

Canadian
Geospatial
Data
Infrastructure



Infrastructure
canadienne
de données
géospatiales

Guide de consultation rapide sur les tests de conformité et l'optimisation de la performance des services de l'*ICDG*

12 mai 2009



Veillez envoyer votre courrier à l'adresse suivante :

GéoConnexions
Ressources naturelles Canada
615, rue Booth
Ottawa (Ont)
K1A 0E9

Téléphone : 613-947-8947
Numéro sans frais : 1 877 221-6213
Télécopieur : 613-947-2410

Courriel : info@geoconnections.org

Site Web : <http://www.geoconnections.org/>

Si votre courrier concerne la présente publication,
veuillez indiquer la date du document (mai 2009).

Le présent document est offert en ligne à l'adresse Web indiquée ci-dessus.

GéoConnexions est une initiative de partenariat national dirigé par Ressources naturelles Canada, dont l'objectif est le développement de l'Infrastructure canadienne de données géospatiales (ICDG).

Grâce à l'ICDG, les Canadiennes et les Canadiens ont accès en ligne à des informations géospatiales (p. ex., des cartes, des images satellitaires), ainsi qu'à des services et applications connexes, ce qui leur permet de prendre des décisions éclairées.

© Sa Majesté la Reine du chef du Canada 2009

GUIDE DE CONSULTATION RAPIDE – TESTS DE CONFORMITÉ ET OPTIMISATION DES SERVICES DE L'ICDG

Préface

La dernière décennie a été marquée par l'émergence de programmes d'infrastructure de données spatiales qui ont donné naissance à ce que d'aucuns appellent le « GéoWeb ». S'il est vrai que les systèmes d'information géographique (SIG) sont des technologies bien établies remontant aux années 60, l'accès aux données géographiques et leur utilisation sur Internet représentent une avancée relativement récente. Les progrès récents accomplis par des entreprises comme Google, Microsoft et Yahoo dans la géomatique grand public ont contribué à une meilleure compréhension de l'utilisation possible des données géospatiales dans de nombreuses applications Web. De plus, les progrès accomplis dans les technologies de développement Web (par exemple, AJAX - Asynchronous JavaScript and eXtensible Markup Language) permettent de disposer d'une plate-forme de développement d'applications Web hautement intégrées et performantes. L'accès transparent et fiable à ces applications exige que les utilisateurs et les développeurs valident et optimisent les services Web utilisés dans le développement d'applications.

Public cible

Le public visé par le présent guide comprend les développeurs de services Web, les analystes de données et les fournisseurs de services, ci-après appelés utilisateurs, en règle générale.

Portée

Le présent guide vise à fournir aux utilisateurs finaux et aux développeurs d'applications un document concis qui identifie et décrit les ressources disponibles pour tester les services Web (OWS) de l'Open Geospatial Consortium (OGC), et présente les meilleures pratiques pour optimiser la performance des OWS. Compte tenu des nombreuses technologies de mise en œuvre d'OWS disponibles, les tests, la validation et l'optimisation de la performance des OWS représentent un vaste domaine. Le présent document se veut un point de départ pour ceux intéressés par les sujets traités. Les OWS abordés sont les suivants : Service de cartographie Web (WMS), Service des entités Web (WFS), serveur réparti z39.50 utilisé pour la mise en œuvre du catalogue des services pour le Web (CSW) de l'OGC, GeoRSS et services KML.

1 Introduction et contexte

La vision de l'Infrastructure canadienne de données géospatiales (ICDG) est de permettre l'accès à des sources d'information géospatiales canadiennes fondées et complètes pour soutenir le processus décisionnel. L'élément moteur de l'ICDG repose sur la prémisse qu'un accès réparti et relié en réseau permet de disposer de données plus rapidement et plus facilement. Les services Web (par exemple, les OWS) fondés sur des normes ouvertes fournissent la base des interactions sur Internet et permettent aux utilisateurs d'accéder à des données géospatiales, d'en fournir et d'en échanger.

