

ActiveData, un modèle de programmation pour la gestion des cycles de vie des données

A. Simonet¹, G. Fedak¹, M. Ripeanu²

(1) {Anthony.Simonet, Gilles.Fedak}@inria.fr
INRIA, University of Lyon, France

(2) matei@ece.ubc.ca
University of British Columbia, Canada

Calcul intensif et Sciences des données VICHY 2014

Plan

- 1 Introduction
- 2 Active Data
 - Principes & fonctionnalités
 - Implémentation
- 3 Etude de cas : Framework de surveillance des données pour APS
- 4 Discussion
- 5 Conclusion

Outline

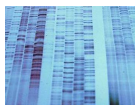
- 1 Introduction
- 2 Active Data
 - Principes & fonctionnalités
 - Implémentation
- 3 Etude de cas : Framework de surveillance des données pour APS
- 4 Discussion
- 5 Conclusion

Big Data ...

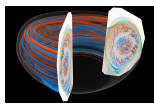
- ▶ Contexte: un déluge de données de sources multiples



Big Science



Instruments



Simulations



Internet



Open Data

- ▶ Quelques exemples:

- ▶ Les grands instruments scientifiques (LSST 30TB/nuit, LHC 1.5PB/week, OOOI), mais pas seulement (sequencing machine)
- ▶ Internet et les réseaux sociaux (Google, Facebook, Twitter, etc.)
- ▶ Open Data (Bibliothèques ouvertes, gouvernementales, code source)

→ impacte le processus de découverte scientifique (4th paradigme de la science)

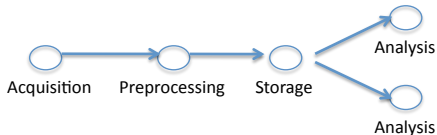
... ou Big Bottleneck ?

- ▶ La science des données produit de nouveaux défis :
 - ▶ comment construire les infrastructures adéquates ?
 - ▶ infra complexes et hétérogènes, amélioration des performances de bout-en-bout, optimisations inter-systèmes.
 - ▶ comment améliorer la productivité des scientifiques confrontés au Big Data ?
 - ▶ outils de gestion orientés données, qualité des jeux de données (provenance).
 - ▶ comment implémenter une science des données collaborative ?
 - ▶ motivation pour la publications et le partage des jeux de données, réseaux sociaux et applications Web.

- ▶ De nouveaux outils conceptuels et logiciels sont nécessaires pour manipuler les jeux de données scientifiques à travers des infrastructures distribuées et hétérogènes.

Le cycle de vie des données

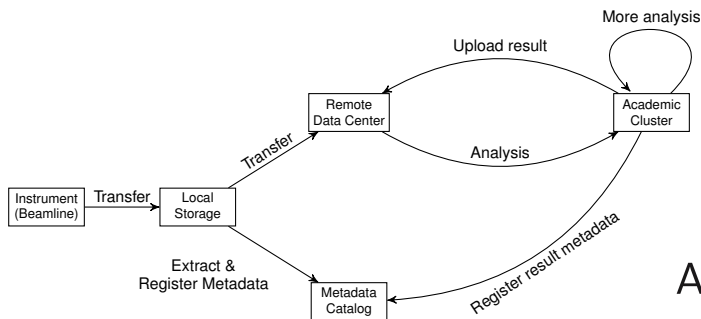
Data Life Cycle (DLC): La succession de phases opérationnelles par laquelle passent les données entre le moment où elles entrent dans un ensemble de systèmes jusqu'au moment où elles le quittent.



Cas d'étude : l'expérience APS

Exemple: Advanced Photon Source (APS) à Argonne National Lab

- ▶ 100TB de données brutes par jour
- ▶ Les données brutes sont pré-traitées et enregistrées dans un catalog Globus.
- ▶ Les données sont analysées par diverses applications
- ▶ Les résultats sont partagés entre les utilisateurs





Nos objectifs

Un système parfait devrait :

- ▶ capturer les étapes et les propriétés essentielles du cycle de vie : création, destruction, fautes, réplication, erreurs checking . . .
- ▶ Permettre aux systèmes existants d'exposer leur DLC intrinsèque
- ▶ Raisonner sur des jeux de données répartis sur des ensembles d'infrastructures et de systèmes hétérogènes.
- ▶ Simplifier la programmation d'applications DLC.

