une base de données qui enregistre en continu l'état du détecteur et de tous les appareillages de l'expérience (déclenchement, alignement...) pendant la prise de données. Ces Conditions Data doivent ensuite pouvoir être lues de façon très efficace et en parallèle par les dizaines de milliers de jobs simultanés de la phase de reconstruction des événements, puis des programmes d'analyse un peu partout dans le monde. Les expériences LHC ont terminé en 2013 la première phase de prise de données (Run1). Celle-ci a permis entre autres la découverte du boson de Higgs. Pour obtenir ce résultat, les expériences ATLAS et CMS ont analysé dans un environnement de

**■ Infrastructure**



### « CYCLONE » : Gestion dynamique d'applications multi-cloud

Les infrastructures de « cloud » de type « Infrastructure as a Service » (IaaS) s'adressent aux développeurs et aux applications métiers. Elles connaissent un grand succès grâce à leur flexibilité permettant l'utilisation de ressources à la demande. Mais les utilisateurs, et les développeurs eux-mêmes, peuvent parfois avoir du mal à maîtriser la gestion d'applications et de services sur une telle infrastructure, dont les outils sont traditionnellement entre les mains des administrateurs « système et réseau ». Ce problème devient d'autant plus ardu que les applications et leurs déploiements se complexifient, se diversifient, en nécessitant, aujourd'hui, l'utilisation non plus d'un seul cloud, mais plusieurs clouds simultanément. Ce choix du « multi cloud » est motivé par un souci de proposer une meilleure résilience, de meilleurs temps de réponse, des politiques spécifiques de placement ou d'ordonnancement de calcul ou de données, ou encore, de réduire les coûts. Le projet

### Arrêt du service Webmail Horde le 31/12/2016

Le Centre de Calcul propose depuis de nombreuses années un service Webmail basé sur le logiciel (...)

[en savoir plus](#)

### Calcul parallèle et GPGPU au Centre de Calcul

Cette année 2016 apporte deux grands nouveautés en matière de calcul au CC-IN2P3 ! En février (...)

[en savoir plus](#)

### Les Webinaires, le retour

Depuis le 31 mars 2016, les Webinaires se sont installés à nouveau dans le paysage de (...)

[en savoir plus](#)

### Indico : 10 000e demande de compte !

Le 24 juin à 13:05, indico@in2p3 recevait sa 10 000e demande de compte : Bien sûr, cela ne (...)

[en savoir plus](#)

### Le projet PLUME devient le service FENIX

Le site PLUME a été mis en veille fin 2013. A l'été 2015, Resinfo (avec le soutien de Mathrice) a (...)

[en savoir plus](#)

### Agenda

#### J12016 : Ouverture des inscriptions !

Les prochaines Journées Informatiques de l'IN2P3 et de l'IRFU se tiendront du 26 au 29 septembre (...)

[en savoir plus](#)

en temps réel des conséquences d'une intervention. Sa faible latence (au plus (...))

[lire la suite](#)

reconstruction sont toujours appliqués de manière séquentielle à chacun de ces événements. Cependant, chaque nœud de calcul (...)

[lire la suite](#)

calcul distribué des centaines de Pétaoctets de données de Physique (ou Event Data). En complément, d'autres données, que nous appelons Conditions Data (faute de trouver un terme adéquat en français), (...)

[lire la suite](#)

européen CYCLONE, démarré en (...)

[lire la suite](#)

## Offres d'emploi

**CDD IE : Administration des systèmes et réseaux informatiques - LAL, Orsay**  
[en savoir plus](#)

**CDD : Développeur d'applications web - CC-IN2P3, Villeurbanne**

[en savoir plus](#)



## Equipe

**Directeur de la publication** : Alain FUCHS.

**Responsable éditorial** : Pierre-Etienne MACCHI.

**Comité de rédaction** : Frédérique CHOLLET LE FLOUR, Virginie DELEBARRE DUTRUEL, Christian HELFT, Dirk HOFFMANN et Gaëlle SHIFRIN.

## Archives

## Abonnement

Pour vous abonner/désabonner, suivez ce [lien](#).

## Proposer un article

Vous souhaitez proposer un article ? Envoyez un mail à [LettreInformatique@in2p3.fr](mailto:LettreInformatique@in2p3.fr).

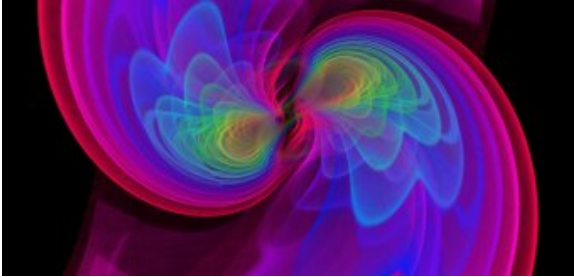


© 2016 CCIN2P3



## Les ondes gravitationnelles dans le cloud

Technologie



Simulation numérique des ondes gravitationnelles émises lors de la fusion de deux trous noirs - © simulation numérique S. OSSOKINE, A. BUONANNO (Max Planck Institute for Gravitational Physics), visualisation W. BENGER (Airborne Hydro Mapping GmbH)

**Les calculs associés à la détection des ondes gravitationnelles vont engendrer des pics de charge espacés dans le temps et semblent d'excellents candidats à l'utilisation de l'élasticité offerte par le cloud computing. L'IN2P3 enquête.**

La mise en évidence expérimentale récente des ondes gravitationnelles par les expériences LIGO/Virgo a ouvert un nouveau champ de l'astronomie. Les objets massifs comme les trous noirs (effondrement d'une étoile très massive à la fin de sa vie) sont étudiés classiquement de manière indirecte (disque d'accrétion, jets...). L'observation des ondes gravitationnelles en provenance de ces objets permet d'obtenir des informations très précises sur leur nature comme leurs masses et leurs vitesses de rotation.

Les ondes gravitationnelles sont des déformations de l'espace-temps créées par l'accélération d'objets massifs. Elles sont des vecteurs d'information qui nous parviennent tout comme la lumière. Ces oscillations de l'espace-temps modifient la distance entre des objets en chute libre (masse-test, Terre, étoiles à neutron de type pulsar...) et peuvent se détecter via plusieurs méthodes : chronométrie des pulsars (détection semi-directe), interférométrie spatiale (eLISA, evolved Laser Interferometer Space Antenna) et interférométrie au sol (LIGO, Virgo...). Le laboratoire APC est impliqué dans le projet eLISA à travers la simulation, l'analyse de données et dans la mise en place d'un prototype de DPC (Data Processing Center).

La mission spatiale eLISA, qui sera lancée vers 2030 en tant que mission large L3 de l'ESA, a pour but d'étudier les ondes gravitationnelles à partir de l'espace. La mission consiste à mettre en orbite un interféromètre constitué de trois satellites séparés par quelques millions de kilomètres afin de mesurer précisément le déplacement engendré par le passage des ondes gravitationnelles. A la fin de l'année 2015, le démonstrateur LISAPhfinder a été lancé afin de prouver que la technologie d'un interféromètre spatial était réaliste : deux masses-tests ont été placées dans des conditions de chute libre afin de caractériser le bruit instrumental. Les résultats du démonstrateur seront présentés par l'ESA ce mois-ci.

A l'APC, l'équipe eLISA est en charge du prototype du DPC, dont le but est de mettre en place les outils qui permettront d'organiser de manière optimale l'analyse des données de la mission. Dans ce cadre et au titre de R&D, l'équipe évalue l'apport de nouvelles technologies comme le cloud computing de type IaaS (Infrastructure-as-a-Service, solutions StratusLab et OpenStack) et PaaS (Platform-as-a-Service, outil SlipStream), les conteneurs Docker ou l'intégration continue avec Jenkins. L'étude de phase 0 de la mission par le CNES a permis de montrer que l'analyse de données de la mission n'allait pas se réaliser de manière continue (détection de certaines sources ponctuellement dans le temps). Pour gérer ces pics de charge, il sera probablement judicieux d'utiliser un système hybride cluster (charge continue) / cloud (pic).

Dans le cadre de l'étude de la gestion des pics de charge réalisée par Antoine Petiteau (chercheur) et Cécile Cavet (ingénieur), le cloud IaaS StratusLab@LAL, maintenant OpenStack@LAL (Laboratoire de l'Accélérateur Linéaire, IN2P3, à Orsay), a été utilisé. Ce cloud public fait partie de la fédération française de clouds académiques orchestrée par France Grilles (FG-cloud), qui regroupe plusieurs infrastructures de l'IN2P3 (CC-IN2P3 à Lyon, IPHC à Strasbourg, LUPM à Montpellier...). Le cloud IaaS permet d'instancier des machines virtuelles à la demande et donc de créer un cluster virtuel semblable au cluster local du FACe (Centre François Arago, centre de traitement de données spatiales de l'APC). A l'aide de l'outil SlipStream, le déploiement du cluster virtuel est réalisé automatiquement via un orchestrateur qui va aussi permettre d'ajouter des nœuds de calcul (passage à l'échelle horizontale) à la demande.