Plusieurs applications de l'ICDG reposent sur les services Web, plus particulièrement sur les OWS, pour les services de données et de traitement. C'est pourquoi une application fiable et adéquatement performante exige que les utilisateurs et les développeurs valident et optimisent les OWS utilisés dans le développement d'applications. Le présent document se veut une introduction à deux aspects importants du développement d'OWS fiables et utiles :

(1) Tests de conformité et d'interopérabilité : pour s'assurer que les services sont invoqués et qu'ils transmettent les données conformément aux spécifications approuvées.

(2) Optimisation de la performance des services : pour améliorer le temps de réponse et l'efficacité des services. Il s'agit d'un aspect important pour fournir une application utile et réduire la charge des fournisseurs de services.

La section 1 présente un aperçu des OWS et des applications client sélectionnées qui peuvent être utilisées pour accéder à ces services. La section 2 contient des directives sur les tests de conformité et d'interopérabilité. La section 3 renferme des renseignements sur l'optimisation de la performance des services.

1.1 Aperçu des services Web

Actuellement, les OWS fournissent la base des interactions sur Internet et permettent aux utilisateurs d'accéder à des données géospatiales, d'en fournir et d'en échanger. La présente section contient des références à des documents qui traitent de façon détaillée d'une approche en matière d'architecture de services Web dans le contexte de l'Infrastructure canadienne de données géospatiales (ICDG). Pour des informations plus détaillées sur ces sujets, consultez les documents suivants : [CGDI Vision](#), [CGDI Roadmap](#), [CGDI Architecture](#), [Formation en ligne de l'ICDG](#), [Coin des développeurs de l'ICDG](#), [Site Web de l'Open Geospatial Consortium](#) et [OGC Compliant Products](#).

1.2 Application client de services Web

Avant de passer aux tests de conformité et à l'optimisation des services, vous devez établir une méthode d'accès aux OWS. Plusieurs applications client conçues spécialement pour accéder à un ou à des services particuliers (par exemple, WMS, WFS) sont disponibles. Certaines de ces applications sont présentées à titre d'exemples dans la section 1.2.2. Toutefois, avant d'examiner les applications client propres aux services, il importe de souligner que vous pouvez accéder à la plupart des services Web conformes à l'ICDG au moyen d'un navigateur Web ordinaire, à jour et conforme aux normes.



Figure 1.1 : Réponse d'un navigateur Web à une demande à WMS

1.2.1 Navigateur Web comme application client

Les spécifications de mise en œuvre des interfaces de plusieurs OWS permettent d'utiliser un navigateur Web ordinaire comme application client. Au moyen d'une adresse URL, un navigateur Web peut invoquer une demande à un service Web donné de l'OGC, avoir une incidence sur l'exécution de la demande au moyen du protocole HTTP et recevoir une réponse du serveur au moyen du même protocole HTTP. Voici des exemples de demandes de services invoquées au moyen d'une adresse URL :

Demande à WMS :

```
http://wms.geobase.ca/wms-bin/cubeserv.cgi?request=GetMap &service=WMS&version=1.1.1& \
width=205&height=205&layers=DNEC_250K:ELEVATION/ELEVATION&styles=Dnec_CITS& \
format=image/png&srs=EPSG:42304&transparent=false&bbox=450000,2100000,550000,2200000
```

La réponse est présentée dans la figure 1.1. Veuillez noter que des paramètres (séparés par &) sont utilisés pour définir des aspects particuliers de la carte produite, soit : largeur (*width*) et hauteur (*height*) de l'image de sortie, couches (*layers*) affichées, *format* de la sortie, système de référence spatiale (*srs*), transparence (*transparent*) de l'image et cadre d'objet (*bbox*) dans les coordonnées *srs*.

