

Description d'une architecture de flux automatisés.

C. Boitel, septembre 2014

Sites de mesure, sites d'observations, laboratoires, centre d'expertise, de calcul ou de diffusion, réseaux nationaux ou internationaux, constituent à différentes échelles les maillons qui permettent de produire et proposer à la communauté scientifique des données d'observations simples ou élaborées, répondant à certains critères de qualité.

Les activités représentés qui vont de l'acquisition à la mise à disposition des données en passant par la mise en forme et l'application de traitements, se trouvent rarement toutes localisées sur un site unique. Ainsi le choix des lieux d'observations répond à des critères différents de ceux d'un centre de calcul.

Cela signifie que les données doivent circuler entre différents acteurs ayant la double vocation de contribuer au transfert de données vers un service connu de l'utilisateur et/ou produire des données à valeur ajoutée.

Ce document basé sur l'expérience de plusieurs sites d'observations, propose une description d'une architecture de flux de données applicable à la fois à une gestion locale et à une gestion répartie entre différents lieux disposant des ressources utiles à l'élaboration des données finales.

Au-delà de cette description, l'objectif est de proposer aux centres d'observation, une organisation et des règles de fonctionnement les plus simples possibles qui puissent servir de base soit à des développements communs soit à des développements "compatibles" d'outils de transfert et de traitement de données.

Nœuds et modules

Nœud est le terme choisi dans ce document pour désigner les sites, matériels locaux ou distants par lesquels les données transitent pour atteindre leur destination finale. Ces nœuds peuvent être proches et même situés sur une même machine physique ou éloignés et communiquer par le réseau. Leur rôle est d'héberger des modules logiciels qui peuvent simplement servir à relayer des données entre une source et une destination (passerelle réseau) ou à réaliser un traitement sur des données reçues.



Suivant le détail apporté à sa description, le flux entre deux nœuds peut jusqu'à un certain point être décrit comme étant constitué de plusieurs étapes relativement indépendantes et en tous les cas capables de fonctionner de manière autonomes et asynchrones l'une envers l'autre.



Ce travail d'identification des étapes minimales est un principe général applicable à tout développement jusqu'à obtenir dans l'idéal des modules irréductibles c'est-à-dire dédiés à la réalisation d'une tâche élémentaire. Les modules obtenus ont alors chacun un rôle bien défini et un niveau de complexité faible facilitant la maintenance du code.

Dans cette architecture il n'y a pas de code central pour piloter l'enchaînement des tâches à réaliser ou pour transmettre des informations. Les modules doivent donc s'entendre sur un protocole plus ou

moins élaboré suivant la complexité des informations à transmettre. L'exécution des modules est asynchrone et dépendante des données d'autres modules. Quant à l'échange de données il s'appuie sur des protocoles réseau classiques (ftp, ssh, mél, partage réseau, ...), il suffit que les deux nœuds l'aient à disposition, l'un comme client et l'autre comme serveur et que le contexte de sécurité le permette (pare-feu, proxy unidirectionnel, etc...). Dans le cas de modules situés sur une même machine, l'échange des données se fait en direct, par exemple au travers de répertoires partagés.

Dans tous les cas les échanges de données sont unidirectionnels. Aucune réponse immédiate n'est attendue en retour d'une transmission. Ce n'est donc pas une architecture de type client/serveur classique (genre client et serveur web) qui viserait une communication temps réel ou quasi temps réel entre les deux nœuds. Par contre rien n'empêche d'inverser les rôles des nœuds avec un nœud B prenant l'initiative et retournant en temps différé une réponse à un nœud A.

Les données

Deux types de données sont à distinguer :

- Les données scientifiques issues directement ou indirectement des observations
- Les informations accompagnant les données d'observation qui peuvent être de type métadonnées, simples descriptions des données, ou de type ordre qui permettent de préciser les tâches à réaliser.



Les protocoles réseau utilisés pour la transmission de ces deux types de données peuvent pour des facilités de traitement, être dissociés (transfert des données d'observation par ftp et données d'information par mél). Il reste cependant plus cohérent et probablement plus solide de réaliser l'envoi des deux types de données ensemble au cours d'une seule action de transfert.