Cette étude a donc permis de montrer que la technologie de cloud est très bien adaptée aux problématiques des missions spatiales. La possibilité de basculer rapidement des calculs depuis un cluster local vers un cluster virtuel est en effet réalisable avec un gestionnaire de déploiement automatique comme SlipStream. Les machines virtuelles permettent aussi à des utilisateurs extra-institut comme ceux du consortium eLISA d'avoir accès facilement aux codes d'analyse et aux données de la mission. Le cloud est donc approprié pour répondre aux contraintes d'un large éventail d'applications scientifiques.

Pour en savoir plus :

- Présentation aux journées SUCCES 2015 : [http://succes2015.sciencesconf.org/conference/succes2015/C\\_cavet.pdf](http://succes2015.sciencesconf.org/conference/succes2015/C_cavet.pdf) et [http://webcast.in2p3.fr/videos-etude\\_des\\_ondes\\_gravitationnelles\\_de\\_lespace\\_au\\_cloud](http://webcast.in2p3.fr/videos-etude_des_ondes_gravitationnelles_de_lespace_au_cloud)
- **FG-cloud**
- **SlipStream**, service France Grilles

Cécile CAVET (APC)



## "L'évolution de notre infrastructure de calcul est un réel défi. Face au déluge de données issues de nos expériences, nous devons être proactifs"

Volker Beckmann, Directeur Adjoint Scientifique Calcul et Données à l'IN2P3



**- Pourriez-vous rappeler brièvement votre parcours et expliquer ce qui vous a amené à accepter le rôle de DAS Calcul et Données ?**

J'ai fait une thèse en astrophysique à l'Université de Hambourg durant laquelle je travaillais déjà sur les balayages du ciel et l'identification automatique d'objets célestes. Après cela, j'ai fait un post-doc à l'INTEGRAL Science Data Center (ISDC) à Genève afin de développer un pipeline d'analyse de données pour l'un des instruments du satellite INTEGRAL de l'ESA. En 2003, j'ai été employé au Goddard Space Flight Centre de la NASA, près de Washington D.C. en tant que responsable du support utilisateurs d'INTEGRAL et j'ai eu également l'opportunité d'enseigner l'astrophysique à l'Université du Maryland (Comté de Baltimore). Je suis ensuite retourné à l'ISDC en 2007 en tant que responsable de projet pour les activités d'analyse de données d'INTEGRAL. J'ai rejoint le CNRS en 2009 et suis devenu responsable scientifique du Centre François Arago, un centre de calcul et d'analyse de données pour les expériences d'astroparticules au sol et dans l'espace. En 2010, j'ai obtenu mon Habilitation à Diriger les Recherches et ai été titularisé en 2013 comme ingénieur de recherche expert en calcul scientifique.

**- Quel est le périmètre de votre mission et comment s'articule votre rôle au sein de la direction de l'Institut ?**

Le principal objectif de la nouvelle direction de l'IN2P3 est de développer et de coordonner les activités de recherche en calcul appliqué à la physique nucléaire, à la physique des particules, l'astroparticule et à la cosmologie (domaines regroupés sur l'acronyme NPAC) au sein de l'Institut. Cela inclut de coordonner ces activités de recherche avec les institutions partenaires, aussi bien en France qu'à l'étranger, dont l'Union Européenne.

Un point très important est la formation : j'entends par là non seulement la formation sur des sujets spécifiques mais aussi la capacité à faire un master, une thèse et finalement, à obtenir une habilitation dans ce domaine du calcul appliqué, par exemple.

L'évolution de notre infrastructure de calcul est un réel défi. Face au déluge de données issues de nos expériences, nous devons être proactifs, en particulier en continuant à travailler à

stratégie solide à moyen terme. La grille a été extrêmement efficace mais nous allons faire face à des défis énormes avec les prochains runs du LHC en terme de volumes de données et de besoins de calcul. Dans le même temps, les projets d'astroparticules vont commencer à produire des quantités significatives de données. Nous devons trouver un moyen de fournir le soutien nécessaire aux projets dans un budget contraint. Il sera alors peut-être nécessaire de penser à des solutions incluant par exemple l'utilisation de clouds commerciaux, d'infrastructures HPC locales et des approches hybrides. Cette discussion est également en lien avec le futur European Open Science Cloud (EOSC).

Enfin, je suis impatient de travailler avec France Grilles, LCG, le CC-IN2P3 et tous les autres partenaires impliqués afin de fournir la meilleure infrastructure de calcul pour la science de l'IN2P3 !

**- Comment envisagez-vous la recherche en informatique à l'IN2P3 ? Comment serait-il possible de faire "émerger" ces projets et sur quels sujets, selon vous ?**

Nous avons de très intéressants projets en cours dans ce domaine ! Certaines recherches mènent directement à des publications dans des revues en informatique. Nous voudrions identifier les équipes, aussi bien les chercheurs que les ingénieurs, dans les laboratoires qui pourraient constituer des équipes Calcul dédiées. Ils pourront alors proposer des projets soit existants, soit nouveaux pour obtenir un soutien direct de l'IN2P3 de la même manière que les projets expérimentaux. Pour toute la partie de l'informatique qui vient en soutien aux expériences, nous souhaiterions renforcer les activités transverses au sein de l'Institut mais aussi en direction d'autres partenaires. Nous avons déjà de bonnes initiatives qui vont dans ce sens, telles que la HEP Software Foundation, le RI3 et le CCRI. Mais nous souhaiterions former de plus petites équipes collaborant sur des sujets tels que le Machine Learning, le Big Data, le HPC et autres sujets émergents. Si cela s'avère utile, nous pourrions former des groupes de recherche GdR sur des sujets de recherche spécifiques.

**- Selon vous, que faut-il retenir du colloque calcul de juin 2015 ? Que pensez-vous par exemple de l'idée évoquée à cette occasion, de promouvoir la mutualisation autour de projets ou thèmes forts, d'assurer la cohérence et de mettre en perspective nos contributions aux différents projets européens ?**

Nous devons certainement travailler en collaboration plus étroite au sein de l'Institut sur certaines activités transverses. De plus, dans notre domaine, les financements dépendent de plus en plus de notre capacité à répondre à des appels à projets européens, à travailler au-delà des frontières et à prendre les responsabilités dans de grandes collaborations, comme par exemple l'EOSC. Quasiment toutes nos expériences sont internationales, c'est également le cas pour le calcul. Toutefois, chacun sait combien il est difficile d'identifier les appels Horizon 2020 les mieux adaptés, d'organiser une équipe internationale et d'aboutir à des réponses cohérentes, fortes et concises. Nous devrions pour cela faire équipe car le poids de la réussite à un appel H2020 peut être trop lourd pour un seul laboratoire IN2P3. Nous pouvons également tous profiter de nos expériences passées. J'estime qu'il est de la responsabilité de la direction de l'IN2P3 d'identifier les opportunités auxquelles nous devrions répondre et d'animer la collaboration, comme le fait Ursula Bässler pour l'EOSC.

**- Il existe à l'IN2P3 un écosystème autour du calcul et des données qui assure "l'informatique au quotidien", l'administration systèmes et réseaux, le bon**

l'évolution de la grille et du CC-IN2P3. Il faut également discuter de la concentration optimale de l'infrastructure de calcul, par exemple en termes de mésocentres dédiés à la NPAC et supportés par l'IN2P3.

**- Quelles priorités vous fixez-vous ?**

Ma première priorité est de déterminer qui est engagé dans la recherche en calcul appliqué au sein de l'IN2P3. Nous souhaitons soutenir ces activités dans ce domaine en identifiant des équipes et des projets dans les laboratoires de l'IN2P3. Malheureusement, afin de pouvoir prendre en compte ces projets dès l'an prochain, nous allons devoir les identifier et les soumettre très vite. Or j'ai pu visiter certains laboratoires depuis ma prise de fonctions il y a deux mois mais pas tous. Il y a une certaine activité en recherche sur le calcul appliqué à l'IN2P3 mais il serait bien d'accompagner non seulement celle-ci mais également d'encourager des collaborations au sein de l'IN2P3 et au-delà.

Un autre point urgent est l'évolution de l'infrastructure de calcul de l'IN2P3. Concernant la grille, nous avons décidé de renouveler la convention actuelle pour une année seulement et de la mettre à profit afin d'avoir le temps de décider d'une

**fonctionnement des infrastructures de calcul mais aussi et surtout, le support aux utilisateurs présents dans les laboratoires. Quelle est la stratégie de la nouvelle direction dans ce domaine ?**

On ne change pas une équipe qui gagne ! ;-)) En même temps, ce serait intéressant d'avoir une meilleure vision de qui fait quoi dans les laboratoires, et notamment dans les projets. Il n'est pas toujours clair, du moins pour moi, qui est en charge de calcul pour un projet dans chacun des laboratoires impliqués. Centraliser l'information nous permettrait de contacter plus facilement les collègues afin de distribuer une information spécifique ou de réagir à un problème particulier. Je voudrais signaler, qu'au sein de la direction de l'IN2P3, l'interlocuteur des services informatiques des laboratoires, comme pour les autres services techniques, est le directeur adjoint technique (DAT), Christian Olivetto. Mais il y a évidemment un recouvrement avec les aspects scientifiques et donc avec mon rôle de directeur adjoint scientifique Calcul et Données. Je suis donc curieux de participer aux réunions du CCRI et évidemment aux Journées Informatique.