Demande à WFS :

```
http://map.ns.ec.gc.ca/MapServer/mapserv.exe?map=/mapserver/services/envdat/config.map\
&service=WFS&version=1.0.0& request=GetFeature&TypeName=envirodat
```

Par défaut, la réponse à une demande à WFS est un document GML (Geography Markup Language). Ce document [eXtensible Markup Language](#) (XML) contient des données géospatiales et des métadonnées plutôt que la représentation graphique des données géospatiales produites par WMS. La demande à WFS présentée ci-dessus à titre d'exemple est utilisée pour obtenir des données des stations de surveillance de la qualité de l'eau à Terre-Neuve-et-Labrador. La réponse produite par un navigateur est présentée dans la

figure 1.2. Veuillez noter que le document GML combine les données géospatiales sous la forme de coordonnées x,y, avec des métadonnées décrivant les attributs importants des données. Par exemple, la balise `<gml:Box srsName="EPSG:4326">` utilise un code [EPSG](#) pour établir que les coordonnées spatiales imbriquées sont exprimées en degrés décimaux au moyen de l'ellipsoïde et du système de référence WGS 84.

```
- <wfs:FeatureCollection xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd http://www.ec.gc.ca/envirodat/atlantic http://map.ns.ec.gc.ca/envdat/map.aspx?SERVICE=WFS&VERSION=1.0.0&REQUEST=DescribeFeatureType&TYPENAME=envirodat&OUTPUTFORMAT=XMLSCHEMA">
- <gml:boundedBy>
- <gml:Box srsName="EPSG:4326">
  <gml:coordinates>-63.234200,46.845300 -52.691400,54.661700</gml:coordinates>
</gml:Box>
</gml:boundedBy>
- <gml:featureMember>
- <envdat:envirodat>
- <gml:boundedBy>
  - <gml:Box srsName="EPSG:4326">
    <gml:coordinates>-61.322500,52.228300 -61.322500,52.228300</gml:coordinates>
  </gml:Box>
  </gml:boundedBy>
- <envdat:msGeometry>
  - <gml:Point srsName="EPSG:4326">
    <gml:coordinates>-61.322500,52.228300</gml:coordinates>
  </gml:Point>
  </envdat:msGeometry>
  <envdat:Station_ID>NF02XA0001</envdat:Station_ID>
  <envdat:lat>52.2283</envdat:lat>
- <envdat:url_station_data>
  http://map.ns.ec.gc.ca/sp/profile.aspx?NL_ENV_SER_ID=NF02XA0001
  </envdat:url_station_data>
  <envdat:lon>-61.3225</envdat:lon>
```

Figure 1.2 : Réponse de WFS rendue par un navigateur Web

Bien qu'un navigateur Web soit une application client simple, sa simplicité en fait un outil intéressant pour tester les OWS et en optimiser la performance. Un navigateur Web offre une méthode d'accès aux OWS relativement transparente et peu coûteuse.

1.2.2 Applications de bureau

Certes, un navigateur Web permet d'accéder aux OWS conformes à l'ICDG, mais plusieurs applications exigent des fonctions client plus étendues. Par exemple, la visualisation d'une réponse de WFS est une utilisation type. Plusieurs applications client conformes à l'OGC ([OGC Compliant Products](#)) sont offertes. La copie d'écran présentée dans la figure 1.3 illustre comment l'outil de bureau libre [uDig](#) peut être utilisé pour accéder aux services WMS et WFS.

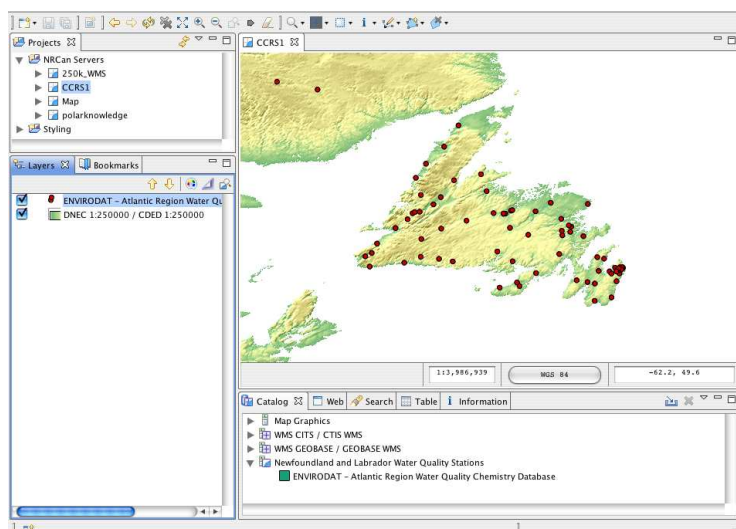


Figure 1.3 : Données provenant des ressources WMS and WFS produites par l'application uDig