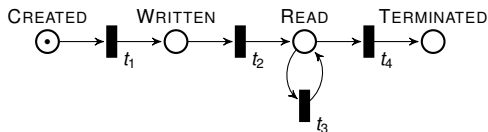
Outline

- 1 Introduction
- 2 Active Data
Principes & fonctionnalités
Implémentation
- 3 Etude de cas : Framework de surveillance des données pour APS
- 4 Discussion
- 5 Conclusion



Active Data principles

Le cycle de vie des données d'un système est représenté par un modèle inspiré des réseaux de Petri :
il contient des **Places** (états des données), des **Transitions** (actions sur les données) et des **Tokens** (une donnée).

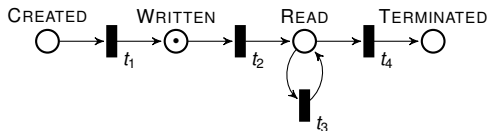


Chaque token a un identifiant unique qui correspond à la donnée dans le système.



Active Data principles

Le cycle de vie des données d'un système est représenté par un modèle inspiré des réseaux de Petri :
il contient des **Places** (états des données), des **Transitions** (actions sur les données) et des **Tokens** (une donnée).

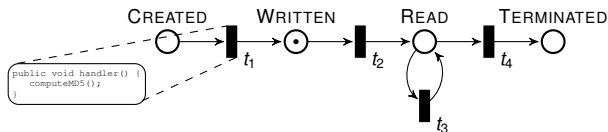


Une transition est déclenchée lorsqu'une donnée change d'état.



Active Data principles

Le cycle de vie des données d'un système est représenté par un modèle inspiré des réseaux de Petri :
il contient des **Places** (états des données), des **Transitions** (actions sur les données) et des **Tokens** (une donnée).



Du code peut-être associé aux transitions par le programmeur. Il est exécuté lorsque la transition est déclenchée.



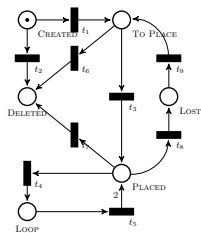
Active Data features

Le modèle de programmation Active Data et l'environnement d'exécution permettent :

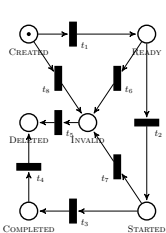
- ▶ de réagir à la progression des données dans leur cycle de vie
- ▶ de représenter des jeux de données distribués de façon transparente
- ▶ peut être intégré dans des systèmes existants de façon non-intrusive
- ▶ performance “scalable” et surcoût minimal



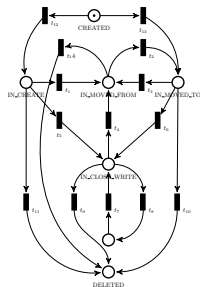
Integration with Data Management Systems



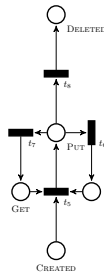
(a) Bitdew Scheduler



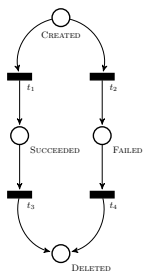
(b) Bitdew File Transfer



(c) inotify



(d) iRODS



(e) Globus Online

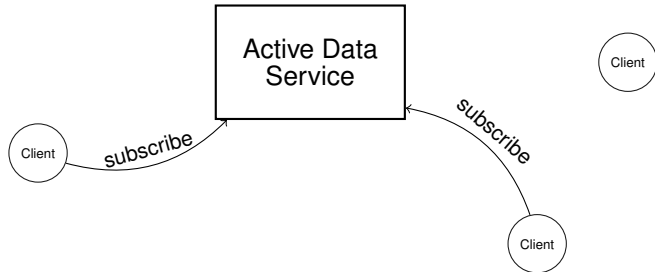
- ▶ BitDew (INRIA), programmable environment for data management.
- ▶ inotify Linux kernel subsystem: notification system for file creation, modification, write, movement and deletion.