**Propos recueillis par Gaëlle SHIFRIN**



## Gestion des journaux et métriques au CC-IN2P3



### Contexte

Le projet COLOSS, visant à moderniser la plateforme de gestion de journaux [1] du CC-IN2P3, est entré progressivement en production depuis le début de l'année 2016. La plateforme gère tous les jours plus d'un milliard d'événements, dont 10% sont stockés sur disque pendant un an. La provenance des informations est très variée :

- Journaux du système d'exploitation
- Messages applicatifs
- Événements d'appareillages d'infrastructure (climatiseurs, onduleurs, etc.)
- Mesures de performance des serveurs (cpu, mémoire, réseau, etc.)
- Comptabilité des "jobs" exécutés sur la ferme de calcul

Le système permet aux administrateurs et exploitants de services du CC-IN2P3 d'injecter leurs données d'exploitation et de les consulter par la suite. Deux interfaces utilisateur sont mises à disposition à cet effet. La première permet l'analyse instantanée du flux d'événements via un système d'abonnement synchrone. Elle est utilisée notamment pour la remontée d'alertes et le suivi en temps réel des conséquences d'une intervention. Sa faible latence (au plus 200ms) est très utile pour un suivi haut niveau à la suite d'une mise à jour ou d'un changement en amont. La deuxième interface permet une analyse historique des événements, très utile pour les investigations post-mortem, la comptabilité, ou encore la fouille de données. Les deux interfaces sont disponibles à la fois sous forme d'IHM [2] et d'API [3], afin de permettre une intégration facilitée aux autres services.

### Architecture

L'architecture de COLOSS repose entièrement sur une suite de logiciels libres. Un résumé de leur fonction au sein de la pile est exposé aux paragraphes suivants.

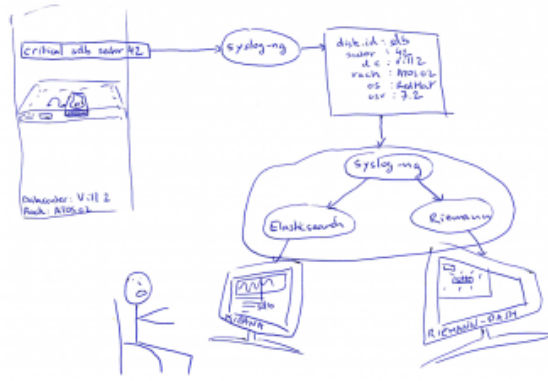
**syslog-ng** Le logiciel **syslog-ng** est un gestionnaire de journaux ("logs"). Édité par la société Balabit, son domaine de fonctionnalités dépasse largement le cadre que son nom indique. En effet, bien qu'initialement développé pour remplacer le service UNIX **syslogd**, il permet dans sa dernière version (3.7.3) de traiter des événements provenant d'un large éventail de sources différentes, et de les acheminer vers un jeu très diversifié de destinations. Il remplit au CC-IN2P3 toutes les fonctions vitales de gestion des "logs", notamment la réception, le filtrage, l'analyse syntaxique, l'enrichissement et le routage vers d'autres systèmes. Il permet par exemple d'indexer les documents dans **Elasticsearch** en utilisant le protocole natif (**node**). Un des grands avantages de **syslog-ng** par rapport à ses alternatives est son efficacité : écrit en langage C et utilisant des routines optimisées, il permet un passage à l'échelle vertical. En effet, une instance au CC-IN2P3 gère en moyenne 2000 événements par seconde avec une charge négligeable. Lors de pics de trafic, une simple instance traite presque 100 000 événements par seconde avec une charge de 50%. L'instance en question s'exécute sur une machine virtuelle avec

mini-programme. Il est utilisé au CC-IN2P3 pour effectuer des agrégations dynamiques, des seuillages de métrique et traiter les abonnements synchrones. Ce dernier point permet à un utilisateur de s'abonner à un flux d'événements grâce au protocole **websocket**, et ceci avec une latence très faible (quelques millisecondes). On peut le voir comme un **tail -f** centralisé pour les logs.

### La vie d'un événement dans COLOSS

Pour mieux comprendre la fonction de COLOSS, nous allons suivre la vie d'un message, depuis sa source jusqu'à l'utilisateur final. Prenons l'exemple du message suivant, qui est émis par un pilote du noyau **Linux** d'un serveur dans la nouvelle salle machine de Villeurbanne :

```
blk_update_request: critical target error, dev sdb, sector 42
```



L'événement est capturé par **syslog-ng** grâce à la directive de son fichier de configuration qui lui indique d'intercepter les "logs" système. Il est ensuite enrichi avec des informations propres au CC-IN2P3 (nom du "datacenter", position dans l'armoire, fonction du serveur, etc.). La suite comprend le passage dans un système d'analyse syntaxique de **syslog-ng (patterndb)** qui permet de classifier le message et de le structurer en extrayant des couples clé/valeur. Pour l'exemple précis, on extrait les valeurs concernant le nom du disque impacté (**sdb**) et le numéro du secteur (**42**). Pour l'heure, le message a évolué et ressemble à :

```
message:      critical target error, dev sdb, sector 42
disk.id:      sdb
disk.sector:  42
datacenter:   Villeurbanne 1
rack:         ATOS 02
osfamily:     RedHat
operatingsystem: CentOS
osversion:    7.2
```

Il est ensuite passé en revue par plusieurs filtres qui permettent de choisir le ou les chemins précis prévus pour cet événement particulier. En l'occurrence, il s'agit d'une erreur sérieuse de type matérielle, et l'événement est donc acheminé vers Nagios, qui est le système d'alerte et de notifications du CC-IN2P3. En outre, comme tous les événements, il est aussi aiguillé vers **Elasticsearch** pour le stockage et l'indexation long terme, et **Riemann** pour la partie faible-latence ("temps réel"). En dernière étape, **Riemann** achemine notre événement dans un navigateur exécuté sur une machine de la "control-room", et apparaît avec une couleur rouge (événement critique) sur un des trois grands écrans, bien visible par l'utilisateur final censé alerter le service concerné. L'événement sera aussi visible pendant un an lorsqu'une personne effectuera une recherche dans **Kibana**, pour autant qu'il corresponde aux critères de recherche.

### La suite

8 vCPU et 8G de RAM. En outre, l'architecture COLOSS permet d'orchestrer plusieurs instances *syslog-ng* grâce au logiciel **puppet**, permettant ainsi un passage à l'échelle horizontale de l'infrastructure.

**collectd collectd** est un système de collection et de traitement de métriques. Il est traditionnellement utilisé pour surveiller les compteurs de performance des systèmes d'exploitation de type UNIX. Il permet notamment de suivre l'évolution dans le temps du remplissage des systèmes de fichier, ou encore de mesurer le taux d'utilisation des processeurs. Son architecture est basée sur un noyau logiciel minimaliste entouré de "plugins", ce qui lui permet d'avoir une empreinte mémoire et "cpu" très réduite. Par conséquent, il convient parfaitement à la fois aux systèmes embarqués et aux grands systèmes, dans lesquels l'on peut activer un grand nombre de "plugins". Le CC-IN2P3 utilise *collectd* pour interroger les compteurs système (cpu, mémoire, disque, réseau), matériels (températures, puissances électriques) et applicatifs. Ceux-ci sont ensuite envoyés à *Riemann* pour le traitement centralisé. On peut configurer des seuils afin d'attacher un état à chaque événement ("ok", "warning" ou "critical"), ce qui permet de générer des notifications.

**Elasticsearch Elasticsearch** est un moteur de recherche distribué, qui permet de stocker et indexer des documents JSON. Il permet notamment des recherches de type "full-text", qui sont particulièrement appréciées par les utilisateurs. Son API très riche permet à la fois d'administrer le "cluster" et d'effectuer des requêtes précises sur le contenu. Un grand nombre de clients est disponible dans la plupart des langages de programmation courants, et sa popularité vient en grande partie de l'excellente interface graphique permettant de consulter les données qu'il contient : **Kibana**. *Elasticsearch* est utilisé au CC-IN2P3 pour stocker la totalité des "logs" pendant un an, ce qui représente près de 60 milliards de documents. Il sera également utilisé pour stocker les métriques dans le cadre du projet *sampler*, dont le but est de remplacer le système actuel basé sur **RRDTool**.