Suivant le besoin, il est par ailleurs envisageable de ne transférer qu'un seul de deux type de données soit des données d'observations soit des données d'informations.

Exemple

Deux nœuds "compatibles", c'est-à-dire avec des fonctionnalités complémentaires en terme de communication (un client et un serveur), échangeant à la fois des données d'observations et des données d'informations.

Nœud A pouvant uniquement	Nœud B pouvant
<ul style="list-style-type: none"> • envoyer des données par ftp • recevoir des informations par mél 	<ul style="list-style-type: none"> • réceptionner des données par ftp • envoyer des méls

Scénario :

Phase 1 : le nœud A envoie des données au nœud B

Phase 2 : B envoie un mél (accusé réception) une fois certaines vérifications effectuées et met un compte rendu à disposition.

Phase 3 : récupération du compte-rendu par A (aurait aussi pu être envoyé par le mél de B).

L'astérisque (*) indique celui qui a l'initiative.



Les directives (ordres)

Dans le schéma le plus simple les actions sont implicites ou plutôt connues à l'avance et dépendantes des données d'observation reçues. Aucune donnée de type information n'est transmise pour préciser le travail à réaliser. L'apparition des fichiers de données sur le nœud de destination et leurs caractéristiques (nom, contenu, etc...) sont des informations suffisantes pour choisir les actions prédéfinies à entreprendre.

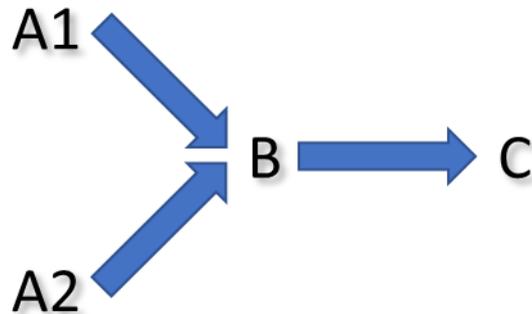
Lorsque les informations portées par les fichiers de données sont insuffisantes pour déterminer l'action à réaliser, il devient nécessaire de transmettre des informations supplémentaires de type "données d'informations" ou directives lorsqu'il s'agit d'actions.

Dans ce cas il est indispensable d'avoir un langage commun, un protocole, entre les nœuds qui communiquent. Il existe de nombreux formats standardisés permettant de structurer de l'information et sur lesquels les modules de gestion de ces directives peuvent s'appuyer. Les plus connus sont peut-être le XML et le JSON. Le texte simple est aussi à envisager si la complexité des directives reste limitée. L'essentiel du travail réside dans la définition du format des directives.

Flux, chaînes de traitement

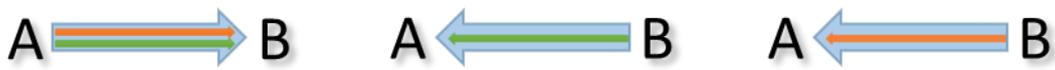
A partir de ces fonctionnalités il est facile dans l'absolu d'imaginer une architecture répartie permettant de construire une séquence de tâches correspondant à l'exécution d'une série de modules, qu'ils soient localisés sur un même système ou dispersés sur des systèmes éloignés. Il est ainsi possible de construire des chaînes de traitement relativement complexes en combinant les différents modules utiles à la réalisation de l'objectif. Chaque module réalise un traitement spécifique et met à disposition le résultat de son travail ou l'envoie vers un autre module, l'enchaînement étant décrit par les directives qui accompagnent les données.

Exemple 1 de deux centres A1 et A2 devant fournir à un réseau international C, des données formatées selon un standard imposé. Le centre B dispose des ressources et a les compétences du formatage des données. A1 et A2 vont tous les deux transmettre leurs données non formatées au centre B après conversion des formats originaux en format standard, transmet les données au nœud C.



Envoi vers B : données brutes + directive
formatage + directive envoi réseau C

Le nœud B peut même être considéré comme une boîte noire intelligente capable de répondre à d'autres directives lui demandant par exemple de garder les données formatées à disposition au lieu de les envoyer dans le réseau C. Ici A désire récupérer les données une fois formatées.