- ▶ iRODS (DICE, Univ. North Carolina), rule-oriented data management system.
- ▶ Globus Online (ANL) offers fast, simple and reliable service to transfer large volumes of data.



Implémentation

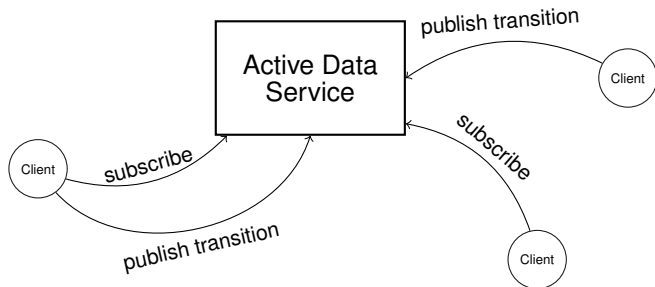
- ▶ Prototypé développé en Java
- ▶ communications basées Publish/Subscribe
- ▶ 2 types of souscriptions:
 - ▶ Toutes les transitions pour une donnée
 - ▶ Toutes les données pour une transition





Implémentation

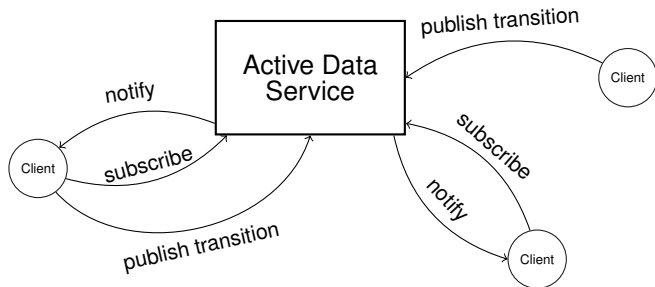
- ▶ Plusieurs possibilités pour publier une transition
 - ▶ Instrumentation du code
 - ▶ Analyse des logs
 - ▶ Système de notification existant
- ▶ The service ordonne les transitions par leur heures d'arrivée





Implémentation

- ▶ Clients exécute le code des transitions localement
- ▶ les codes des transitions sont exécutés
 - ▶ séquentiellement
 - ▶ de façon bloquante
 - ▶ dans l'ordre de publication des transitions





Evaluation de performances

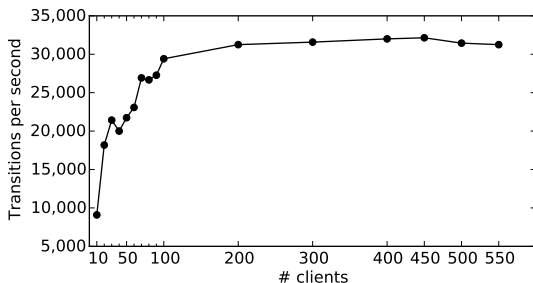


Figure: Average number of transitions per second handled by the Active Data Service

- ▶ Le client publie une série de 10K transitions
- ▶ Le serveur plafonne à 30K transitions/sec.

Applications

- ▶ Développement d'applications distribuées en se basant sur un paradigme de programmation orienté transition.
- ▶ Optimisation ou coordination de plusieurs systèmes entre eux.
- ▶ Rendre des applications "DLC-aware"
- ▶ Monitorer le cycle de vie et reconstituer la provenance des données

Outline

- 1 Introduction
- 2 Active Data
 - Principes & fonctionnalités
 - Implémentation
- 3 Etude de cas : Framework de surveillance des données pour APS**
- 4 Discussion
- 5 Conclusion

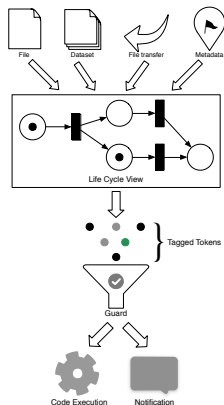
Etude de cas : Framework de surveillance des données pour APS

Objectifs :