**Riemann** Le gestionnaire de flux *Riemann* est sans doute le moins connu de la bande. C'est un logiciel écrit en **Clojure** qui permet une gestion extrêmement flexible du flux d'événements qui le traverse du fait que son fichier de configuration est un vrai

Bien que la nouvelle plateforme de gestion des logs permette le passage à l'échelle en multipliant le nombre de serveurs, il reste la problématique de gestion du contenu. En effet, la quantité de données ne cesse de grandir avec les infrastructures et les projets des chercheurs pour lesquels elles sont mises à disposition. L'analyse syntaxique est statique, dans la mesure où les différentes règles d'analyse sont confectionnées par des humains. Et c'est bien cette partie qui ne grandit pas aussi vite que le reste : le personnel du CC-IN2P3 ! Par conséquent, il est urgent de trouver une solution pour automatiser ce processus. Nous réfléchissons actuellement à un système d'apprentissage automatique qui permettrait d'aller dans ce sens en mettant en évidence automatiquement des tendances dans l'évolution des données, et aussi en prédisant l'état des services : la météo du CC-IN2P3 ! "Demain, il fera beau en salle machine et peu de pannes se manifesteront en salle. Cependant, après-demain il risque d'y avoir une panne du service X avec une probabilité de 43%". Une collaboration avec d'autres sites est prévue dans ce sens, dont le **GSI** de Darmstadt, qui a déjà effectué des **recherches dans le domaine**.

#### **Publications**

La phase de développement du projet a été jalonnée par plusieurs présentations lors de conférences internationales (IEEE HPCMASPA 2014, Open Academy October 2014, HEPiX Spring 2016) et d'événements nationaux ou encore locaux (Elasticsearch Lyon User Group Mars 2015, AG Aramis Avril 2016, Clojure User Group Juin 2016). Les articles suivants ont été publiés et concernent chacun un aspect précis de l'architecture :

- \*syslog-ng, riemann, collectd-notifications, elasticsearch : putting it all together, **devops.com, 2014-09-02**
- Processing log messages with Python in syslog-ng, **Balabit guest blog, 2015-09-03**
- Transferring Conserver Logs to Elasticsearch, **Linux Journal, 2016-03-14**

**Fabien WERNLI (CC-IN2P3)**

[1] "logs" système et applicatifs

[2] Interface Homme Machine

[3] Interface de Programmation Applicative



## Introduction à Go & Retours d'expérience



Il n'y a plus de "Free Lunch" possible : la loi de Moore [1] n'est plus aussi aisée à mettre en œuvre que par le passé. Les développeurs de logiciels, ainsi que les scientifiques, doivent maintenant se familiariser avec la loi d'Amdahl [2]. En effet, la fréquence d'horloge des processeurs n'augmente plus avec chaque nouvelle génération : les processeurs gagnent de plus en plus de cœurs, mais sans changer la fréquence par rapport à la génération précédente.

L'augmentation du nombre de cœurs appelle à une utilisation plus répandue du parallélisme dans les logiciels développés par les communautés HEP et Astro. Traditionnellement, le parallélisme a été exploité au niveau du traitement des événements dans les logiciels de reconstruction et de simulation des expériences de physique des particules : il suffit de lancer  $N$  jobs sur les fermes de calcul, la Grille ou un *cloud*. Plusieurs événements peuvent être traités en parallèle, même si les algorithmes de reconstruction sont toujours appliqués de manière séquentielle à chacun de ces événements.

Cependant, chaque nœud de calcul hébergeant chacun de ces programmes met à disposition des ressources finies (empreinte mémoire, CPU, descripteurs de fichiers, *sockets*, E/S, etc), ressources dont le passage à l'échelle est plus problématique lorsque l'empreinte mémoire des jobs de reconstruction des expériences du LHC atteint 2 à 4 Go de RAM. Cette exploitation du parallélisme atteint ses limites avec les ordres de grandeur du LHC.

Ainsi, il semble plus efficace d'effectuer le traitement parallèle des événements dans le même espace mémoire. Pour un programme écrit en C++, cela se traduit par l'utilisation de plusieurs *threads*. Malheureusement, la programmation parallèle multitâche (*multithreading*) est notoirement connue pour ses difficultés de mise en œuvre :

- il est très ardu d'écrire correctement une application *multithreadée*,
- il est également difficile de la garder correcte en fonction du temps,
- et aussi ardu de la garder efficace et optimisée au cours des développements successifs.

De plus, même si C++11 [3] et C++14 [4] apportent enfin la standardisation tant attendue des APIs de programmation parallèle (`std::thread`, `std::mutex`, `std::future`), cela se fait au prix d'une complexification encore plus poussée de ce langage, sans toutefois exposer une API de haut niveau. Il semble donc judicieux de se demander s'il n'existerait pas un nouveau langage de programmation mieux adapté à l'exploitation des architectures parallèles...

Pourquoi pas Go [5] ?

### Anatomie de Go

Depuis la publication de la version 1.0 de Go en 2012, le langage est considéré comme stable et complètement rétrocompatible : chaque nouvelle version de Go (une tous les six mois en moyenne) compilera correctement un programme valide de 2012. Ce contrat de stabilité est également appliqué à la bibliothèque standard livrée avec le compilateur.

De par son origine et son ADN, Go permet de développer rapidement des programmes d'envergure. Son modèle de compilation d'exécutables statiques permet également de les déployer aisément sur de grandes infrastructures : Go est rapidement devenu la *lingua franca* du développement *cloud*. Est-il adapté à l'environnement HEP et astro ?

### Retours d'expérience

#### Analyse et simulation

Go a fait ses débuts dans HEP *via* la réimplémentation du *framework* de contrôle hors-ligne d'ATLAS et LHCb : Gaudi [10]. Cette réimplémentation, appelée simplement *fwk*, est regroupée sous l'ombrelle de l'organisation *go-hep* [11]. Le but principal de *go-hep/fwkc* [12] était de démontrer la viabilité et l'adéquation de Go pour les *control frameworks* LHC, mais également de montrer l'aisance avec laquelle la programmation parallèle peut être mise en œuvre avec Go.

Un autre axe de travail était de montrer qu'un cadriciel concurrent comme *fwk* était non seulement adapté aux grosses expériences LHC avec leur infrastructure sous-jacente, mais était également adapté aux applications de plus petites tailles telles qu'une analyse individuelle ou bien une bibliothèque de simulation. En effet, une des critiques récurrentes des physiciens vis-à-vis de Gaudi est sa lourdeur d'implémentation ainsi que la difficulté de l'installer sur une machine de bureau, ce qui en fait une plateforme de développement d'analyses peu séduisante, malgré sa robustesse et sa capacité à traiter les volumes de données du LHC.

Cet axe de travail a été concrétisé sous la forme de *fads* [13]. *fads* est la réimplémentation de Delphes [14], un simulateur de détecteur de physique des particules. Delphes est un ensemble de composants C++ basés sur Root, dont le *design* ne se prête pas aisément à une implémentation *multithreadée*. Les détails de la comparaison entre les deux applications sont disponibles dans [15] : le passage à l'échelle de *fads* est nettement meilleur que Delphes, tant en empreinte mémoire qu'en fréquence de traitement d'événements. Ces performances sont reportées dans la figure 4.

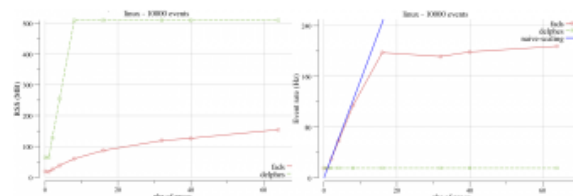


Figure 4 : Empreinte mémoire (à gauche) et fréquence de traitement des événements (à droite). Résultats obtenus sur un serveur Linux de 40 cœurs (20 physiques). A noter que, passés 8 processus concurrents, le serveur n'avait plus assez de ressources (RAM) pour Delphes.

Au cours du développement de *fads*, plusieurs bibliothèques d'interfaçage avec l'existant (Root, HepMC, HEPEVT) ont dû être développées, ainsi que des bibliothèques d'analyse (histogrammes, plots). Ce travail a permis de montrer que Go est un langage viable et compétitif dans le cadre du calcul et de l'analyse de données, et ce, même dans le microcosme HEP.



Go est un langage de programmation, libre (licence BSD-3), initialement développé par Google et annoncé au monde en novembre 2009. C'est un langage compilé, avec gestion automatique de la mémoire *via* un ramasse-miettes (*garbage collector*), le support de la réflexion de type, des fonctions anonymes, des *closures* [6] et de la programmation orientée objet. La syntaxe de Go est une réminiscence de celle du C. Un bref aperçu d'un premier programme en Go est donné dans la figure 1.

```

1 // Command hello provides a greeting in french.
2 package main
3
4 import (
5     "fmt"
6 )
7
8 func main() {
9     fmt.Println("Bonjour la Lettre IN2P3.")
10 }

```

Figure 1 : Un premier programme en Go.