Les applications logicielles de bureau, comme uDig, peuvent être très utiles lorsque vous utilisez des OWS. Ces applications vous permettent de vous familiariser rapidement avec les données offertes par les fournisseurs de services et d'évaluer la performance des services dans un environnement de bureau.

1.2.3 Utilisation d'autres outils pour accéder aux services Web de l'OGC

Comme nous venons de le voir, il est possible d'utiliser un navigateur Web ou une application de bureau pour accéder aux OWS. Certes, ces applications clients sont des outils d'accès puissants aux données des OWS, mais plusieurs applications du monde réel exigent le développement d'une fonctionnalité très précise. Divers outils peuvent être utilisés pour le développement d'applications personnalisées. Plusieurs langages de programmation et de script répandus, comme C, C++, Java, Python, Perl, PHP, JavaScript, etc., peuvent être utilisés pour accéder aux OWS. Ces divers langages permettent d'accéder aux OWS de différentes façons, mais tous fournissent un mécanisme permettant d'invoquer une demande et de recevoir une réponse. L'utilisation d'un langage de programmation et de script représente la méthode d'utilisation des OWS la plus puissante, la plus flexible et la plus complexe. De plus en plus de boîtes à outils de développement d'applications permettant de réduire considérablement la complexité des OWS sont offertes. Un bon exemple à cet égard est la bibliothèque [OpenLayers](#) JavaScript.

L'utilisation d'environnements de programmation et de script, bien que plus complexe que l'utilisation d'un navigateur Web ou d'une application de bureau, pourrait mieux convenir aux tâches plus complexes de validation et d'optimisation des services. Ces environnements permettent au développeur de personnaliser les demandes, de manipuler plus précisément les données fournies par un serveur, de dépister les erreurs et de calculer des paramètres importants, comme le temps de réponse. Des méthodes précises sont abordées plus en détail dans les sections suivantes du présent document.

2 Tests de conformité et d'interopérabilité

2.1 Conformité dans le contexte de l'Infrastructure canadienne de données géospatiales

Les principes directeurs de l'ICDG (reportez-vous au document [CGDI Vision](#)) définissent un modèle de services Web reposant sur des normes et des spécifications ouvertes et interopérables et permettant à l'utilisateur d'accéder aux données et aux services de façon transparente malgré la complexité de la technologie sous-jacente. Pour soutenir ces principes, les services de l'ICDG doivent conserver un certain niveau de conformité avec les normes et les spécifications approuvées par l'ICDG. Un niveau minimal de

conformité est important pour s'assurer que chacun peut explorer, comprendre et partager les données sans qu'il soit nécessaire de développer des outils d'accès propres aux services.

Bien qu'il ne s'agisse pas d'une exigence de l'ICDG, le test et l'évaluation de la conformité et de l'interopérabilité des services sont fortement recommandés pour s'assurer que les OWS sont conformes aux normes et aux spécifications approuvées par l'ICDG. Cette section se veut une introduction aux tests de conformité et d'interopérabilité des services. À titre de fournisseur ou de membre d'un groupe de développement OWS, vous êtes invité à confirmer vos activités reliées aux tests de conformité au moyen du programme [OGC Compliance Testing Program](#).

2.2 XML, XML Schema, XSLT

Au cœur des OWS se trouve un ensemble fondamental de fonctions XML. Le langage XML permet d'obtenir une représentation bien définie des données ainsi qu'un ensemble bien établi de règles de validité. Le langage [XML Schema](#) est une spécification qui permet aux machines d'appliquer les règles codées par des personnes. Étant donné que XML Schema est un élément important des tests et de la validation des services, nous vous recommandons de vous familiariser avec ce langage et les outils associés. De la même façon, le langage [eXtensible Stylesheet Language Transformations](#) (XSLT) est utilisé pour appliquer des tests qui peuvent ne pas être inclus dans le processus de validation XML.