Etape 1 : Envoi données
brutes + directive
formatage + directive
demande d'envoi accusé
de fin de travaux

Etape 2 : Envoi accusé
de fin de travaux

Etape 3 : Récupération des
données formatées par A

Exemple 2 de production sur un serveur de calcul de données géophysiques dérivées de la combinaison de deux sources de données brutes provenant d'instruments de mesure.

A : pc d'acquisition

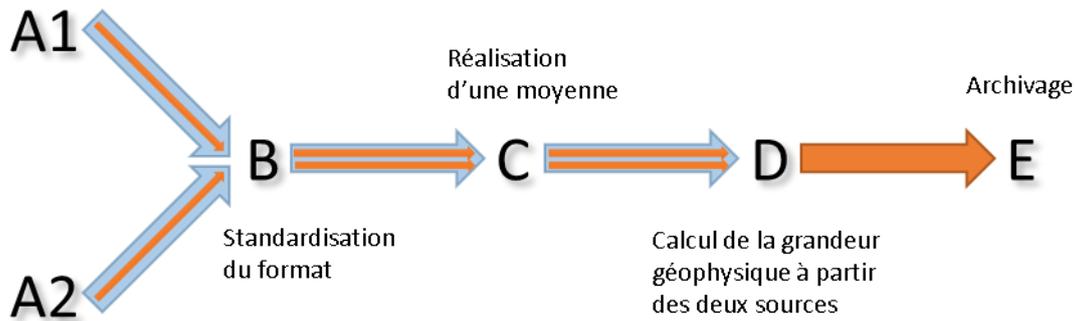
B : standardisation du format

C : calcul d'une moyenne

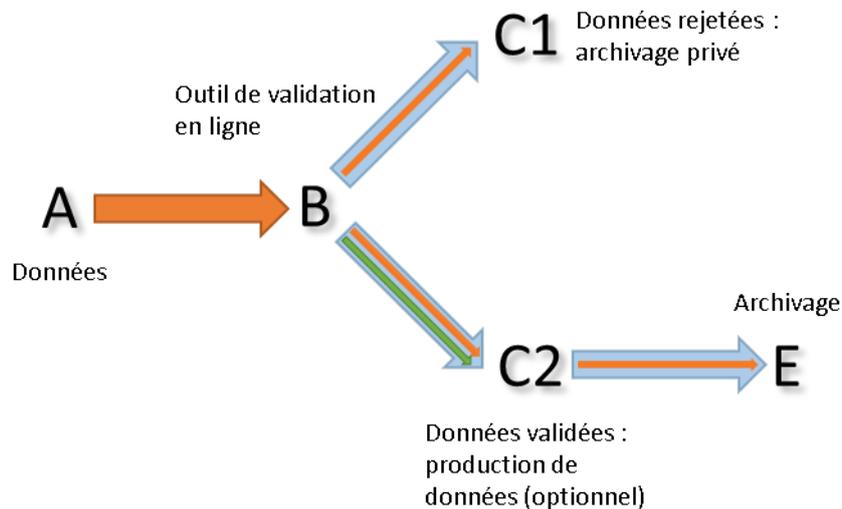
D : calcul grandeur géophysique niveau 3

E : archivage

Ici nous avons affaire à un seul nœud (le serveur) avec plusieurs modules logiciels et aucune directive n'est utilisée à partir du moment où les données sont standardisées.



Exemple 3 de validation de données. Les données sont envoyées sur un nœud disposant d'un outil web de visualisation et de validation de données. Les données sont retenues sur le nœud jusqu'à ce qu'un responsable vienne les valider ou les invalider. Le résultat est la génération par l'outil d'une directive indiquant la poursuite du processus (passage à un traitement, mise à disposition des données, etc...).



La supervision et le suivi

Ce n'est pas parce qu'un système est automatisé qu'il est exempt de défauts et capable de faire face seul à toutes les situations. C'est pourquoi chaque nœud doit produire un compte rendu circonstancié de son activité. La consultation des informations enregistrées doit permettre d'analyser à posteriori les problèmes qui peuvent survenir et de les régler.

Le second type d'information se rapporte d'avantage à un historique du travail réalisé en réponse aux directives soumises. Le but s'apparente à un traçage et doit confirmer ou non la bonne exécution des tâches.