Go est apprécié pour sa capacité à apporter au développeur, ainsi qu'à l'utilisateur final, le meilleur des mondes "dynamique" et "statique" :

- l'aisance de programmation des langages dynamiques, grâce à sa verbosité limitée, son système d'inférence de type et un cycle de développement edit-compile-run extrêmement rapide,
- la robustesse d'un système de typage statique,
- la vitesse d'un exécutable compilé en langage machine.

De plus, le support de Go pour les interfaces ressemble fortement au *duck typing* de Python [7], et se prête très bien à l'écriture de vastes cadres (*frameworks*), tels que ceux utilisés et développés pour les expériences au LHC.

Enfin, Go expose un support accru, et intégré au langage, de la programmation concurrente *via* le modèle CSP (*Communicating Sequential Processes* [8]). En effet, il suffit de préfixer l'appel à une méthode avec le mot-clé `go` pour que son exécution s'effectue de manière concurrente aux autres fonctions, et ce, dans un *thread* léger appelé *goroutine*. Les *goroutines* sont multiplexées sur plusieurs *threads* natifs : une *goroutine* bloquée, par exemple, par une opération d'E/S (disque, réseau, etc...), n'empêchera pas l'exécution des autres *goroutines* du programme. De plus, les *goroutines* sont très peu gourmandes en ressources, grâce à leur pile de petite taille et ajustée à l'exécution. Ceci permet d'en lancer un très grand nombre, et ce sur des machines très ordinaires : il n'est pas rare de voir des programmes comportant des milliers de *goroutines* sur des laptops aux capacités modestes, prouesse impossible avec des *threads* natifs.

Dans le langage Go, le mécanisme permettant d'échanger des données entre *goroutines*, et ce sans remettre en cause la sûreté du système de typage (*type safety*), est appelé un *channel*. Envoyer et recevoir des données par un *channel* sont des opérations atomiques et permettent donc de synchroniser l'exécution des *goroutines*. Un exemple de communication entre *goroutines* est donné dans la figure 2 :

— la *goroutine* principale, `main`, lance deux *goroutines*, `gen` et `square`. — `gen` génère la suite des nombres entiers de 0 à  $+\infty$  et remplit le *channel* `in`. — `square` extrait les nombres entiers de ce *channel et remplit le *channel* `out` avec le carré de ces nombres. La *goroutine* `main` extrait les données du *channel* `out` et les affiche à l'écran.*

Go permet également d'appeler aisément du C *via* `cgo` [9]. Il suffit d'importer le pseudo package "C" et d'indiquer les fichiers d'en-tête et bibliothèques pour utiliser les fonctions et types exposés par cette bibliothèque C. Un exemple est donné dans la figure 3.

## Contrôle commande & monitoring

Un axe de recherche plus récent est l'investigation de Go et sa pertinence dans le monde du contrôle commande. Dans le cadre de l'expérience LSST, un banc de test pour la caractérisation des moteurs pour le changeur de filtres de la caméra devait être réalisé. Ces moteurs peuvent être commandés *via* plusieurs interfaces et protocoles (CANBus, ModBus, ...) : c'est le protocole ModBus qui a été finalement retenu et mis en œuvre.

Malgré la jeunesse de Go, une bibliothèque prenant en charge le protocole ModBus était déjà disponible sous licence libre, distribuée *via* github [16] et écrite par la communauté. Comme pour tous les packages Go, une simple commande a suffi pour installer cette bibliothèque :

```
$> go get github.com/goburrow/modbus
```

L'application permettant d'envoyer des commandes aux moteurs, de *monitorer* leurs positions, températures et autres grandeurs est en fait un serveur web écrit en Go. En effet, l'offre des GUIs en Go est encore limitée : même s'il existe des *bindings* vers la plupart des bibliothèques graphiques portables (Qt, GTK, ...), leur installation n'est pas aussi aisée qu'un package écrit totalement en Go. Il existe bien le début d'une bibliothèque en Go, mais elle est encore en chantier [17]. La solution retenue a donc été l'écriture d'un exécutable servant une page web HTML5 utilisant Polymer [18] pour réaliser l'interface graphique. L'utilisateur envoie des commandes depuis l'interface, commandes qui sont ensuite relayées au serveur *via* des WebSockets. Le serveur se charge de la communication avec les moteurs et renvoie résultats des commandes, histogrammes, flux vidéo et grandeurs *monitorées* à l'utilisateur au moyen d'un autre WebSocket. L'authentification des utilisateurs, ainsi que la syndicalisation des flux et connexions, sont gérées côté serveur. Cette architecture orientée web permet *in fine* une grande transparence réseau, et ce, même pour des clients Windows. La figure 5 constitue un aperçu de l'interface graphique.

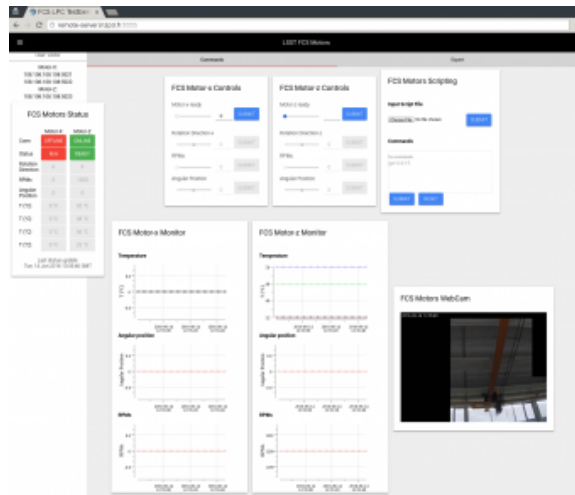


Figure 5 : Interface graphique pour le banc de test du LPC. L'interface est en HTML5/Polymer, le serveur en Go.

## Conclusions

Cet article a présenté le langage Go. Son approche pragmatique et sa volonté de rester simple (mais pas simpliste), couplées à son modèle de programmation concurrente, font de Go un langage résolument adapté à l'environnement plus hétérogène des machines multicœurs d'aujourd'hui et de demain.

Malgré son relatif jeune âge, Go comporte déjà la plupart des bibliothèques nécessaires à la programmation d'applications de calcul et d'analyse. Les outils intégrés à la chaîne de développement de Go permettent, de plus, de rapidement optimiser un code donné (CPU, mémoire, concurrence, I/O, ...). Enfin, l'empreinte mémoire réduite de Go par rapport à Java et ses facilités en programmation concurrente en font une alternative crédible dans le domaine de l'acquisition et *monitoring* de données, et le contrôle commande.

Go est d'ores et déjà le langage du cloud. Peut-être aura-t-il une vie dans HEP et en Astro ? Une chose est sûre : il a tous les atouts pour y arriver et supplanter C++, Python et Java.

Sébastien BINET (LPC)

```
1 package main
2
3 func main() {
4     in := make(chan int)
5     out := make(chan int)
6     go gen(in)
7     go square(out, in)
8     for i := range out {
9         println(i)
10    }
11 }
12
13 // gen generates numbers [0...]
14 func gen(ch chan int) {
15     i := -1
16     for {
17         i++
18         ch <- i
19     }
20 }
21
22 // square returns the square of each number.
23 func square(out, in chan int) {
24     for i := range in {
25         out <- i+i
26     }
27 }
```

Figure 2 : Deux goroutines, gen et square se communiquent des données.

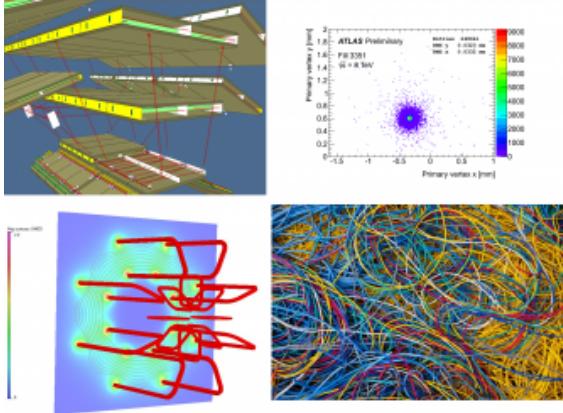
```
1 package main
2
3 // #cgo LDFLAGS: -lm
4 // #include <math.h>
5 import "C"
6
7 func main() {
8     println("C.sqrt(4)=", C.sqrt(4))
9 }
```

Figure 3 : Programme Go utilisant la fonction sqrt de la bibliothèque Clibm.