Une solide connaissance de XML, XML Schema, XSLT et des autres langages associés est utile pour tous les niveaux de tests de conformité. Cette connaissance est nécessaire si vous prévoyez élaborer de nouveaux tests de conformité avec les spécifications OWS ou utiliser des profils ou des schémas d'applications. Les profils et les schémas d'applications permettent respectivement de définir un sous-ensemble d'une spécification de l'OGC (soit GML) et de modéliser les éléments d'information propres à une communauté particulière d'applications.

2.3 Utilisation d'outils de test et de validation de conformité

La présente section se veut une introduction aux principaux programmes et aux principales ressources disponibles pour tester et évaluer la conformité et l'interopérabilité des services, et indique comment établir un environnement de test au moyen d'applications de référence.

2.3.1 Programme CITE de l'OGC et outils associés

Actuellement, la principale ressource établie pour les tests de conformité des OWS est le programme [Compliance & Interoperability Testing & Evaluation Initiative \(CITE\)](#) de l'OGC. Le programme CITE s'inscrit dans un effort continu de l'OGC visant à élaborer des tests de conformité avec ses normes. Les outils de test de conformité peuvent être utilisés dans le cadre du processus de certification de produits ([Product Certification Process](#)) d'un fournisseur ou pour permettre aux fournisseurs de services et aux utilisateurs de tester les services.

Dans un contexte type de tests de conformité d'un OWS, plusieurs conditions doivent être satisfaites pour confirmer la conformité du service. Premièrement, un document XML produit à la suite de la mise en œuvre du OWS doit valider les résultats par rapport à un ou plusieurs documents XML Schema au moyen d'un analyseur de validation (par exemple, <http://xerces.apache.org/>). Deuxièmement, les assertions particulières établies par l'auteur des tests doivent être vérifiées. XML Schema joue un rôle central pour s'assurer que la première exigence est satisfaite. Les documents XML Schema utilisés pour la validation peuvent être ceux produits par l'OGC dans le cadre du processus d'élaboration des normes ou des documents de schéma d'application (*Application Schema*) produits par des personnes ou des organisations pour une application particulière. XSLT est utilisé en langage TSL ([Test Scripting Language](#)), confirmant l'importance de ces langages XML pour le processus de test.

Actuellement, le portail CITE permet de tester la conformité des services WMS, WFS, WCS et CSW avec les spécifications approuvées les plus récentes au moyen du logiciel [Test Evaluation And Measurement \(TEAM\)](#)

[Engine](#). Le portail CITE donne également accès à des ressources permettant de valider des documents GeoRSS, GML et Web Map Context dans une capacité de non-certification (OGC 2009a). Dans ce document, l'expression *testing* désigne l'utilisation du logiciel TEAM Engine pour appliquer la série de tests et l'expression *validation* désigne l'utilisation d'un outil pour valider un document XML par rapport à un document XML Schema.

2.3.2 Aperçu des tests au moyen du logiciel TEAM Engine

Deux éléments doivent être en place avant de commencer des tests au moyen du logiciel TEAM Engine :

- S'il y a lieu, le fournisseur de services doit charger les jeux d'essai requis. Les liens aux jeux d'essai se trouvent sur les pages décrivant les tests particuliers (soit les tests [CSW](#), et [WMS](#) et divers tests [Beta](#)).
- Le fournisseur de services ou l'utilisateur doit s'inscrire ([Register for an account](#)) sur la page d'accueil [TEAM Engine](#) et sélectionner ensuite *Start Testing* (reportez-vous à la figure 2.1) pour démarrer une nouvelle session de test.