L'ensemble de ces informations devrait être centralisé pour permettre une consultation simple et rapide.

La standardisation comme élément de simplification et d'automatisation

Travaillant dans des milieux hétérogènes que ce soit au niveau des matériels informatiques, des systèmes d'exploitation, des logiciels d'acquisition, des formats de fichiers de données, la tentation est de traiter chaque cas au fil de l'eau en mettant sur pied des solutions spécifiques. Cette démarche si elle est acceptable pour un nombre limité de cas devient vite ingérable avec leur augmentation.

Pour la mise en place d'une architecture plus générale dont l'ambition est de pouvoir prendre en charge une grande quantité de données automatiquement, il est important de rationaliser ses ressources en supprimant en particulier toute source de complexité et d'hétérogénéité. Une méthode est de se fixer un cadre de fonctionnement avec des règles simples.

Dans le cas de flux de données d'observation, la complexité provient pour une grande part de la production des données que ce soit les données brutes provenant des mesures que les données de sortie d'algorithmes de traitement. Un moyen de réduire ce problème est de convertir le plus tôt possible les données dans un format standard si possible par l'acquisition ou le traitement lui-même.

Des standards existent déjà comme le NetCDF ou le HDF. Ils ont l'avantage d'être le résultat du travail d'une très large communauté scientifique et ont été adoptés dans de nombreux centres de données. Utiliser son propre standard est également possible mais compliquera la diffusion des données et privera le gestionnaire des données de la très large panoplie d'outils déjà développés pour les formats standards.

Un standard en entraînant un autre, il est possible une fois le format des fichiers de données standardisé d'avoir des codes standardisés au moins pour les aspects de lecture, d'écriture et de contrôle, l'effort pouvant alors se concentrer sur l'algorithme de traitement.

Enfin, la base elle-même peut-être standardisée c'est à dire centralisée et organisée rationnellement (par date, type, site, instrument, niveau, ...).

Tous ces éléments de standardisation combinés réduisent mécaniquement le travail de gestion de la base de données, facilite le développement des algorithmes, et se prête naturellement à l'automatisation. La difficulté qui subsiste est le travail de conversion à fournir pour standardiser les données brutes de chaque flux de données, données dont le format qui plus est, peut parfois varier dans le temps. C'est une difficulté relative qui est largement compensée par la simplification de l'exploitation de la base consécutive aux travaux de standardisation.

L'implémentation

L'architecture présentée dans ce document se veut la plus simple possible tout en permettant la réalisation de tâches complexes. Cette simplicité doit non seulement faciliter sa mise en œuvre mais également sa maintenance et son évolution grâce à la participation d'un maximum de personnes même avec des compétences et des profils informatiques variés. A la simplicité de la conception doit suivre la simplicité de l'implémentation. Cela implique de choisir des outils de développement classiques, éprouvés, adaptés aux traitements à réaliser. Cela nécessite aussi la mise en place d'outils de suivi et de partage de codes entre les contributeurs du projet.

Conclusion

Cette architecture présente à mon sens plusieurs avantages

- elle intègre à la fois les besoins de diffusion et de traitement des données
- elle peut être utilisée pour des besoins locaux ou pour intégrer des contributions dispersées
- étant modulaire, il n'est pas nécessaire d'adopter l'ensemble des mécanismes décrit dans ce document. Ils peuvent être implémentés partiellement et/ou progressivement suivant les besoins et les ressources disponibles
- permet des réalisations simples comme complexes suivant le nombre et la sophistication des modules auxquels il est fait appel
- les modules contribuant à la réalisation d'une tâche peuvent être soit regroupés au sein d'un même système ou répartis sur plusieurs nœuds communiquant par le réseau
- les modules développés peuvent être mis à disposition sur des nœuds partagés à partir du moment où un protocole de communication standard est défini
- son caractère distribué permet d'utiliser les ressources et les compétences là où elles se trouvent
- elle incite à la rationalisation des développements, à la mutualisation et à la standardisation des outils
- elle implique potentiellement plusieurs acteurs de l'observation de l'atmosphère : observatoires, laboratoires associés, centres experts, centre de production et de diffusion des données, ... une communauté suffisamment large pour assurer les développements et leur pérennité.