- [1] Moore E, "Cramming more components onto integrated circuits", Electronics Magazine, 1965
- [2] Amdahl G, "Validity of the Single Processor Approach to Achieving LargeScale Computing Capabilities", AFIPS Conference Proceedings, (30), pp. 483-485, 1967
- [3] The C++11 programming language, <http://www.open-std.org/jtc1/sc22/w...>
- [4] The C++14 programming language, <http://www.open-std.org/jtc1/sc22/w...>
- [5] Go, <https://golang.org>
- [6] Les closures sont des fonctions qui capturent des références à des variables libres de l'environnement.
- [7] The Python programming language, <http://python.org>
- [8] CSP, <http://en.wikipedia.org/wiki/Commun...>
- [9] cgo, <https://golang.org/cmd/cgo/>
- [10] Mato P 1998 Gaudi-architecture design document Tech. Rep. LHCb-98-064 Geneva
- [11] The go-hep project, <https://github.com/go-hep>
- [12] The go-hep/fwk concurrent control framework, <https://github.com/go-hep/fwk>
- [13] fads : a Fast Detector Simulation toolkit, <https://github.com/go-hep/fads>
- [14] DELPHES 3 : a modular framework for fast simulation of a generic collider experiment, arXiv:1307.6346
- [15] fads @ HEP Software Foundation workshop, <https://indico.cern.ch/event/357737...>
- [16] <https://github.com/goburrow/modbus>
- [17] <https://github.com/golang/exp/tree/...>
- [18] <https://www.polymer-project.org/>



## ATLAS, CMS et les « Conditions Data »



ATLAS et CMS, expériences phare au CERN, ont lancé un projet commun pour le développement d'une infrastructure de gestion de données de type Conditions de nouvelle génération. Au cœur de cette infrastructure, la « Conditions DB » est une base de données qui enregistre en continu l'état du détecteur et de tous les appareillages de l'expérience (déclenchement, alignement...) pendant la prise de données. Ces Conditions Data doivent ensuite pouvoir être lues de façon très efficace et en parallèle par les dizaines de milliers de jobs simultanés de la phase de reconstruction des événements, puis des programmes d'analyse un peu partout dans le monde.

Les expériences LHC ont terminé en 2013 la première phase de prise de données (Run1). Celle-ci a permis entre autres la découverte du boson de Higgs. Pour obtenir ce résultat, les expériences ATLAS et CMS ont analysé dans un environnement de calcul distribué des centaines de Pétaoctets de données de Physique (ou *Event Data*). En complément, d'autres données, que nous appelons *Conditions Data* (faute de trouver un terme adéquat en français), ont été nécessaires à la reconstruction et l'analyse des événements de physique. Celles-ci caractérisent les conditions dans lesquelles les *Event Data* ont été enregistrées. A la différence des *Event Data*, qui contiennent la réponse des détecteurs au passage des particules, les *Conditions Data* contiennent une grande variété d'informations provenant, par exemple : des calibrations des détecteurs, du contrôle des alignements, des mesures liées au faisceau du LHC telles que la luminosité, ou encore de certaines configurations utilisées par les différents détecteurs au cours de la prise de données. Dans le cadre de l'expérience ATLAS, cela représente environ 1,5 Téraoctet de données accumulées pendant le Run1.

Aujourd'hui, les systèmes de gestion des *Conditions Data* d'ATLAS et CMS diffèrent : le système adopté par ATLAS est basé sur COOL, la solution développée au CERN dans le cadre de LCG [7], et celui de CMS exclusivement sur la couche logicielle C++ CORAL (Common Relational Abstraction Layer). Mais les deux expériences font appel à la même solution de cache *Frontier/squid* pour améliorer les performances d'accès à la base de données, et utilisent des concepts très proches pour la définition des métadonnées. Ceci a permis aux équipes chargées de la mise en œuvre de ces systèmes d'entamer une réflexion sur une approche commune en vue du Run3 [2], qui puisse éventuellement être réutilisée dans le cadre d'autres expériences de physique. Pour l'instant, les équipes concernées par ce développement sont petites. En France, l'Irfu au CEA et le LAL à Orsay sont impliqués.

La définition de la nouvelle architecture de gestion des *Conditions Data* s'articule autour de trois éléments :

format binaire sauvegardé comme une seule entité, ce qui signifie que l'infrastructure proposée ne peut effectuer aucune requête sur le contenu des *Conditions Data* proprement dites. Ainsi, la base de données est agnostique par rapport au contenu des *Conditions Data*, et même au format binaire utilisé par le client. Ceci permet de conserver une structure de base de données simple, en dépit de la variété des informations enregistrées et surtout de la complexité et de l'évolution des besoins de l'expérience. Ce choix rend également possible l'exploration de nouvelles solutions basées sur des technologies NoSQL par exemple.

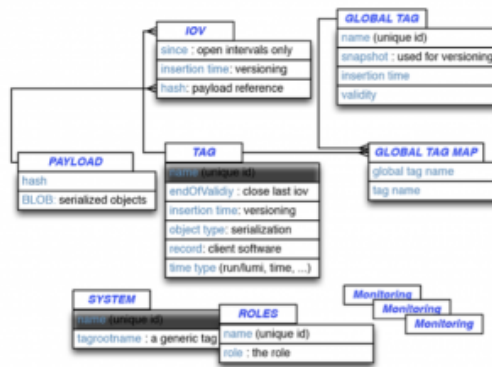


Figure 1. Structure des données pour la nouvelle architecture

Les concepts principaux utilisés pour caractériser les *Conditions Data* sont les suivants :

- *Payload* : Ce terme représente la donnée utile, à un moment précis (par exemple un paramètre de correction pour la géométrie du détecteur à muons). Chaque *Payload* est associé à une information en temps ou à un *run number*.
- *iov (Interval Of Validity)* : Une caractéristique des *Conditions data* est de varier (lentement) dans le temps. L'*iov* représente l'intervalle de validité pour un *Payload* spécifique.
- *Tag* : Il regroupe de manière univoque l'historique d'une calibration, c'est-à-dire l'ensemble des *iovs* pour des grandeurs (efficacités, corrections ou autres) qui résultent d'une même version d'un algorithme de calcul.
- *Global Tag* : Il permet d'identifier un ensemble cohérent de *Tags* réunissant toutes les données utiles (*Payload*) pour analyser les *Event Data*.

La Figure 2 montre les composants principaux de la nouvelle architecture. L'approche *multi-tier* constitue un élément fondamental pour l'environnement du calcul distribué dans lequel ces données sont utilisées. La mise en place d'une interface REST est aussi très importante, à la fois pour l'aspect multilingage qu'elle supporte de manière naturelle, mais aussi pour la possibilité d'utiliser des systèmes de cache de type WEB éprouvés, provenant de la communauté Open Source (comme Squid).

La première implémentation de l'application côté serveur utilise des technologies de type Java, avec l'utilisation des standards promus par les spécifications JEE notamment en termes de services REST (JAX-RS) ainsi que de persistance (JPA). Ceci garantit par exemple la portabilité de l'implémentation sur différentes technologies de back-end (comme des bases de données non relationnelles). Les données échangées entre les clients et le serveur sont de simples structures JSON ou XML.

- La structure de la base de données : Il s'agit de bien définir les données descriptives (ou métadonnées) des quantités que l'on veut enregistrer, pour garantir la robustesse de la gestion de l'historique et des versions de *Conditions Data*.
- Le logiciel d'interaction avec la base de données : Pour isoler les clients de la complexité liée à la couche de persistance, nous avons décidé pour ce projet de regrouper toutes les fonctionnalités de gestion des *Conditions Data* au sein d'un serveur intermédiaire (suivant un classique modèle multi-tier, comme dans la plupart des services WEB). Ce serveur expose une partie de ces fonctionnalités à travers des services WEB de type REST.
- Les bibliothèques clientes : celles-ci doivent tout simplement permettre d'interagir avec le middle-tier, en utilisant le langage de son choix (Python, C++, ...). Parmi ces clients, nous prévoyons aussi le développement d'une interface WEB générique pour la navigation dans les métadonnées (et d'un client Javascript).

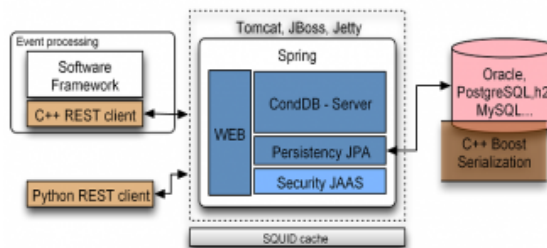


Figure 2. Prototype pour la nouvelle architecture de gestion des *Conditions Data*

En conclusion, ce projet se propose d'explorer une solution simple, performante et générique pour la gestion des *Conditions Data* d'une expérience de physique. Les choix effectués en termes d'architecture et de modèle de données prennent en compte l'expérience accumulée dans ce domaine par ATLAS et CMS au cours des Run1 et 2 du LHC. Le projet suscite un vif intérêt au sein de la communauté HEP, et il est répertorié (parmi d'autres qui explorent des solutions différentes pour le même problème) dans les pages de la HSF (*Hep Software Foundation*).