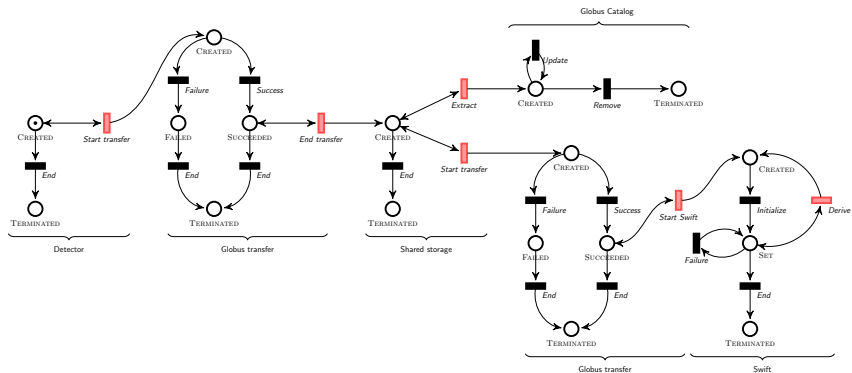
- ▶ monitorer la progression dans le DLC
- ▶ meilleure automatisation
- ▶ Notification et partage
- ▶ Détection et reprise sur erreurs

Extension à Active Data

- ▶ Associer des **Tags** aux données
- ▶ Installer des **Taggers** sur les transitions
- ▶ **Transitions gardées** : ne s'exécutent que pour les tokens possédant certains tags.
- ▶ Handlers de notification: Push.co, Twitter, gdoc, ifttt etc...



Modèle du cycle de vie des données de APS



Exemple: Data Provenance

Definition

Historique complète des dérivations et des opérations sur une donnée

- ▶ Estimer la qualité du jeu de données
- ▶ Garder les conditions d'acquisition et de transformation des données
- ▶ PASS: Provenance Aware Storage Systems
→ What about heterogeneous systems?

Example with **Globus Online** and **iRODS**

File transfer service

Data store and metadata catalog

Exemple: Data Provenance

Definition

Historique complète des dérivations et des opérations sur une donnée

- ▶ Estimer la qualité du jeu de données
- ▶ Garder les conditions d'acquisition et de transformation des données
- ▶ PASS: Provenance Aware Storage Systems
→ What about heterogeneous systems?

Example with **Globus Online** and **iRODS**

File transfer service

Data store and metadata catalog

Exemple: Data Provenance

Definition

Historique complète des dérivations et des opérations sur une donnée

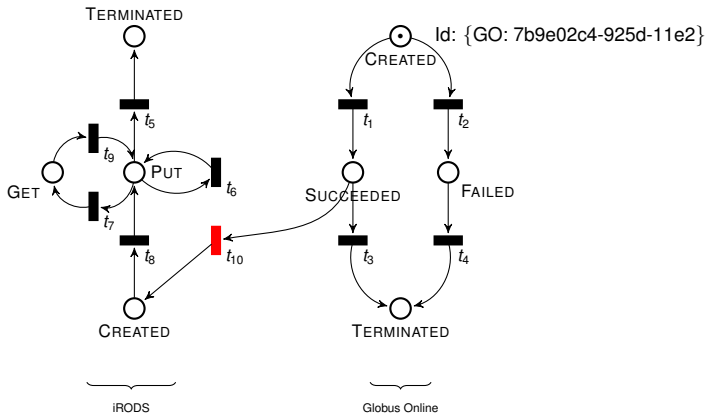
- ▶ Estimer la qualité du jeu de données
- ▶ Garder les conditions d'acquisition et de transformation des données
- ▶ PASS: Provenance Aware Storage Systems
→ What about heterogeneous systems?