Il est important de souligner que plusieurs tests de conformité exigent qu'un jeu d'essai soit installé pour permettre l'exécution des tests. Si vous êtes un développeur d'applications ou un analyste de données sans privilèges administratifs pour le service, vous devez travailler en collaboration avec votre fournisseur de services pour exécuter les tests de conformité. De plus, dans certains cas, des couches d'authentification (soit, connexion) et de sécurité (soit, connexion sécurisée) peuvent avoir été activées pour les OWS par un fournisseur de services. Bien que l'authentification et la sécurité ne fassent pas partie des spécifications de l'OGC, certains services mettent en œuvre cette fonctionnalité « au haut » des OWS. Dans ce cas, un script de test donné doit être modifié pour tenir compte des étapes supplémentaires reliées aux couches d'authentification et de sécurité.

2.3.3 Mises en œuvre de référence

L'OGC a établi une série de mises en œuvre de références ([Reference Implementations](#)) qui sont définies comme suit : mise en œuvre pleinement fonctionnelle d'une spécification en regard de laquelle d'autres mises en œuvre peuvent être évaluées. L'OGC fournit des mises en œuvre de référence libres pour assurer la transparence maximale de ses spécifications pour les fournisseurs et les clients (OGC 2009b). Les mises en œuvre de référence sont des outils de test utiles si vous ne disposez pas d'un accès administratif à un service Web existant de l'OGC. Ces mises en œuvre de services sont déployées au moyen de conteneurs de servlets, comme le logiciel très répandu [Apache Tomcat](#). De plus, [GeoServer](#), la mise en œuvre de référence de WFS, et [GeoNetwork](#), la mise en œuvre de référence de CSW, peuvent être déployées au niveau local comme application autonome.

TEAM Engine

(Test, Evaluation, And Measurement Engine)



Welcome

The Test, Evaluation, And Measurement (TEAM) Engine is a test script interpreter. It executes test scripts written in [Compliance Test Language \(CTL\)](#) to verify that an implementation of a specification complies with the specification.

The following test suites are available:

- [Web Map Context Documents 1.1.0](#)
Validates that a Web Map Context 1.1.0 document is compliant.
- [CSW 2.0.1 Compliance Test Suite](#)
Validates a CSW 2.0.1 catalogue implementation against all applicable abstract test suites. No specific CSW application profile is presumed. These tests are not intended to stand alone, but to be incorporated into profile-specific test suites.
[Test data](#)
- Web Feature Service (WFS) 1.0.0
Validates that a WFS 1.0.0 is compliant.
- [WFS 1.1.0 Compliance Test Suite](#)
Verifies that a WFS 1.1.0 implementation complies with a given conformance class.
[Test data](#)
- Web Map Service (WMS) 1.1.1
Validates that a WMS 1.1.1 is compliant.
[Test data](#)
- [Web Map Service \(WMS\) 1.3.0](#)
Validates that a WMS 1.3.0 is compliant.
[Test data](#)

⚠ Test data must be loaded before running a test suite!

[Start Testing](#)

Figure 2.1 : Page de bienvenue du logiciel TEAM Engine qui se trouve à l'adresse suivante : <http://cite.opengeospatial.org/teamengine/>. Cette page contient des liens à la documentation sur les tests des services et permet à l'utilisateur de démarrer une session de test au moyen du lien *Start Testing*.

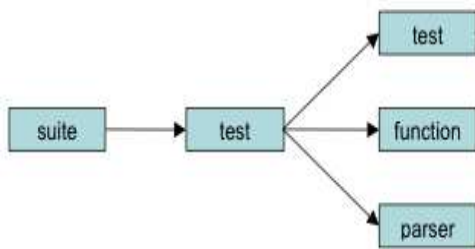
Un exemple de test de conformité et d'interopérabilité faisant appel à une mise en œuvre de référence de l'OGC est présenté dans la section 2.3.5. Pour faciliter la compréhension de cet exemple, la section 2.3.4 présente un bref aperçu du langage CTL (Compliance Test Language).