Andrea FORMICA (IRFU)

Le modèle de données se décline dans un nombre très limité de tables (cf. Figure1), qui contiennent tous les éléments nécessaires aux *data-flows* impliquant les *Conditions Data*, et aux différents cas d'utilisation : traitement *online*, première reconstruction rapide au CERN, campagne de *reprocessing* etc. Il est important de souligner que les données utiles sont agrégées, sérialisées et enregistrées dans la base sous forme de BLOB (Binary Large Object), ensemble de données au

[1] Valassi A, Basset R, Clemencic M, Pucciani G, Schmidt S A and Wache M 2008 IEEE Nuclear Science Symposium Conference Record, 2008. NSS '08 pp 3021–3028 URL <http://ieeexplore.ieee.org/xpls/icp...>

[2] Barberis D, Formica A, Gallas E, Govi G, Miotto G L and Pfeiffer A 2015 Journal of Physics : Conference Series vol 664 (IOP Publishing) p 042015 URL <http://iopscience.iop.org/article/1...>



n°34  
Juillet  
2016

# La lettre IN2P3 Informatique

Réseau des Informaticiens de l'IN2P3 et de l'IRFU



## « CYCLONE » : Gestion dynamique d'applications multi-cloud



Les infrastructures de « cloud » de type « Infrastructure as a Service » (*IaaS*) s'adressent aux développeurs et aux applications métiers. Elles connaissent un grand succès grâce à leur flexibilité permettant l'utilisation de ressources à la demande. Mais les utilisateurs, et les développeurs eux-mêmes, peuvent parfois avoir du mal à maîtriser la gestion d'applications et de services sur une telle infrastructure, dont les outils sont traditionnellement entre les mains des administrateurs « système et réseau ».

Ce problème devient d'autant plus ardu que les applications et leurs déploiements se complexifient, se diversifient, en nécessitant, aujourd'hui, l'utilisation non plus d'un seul cloud, mais plusieurs clouds simultanément. Ce choix du « *multi cloud* » est motivé par un souci de proposer une meilleure résilience, de meilleurs temps de réponse, des politiques spécifiques de placement ou d'ordonnement de calcul ou de données, ou encore, de réduire les coûts.

Le projet européen **CYCLONE**, démarré en janvier 2015 pour une durée de trois ans, réunit sept partenaires de six pays. Il a pour objectif de développer les services et les protocoles nécessaires au développement et au déploiement d'applications complexes dans un environnement multi-cloud. Ces services et protocoles concernent l'orchestration, le déploiement, la gestion, la sécurité (authentification, droits, cryptage des communications, cycle de vie des données etc.), mais aussi le réseau (*VPN*, *firewall*, *load balancing*...) et permettront d'abstraire certains aspects techniques généralement peu connus des développeurs et des utilisateurs.

Le projet réutilise, autant que faire se peut, les services et les protocoles existants. Il livrera des logiciels et des services de

production dont la qualité sera attestée par leur mise en œuvre dans des cas concrets : les applications de l'Institut Français de Bio-informatique (*IFB*), d'une part, et les applications de redistribution énergétique de *QSC*, une entreprise basée à Berlin (Allemagne), d'autre part.

Le LAL a pris deux responsabilités dans ce projet :

- D'une part, dans le développement logiciel autour de la sécurité et du réseau pour lequel deux ingénieurs (un permanent et un CDD) travaillent à plein temps. Ces travaux permettent le déploiement d'une fédération d'identités, la mise en place de VPN et de firewall « *as a Service* » ainsi que le cryptage des communications point à point. L'innovation réside dans la mise en place et l'utilisation de ces technologies sur un ensemble de clouds.
- D'autre part, dans le déploiement et la maintenance d'une plate-forme de cloud « *IaaS* », activités pour lesquelles des ingénieurs encadrent un apprenti. La plate-forme fournie par le LAL est une composante du « *testbed* » du projet qui inclut aussi les clouds fournis par *QSC*, à Berlin, et par le fournisseur *InterRoute*, à Rome (Italie). Ce *testbed* permet aux développeurs et aux primo utilisateurs de tester les services, les protocoles et les applications complexes dans un environnement multi cloud.

Le cloud du LAL s'appuie sur le middleware « **OpenStack** ». Il est aujourd'hui composé d'une soixantaine d'hyperviseurs pour mille deux cents cœurs environ ; il est appelé à s'agrandir dans le courant de l'année 2016.

Nous vous invitons à **nous contacter** si vous développez ou utilisez une application qui pourrait tirer profit d'un environnement multi cloud.

**Oleg LODYGENSKY (LAL)**



n°34  
Juillet  
2016

# La lettre IN2P3 Informatique

Réseau des Informaticiens de l'IN2P3 et de l'IRFU



## JI2016 : Ouverture des inscriptions !

Les prochaines Journées Informatiques de l'IN2P3 et de l'IRFU se tiendront du 26 au 29 septembre 2016 au Lioran en Auvergne (à 2h de Clermont).

Ces journées sont un lieu de rencontre de la communauté des informaticiens de l'IN2P3 et de l'IRFU. Elles abordent tous les aspects de la production de programmes, de la mise en œuvre et de l'utilisation de l'informatique au sein de la communauté de la recherche en physique des particules et astroparticules, en astrophysique et en physique nucléaire.

Cette année vous noterez un certain nombre de nouveautés qui ont pour but d'augmenter le côté participatif de ces journées :

- La montée en puissance des ateliers qui avaient été très appréciés en 2014 et qui seront au nombre de 7 cette année.
- L'introduction de créneaux pour des discussions par groupe de 10 personnes maximum, sur des sujets qui peuvent être de toute nature allant de la technique à l'organisationnel. Notez vos idées car vous serez sollicités pour proposer des sujets le premier jour des JIs.
- Dès le lundi, une immersion en "*piscine numérique*", où chacun pourra essayer un ou plusieurs "plongeurs" sur les technologies de son choix.

A ce titre, vous êtes tous invités à enfilez votre bonnet de maître-nageur et à nous proposer des mini-tutoriels et exercices courts (15-20 minutes), sur vos sujets de prédilection, pour aider vos collègues à se mettre à l'eau.

Vous pouvez consulter **le programme**, notez qu'il y aura encore probablement quelques ajustements à la marge.

**Inscriptions**



n°34  
Juillet  
2016

# La lettre IN2P3 Informatique

Réseau des Informaticiens de l'IN2P3 et de l'IRFU



## CDD IE : Administration des systèmes et réseaux informatiques - LAL, Orsay

Grand laboratoire du CNRS en physique des particules (300 personnes) situé à Orsay (91) recrute :

Poste en CDD de la fonction publique pour une durée de vingt-quatre mois (titre ou diplôme de niveau II : licence, licence professionnelle, master1, ...)

### Contexte :

Le Service Informatique du Laboratoire de l'Accélérateur Linéaire est en charge de la mise en œuvre des ressources informatiques du laboratoire et de développements logiciels dans le cadre des expériences ou projets, souvent internationaux. Le candidat sera intégré à l'équipe exploitation du service informatique qui, outre ses missions locales, participe fortement à l'infrastructure informatique de la discipline au niveau national (IN2P3-IRFU) et international (notamment CERN, grille EGI).

### Missions :

- Administration des serveurs informatiques Linux du laboratoire et gestion de leur évolution.
- Administration des ressources de stockage et gestion de leur évolution.
- Gestion des applications scientifiques et techniques disponibles au laboratoire.

### Activités :

- Mettre en oeuvre et faire évoluer l'ensemble des services Linux (services de fichiers, bases de données, Web, DNS, LDAP, ...)
- Mettre en place des moyens et procédures pour garantir les performances et la disponibilité des ressources liées aux services et au réseau.
- Participer activement au support utilisateur.
- Installer et maintenir les applications scientifiques et techniques
- Rédiger des documentations techniques à usage interne et externe
- Effectuer une veille technologique.

### Compétences :

- Connaissance approfondie de l'architecture et de l'administration Linux.
- Maîtrise des technologies réseaux
- Écriture de scripts shell
- Connaissance des technologies de stockage (NAS, iSCSI, NFS, CIFS)
- Capacité de dialogue avec les utilisateurs
- Initiative et travail en équipe
- Pratique courante de l'anglais technique.
- La connaissance de Python est un plus.

### Rémunération :

Salaire net mensuel, selon expérience : 1650€ à 2000€.

Merci d'envoyer CV et lettre de motivation par email à : [valerie.givaudan@lal.in2p3.fr](mailto:valerie.givaudan@lal.in2p3.fr)



## CDD : Développeur d'applications web - CC-IN2P3, Villeurbanne

Le Centre de Calcul de l'Institut national de physique nucléaire et de physique des particules (CC-IN2P3) est une unité de service et de recherche du CNRS. Classé parmi les grandes infrastructures françaises de recherche, il a pour vocation de fournir des moyens de calcul et de stockage de données aux chercheurs impliqués dans les expériences de physique corpusculaire. Plus de 4000 utilisateurs organisés en plus de 70 collaborations internationales utilisent ses services 24h/24, 7j/7.

### Plus d'informations

#### Fonctions

Le Centre de Calcul de l'IN2P3 (CC-IN2P3) est l'un des acteurs majeurs du projet européen EGI (European Grid Infrastructure). Financé par l'Union Européenne, et comprenant plus de 90 instituts repartis sur plus que 50 pays, ce projet a pour but d'intégrer le travail des projets Grille nationaux, régionaux de manière à créer une infrastructure Grille mondiale destinée à la communauté scientifique.