Example with **Globus Online** and **iRODS**

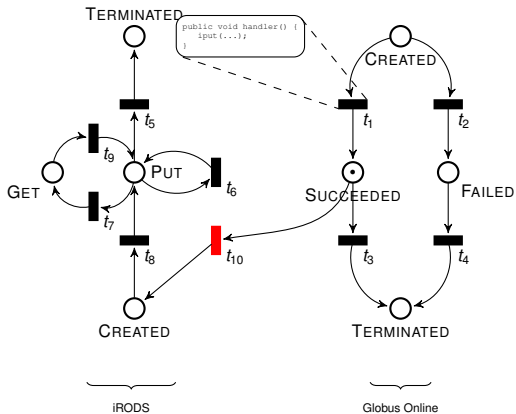
File transfer service

Data store and metadata catalog

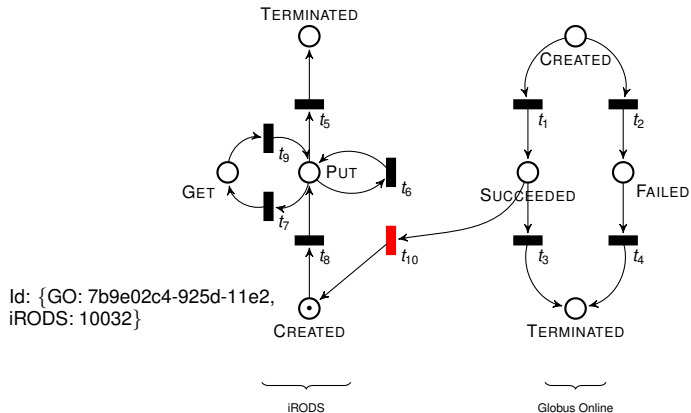
Data events coming from Globus Online and iRODS



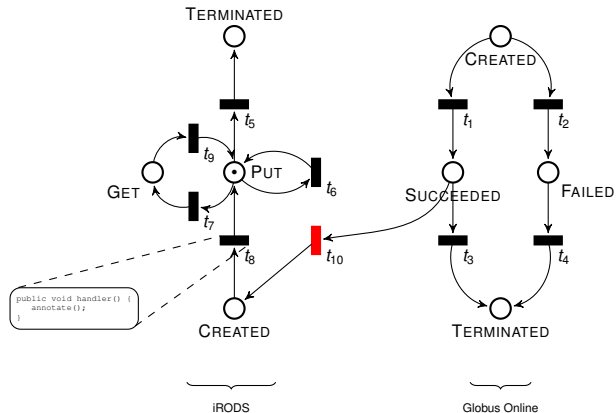
Data events coming from Globus Online and iRODS



Data events coming from Globus Online and iRODS



Data events coming from Globus Online and iRODS



```
$ imeta ls -d test/out_test_4628
AVUs defined for dataObj test/out_test_4628:
attribute: GO_FAULTS
value: 0
----
attribute: GO_COMPLETION_TIME
value: 2013-03-21 19:28:41Z
----
attribute: GO_REQUEST_TIME
value: 2013-03-21 19:28:17Z
----
attribute: GO_TASK_ID
value: 7b9e02c4-925d-11e2-97ce-123139404f2e
----
attribute: GO_SOURCE
value: go#ep1/~/test
----
attribute: GO_DESTINATION
value: asimonet#fraise/~/out_test_4628
```

Outline

- 1 Introduction
- 2 Active Data
 - Principes & fonctionnalités
 - Implémentation
- 3 Etude de cas : Framework de surveillance des données pour APS
- 4 Discussion**
- 5 Conclusion



- ▶ Façon simple et graphique de programmer des applications DLC
- ▶ Permet de vérifier formellement des propriétés du DLC
- ▶ Coordination entre les systèmes
- ▶ Facile à monitorer
- ▶ Facilite la tolérance aux pannes
- ▶ Interaction fine avec le DLC

- ▶ Complexité à raisonner en terme d'événements
- ▶ Manque de standardisation

Outline

- 1 Introduction
- 2 Active Data
 - Principes & fonctionnalités
 - Implémentation
- 3 Etude de cas : Framework de surveillance des données pour APS
- 4 Discussion
- 5 Conclusion



Active Data c'est . . .

- ▶ Data-centric & Event-driven
- ▶ Intégration des données au niveau du système

La suite

- ▶ Active Data comme spécification de composition de services
- ▶ Collection et flux de données

Merci!

Questions?