### Plus d'informations

Dans le contexte de ce projet, le CC-IN2P3 est en particulier responsable du portail des opérations du projet EGI [voir <http://operations-portal.egi.eu>]. Le candidat retenu aura pour mission de :

- Participer activement au développement du site web,
- En accord avec les autres membres de l'équipe, concevoir et présenter les documents décrivant les fonctionnalités du site,

- Assurer le suivi quotidien du fonctionnement de ce portail, en étroite collaboration avec d'autres instituts impliqués dans le projet.
- Assurer les développements du portail composants l'accord EGI-Engage

Le candidat sera intégré à l'équipe « Applications », composée d'environ 17 ingénieurs.

### **Qualifications requises**

#### *Formation*

De formation BAC + 3 ou universitaire équivalent (Maîtrise, Master, ...) en informatique ou en une discipline scientifique connexe.

#### *Compétences*

Le candidat aura des compétences dans les domaines ci-dessous :

- Connaissance approfondie des langages de conception de sites web, en particulier Javascript et PHP, HTML, CSS
- Connaissance pratique de Java,
- Savoir concevoir et utiliser des services web,
- Bonne connaissance des technologies autour de XML (XSL, XSLT),
- Bonne connaissance du développement orienté objet
- Savoir écrire et concevoir des documentations techniques

Les compétences suivantes seront appréciées sans être indispensables pour la candidature :

- Connaissance des technologies de sécurisation des services informatiques
- Connaissance du framework Symfony
- Connaissance d'outils de gestion de version (SVN, git)
- Connaissance des méthodes de développement agile

Outre les compétences techniques, le candidat doit :

- Avoir un bon relationnel, le sens de l'organisation et savoir travailler en équipe dans le cadre d'un projet international,
- Avoir des capacités d'adaptation, un haut degré d'autonomie et être réactif,
- Avoir une bonne aptitude à communiquer aussi bien à l'oral qu'à l'écrit
- Maîtriser l'anglais comme langue de travail,
- Avoir ou vouloir acquérir une bonne connaissance du français,
- Être rigoureux

### **Conditions contractuelles**

Le poste proposé est de durée limitée et directement lié au projet EGI. Un contrat type CNRS à durée déterminée de un an sera proposé pour un début effectif du travail dès que possible.

### **Dépôt des candidatures**

Envoyez votre candidature (CV et lettre de motivation) par courrier électronique (format PDF de préférence, MS Word ou texte également acceptés) à l'adresse **Job72@cc.in2p3.fr**.

Les candidatures sont acceptées jusqu'à l'attribution du poste. Les candidats retenus seront convoqués à un premier entretien dans les locaux du Centre de Calcul de l'IN2P3 à Lyon. Une partie de l'entretien sera tenue en anglais.





n°34  
Juillet  
2016

# La lettre IN2P3 Informatique

Réseau des Informaticiens de l'IN2P3 et de l'IRFU



## Arrêt du service Webmail Horde le 31/12/2016

Le Centre de Calcul propose depuis de nombreuses années un service Webmail basé sur le logiciel Horde pour accéder aux serveurs de messagerie des laboratoires de l'IN2P3 à travers un navigateur internet. Le **site web correspondant** sera arrêté définitivement le 31 décembre 2016. Aucune solution alternative ne sera proposée.

Les laboratoires qui pourraient encore utiliser ce site sont invités à déployer une solution locale. Les utilisateurs éventuels sont quant à eux invités à exporter leurs carnets d'adresses, agendas, tâches et notes en utilisant la fonction d'export.



## Calcul parallèle et GPGPU au Centre de Calcul

Cette année 2016 apporte deux grandes nouveautés en matière de calcul au CC-IN2P3 !

En février dernier, le Centre de Calcul lançait son nouveau cluster "High Performance Computing" (HPC) : 16 serveurs CentOS7 connectés via un réseau privé Infiniband QDR 40 Gb/s, embarquant chacun 32 coeurs cadencés à 2,3 GHz et 128 Go de mémoire, pour un total de 512 coeurs et 2 To de mémoire.

Cet été, un tout nouveau projet de calcul verra le jour. Le CC-IN2P3 s'équippa d'un cluster GPGPU, composé de 10 serveurs dotés chacun de 2 cartes NVIDIA Tesla K80, soit un total de 40 GPU NVIDIA GK210. Chaque GPU pourra disposer de 4 coeurs réels à 2,6 GHz, de 32 Go de RAM (12 Go de DDR5 par GPU sur la carte) et d'un stockage local performant (SSD) sur le serveur. Ces 10 serveurs seront connectés via un réseau privé InfiniBand QDR pour d'éventuelles applications multi-GPU.

■ **Informations pratiques sur l'environnement logiciel et l'utilisation du cluster HPC**

■ Pour les informations sur le cluster GPGPU, veuillez contacter **le support utilisateur**



## Les Webinaires, le retour

Depuis le 31 mars 2016, les Webinaires se sont installés à nouveau dans le paysage de l'informatique à l'IN2P3/IRFU, avec un format rénové !!

Un webinaire c'est quoi exactement ? Wikipedia et autres encyclopédies du Web nous disent que c'est un séminaire diffusé sur le web, certes, mais la cellule webinaire du RI3 souhaite avant tout que cela soit un outil d'échange entre les informaticiens. Dans un webinaire, on expose ses travaux, on découvre les travaux des autres, on s'enrichit des retours d'expérience de ses collègues.

Pour favoriser ces échanges, nous privilégions les webinaires avec plusieurs orateurs, si possible de laboratoires différents, ce qui permet de donner des points de vue sous des angles complémentaires. Nous profitons également de l'expertise du CC-IN2P3 en organisant des webinaires conjoints CC-IN2P3/labos, voire en diffusant les séminaires propres au CC-IN2P3.

Côté technique, Renaviso, l'outil choisi pour les webinaires, nous joue parfois des tours au dernier moment, mais... rien qui n'ait découragé les orateurs et le public de participer.

Vous souhaitez présenter, vous avez des sujets à proposer, vous êtes tenté par l'expérience ? N'hésitez pas, nul besoin d'être un « expert » pour participer, quelle que soit votre motivation, les webinaires sont faits pour vous ! Contactez-nous.

Retrouvez aussi les **webinaires présentés ou à venir**.

### Contact



## Indico : 10 000e demande de compte !

Le 24 juin à 13:05, [indico@in2p3](mailto:indico@in2p3) recevait sa 10 000e demande de compte :

```
Indico Mailer                               13:05
[Indico] Demande de nouveau compte provenant de Mauricio GÓMEZ VILORIA      2 Ko
Dear Administrator, Mauricio GÓMEZ VILORIA has created a new account in Indico. In order to
activate it, please go to this URL: <https://indico.in2p3.fr/user/10000/account/>
```

Bien sûr, cela ne donne pas le nombre de comptes réellement actifs, mais cela indique une utilisation soutenue, et toujours grandissante, de cet outil de gestion d'évènements adapté à nos disciplines puisque développé et maintenu par le CERN, et qui se révèle assez unique sur le marché.

**Le site Indico** **Le site Indico à l'IN2P3**



## Le projet PLUME devient le service FENIX

Le **site PLUME** a été mis en veille fin 2013. A l'été 2015, **Resinfo** (avec le soutien de **Mathrice**) a proposé de reprendre ce projet. Le site est actuellement hébergé dans une VM offerte par Mathrice et administrée par Resinfo. Le comité de pilotage composé de Sophie Nicoud (chef de projet, CNRS, LIRMM), Anne Cheylus (CNRS, ISCMJ), Damien Ferney (UBP, LMBP), Dirk Hoffmann (CNRS/AMU, CPPM/IN2P3), Karl Oulmi (CNRS, IBL), Laurent Perochon (VetAgro-Sup, METAFORT) et Henri Valeins (CNRS, RMSB), s'est donné pour objectif de concevoir une nouvelle plateforme, à partir de l'expérience acquise de PLUME.

Les objectifs du service sont : Le partage et la valorisation des réalisations techniques et des outils utilisés et/ou développés par la communauté ESR à l'aide de fiches descriptives. Ces réalisations ou outils pouvant être des procédés, des méthodes, des logiciels, des algorithmes, des diagrammes, des réalisations internes et « bouts de code » développés dans une entité de l'ESR.

Un nouveau site devrait voir le jour en début 2017, sous le nom FENIX, pour « Fiches d'Évaluation Normalisées Issues de l'eXpérience ». L'information sur l'avancement du projet sera diffusée via les listes ASR et devLog, les réseaux, et le site historique PLUME.

Cette démarche construite sur le succès de PLUME, en faveur de la valorisation des logiciels, sera à terme portée à d'autres domaines techniques spécifiques aux réseaux de la **Mission pour l'interdisciplinarité du CNRS**.

Pour en savoir plus, vous pouvez vous inscrire sur **la liste fenix-news**.