2.3.4 Aperçu du langage CTL (Compliance Test Language)

TEAM Engine est un produit logiciel qui exécute des scripts écrits dans un langage XML appelé *Compliance Test Language* (CTL). Des explications détaillées sur le langage CTL dépassent la portée du présent guide, mais nous en abordons ici les éléments importants pour faciliter la compréhension des sujets qui seront traités ultérieurement. Le résumé présenté ici repose sur le document [Compliance Test Language \(CTL\) Discussion Paper](#). Pour des informations plus détaillées, veuillez vous reporter à ce document.

Un document CTL contient divers éléments, soit *suite*, *test*, *function*, *request* et *parser* (figure 2.2).

L'élément **<suite>** est un objet de conteneur. L'élément **<test>** établit un test individuel qui inclut une



assertion et un code de programme pour tester l'assertion. Une assertion est un énoncé tiré d'une spécification de l'OGC testée, qui est vraie pour une mise en œuvre valide. Le code de programme utilisé pour vérifier l'assertion est écrit en langage XML et peut contenir des instructions CTL ou XSL. Pour simplifier l'élaboration d'un code de programme, les auteurs de scripts peuvent également utiliser des éléments **<function>** personnalisés, qui permettent de vérifier l'assertion et d'établir le déroulement du programme. L'élément **<parser>** indique l'analyseur qui sera utilisé dans un test et relie ce dernier à sa mise en œuvre externe Java^{MC}. Des analyseurs sont utilisés dans l'instruction **<request>** pour convertir la réponse d'un OWS en un document XML bien formé aux fins de traitement.

Figure 2.2 : Modèle d'objet en langage CTL

Les éléments ci-dessus sont combinés à d'autres éléments pour

créer un script de test. La figure 2.3 présente le rendu d'une section du script de test WFS 1.0.0. Les éléments **<function>** et **<parser>** sont initialisés, suivis par la définition des éléments **<suite>** et **<test>**. Chaque test est appliqué et les messages d'avertissement et d'erreur sont signalés à l'utilisateur. Le logiciel TEAM Engine, développé dans le cadre du programme CITE, offre une façon relativement conviviale d'accéder aux outils de tests de conformité et de les gérer. Toutefois, pour exécuter avec succès les tests de conformité, il importe de vous familiariser avec le logiciel TEAM Engine et ses technologies sous-jacentes.

```

+ <function name="citef:validate-gml-schema"></function>
+ <function name="wfs:sleep"></function>
+ <parser name="parsers:HeadersParser"></parser>
+ <parser name="parsers:XMLValidatingParser"></parser>
+ <parser name="example:ImageStats"></parser>
+ <parser name="example:StringLength"></parser>
+ <parser name="example:TransformXML"></parser>
+ <parser name="wfs:GMLValidatingParser"></parser>
+ <test name="wfs:main"></test>
+ <suite name="wfs:main_wfs"></suite>
+ <test name="wfs:test1.0.0-basic-describefeaturetype-get-1"></test>
+ <test name="wfs:test1.0.0-basic-describefeaturetype-get-2"></test>
  
```

2.3.5 Test d'un service Web de l'OGC

Figure 2.3 : Éléments de niveau supérieur du script de test WFS 1.0.0 en langage CT

Tel qu'indiqué précédemment, le logiciel TEAM Engine est en mesure de tester plusieurs OWS. Dans le présent exemple, nous utilisons la spécification WFS. Le processus de test des autres spécifications (par exemple, WMS ou CSW) est très semblable à celui présenté ici. Le logiciel TEAM Engine peut tester les spécifications [WFS 1.0.0](#) et [WFS 1.1.0](#). Chaque série teste les demandes WFS de base (GetCapabilities, DescribeFeatureType, GetFeature), transactionnelles (Transaction) et de verrouillage (LockFeature, GetFeatureWithLock).

Une fois vous vous êtes inscrit et que le jeu d'essai est installé, vous pouvez démarrer une session de test. Pour ce faire, il suffit de cliquer sur le lien *Start Testing* dans la page d'accueil TEAM Engine (<http://cite.opengeospatial.org/teamengine>, figure 2.1) : le système affiche alors l'écran de connexion. S'il s'agit de la première connexion, vous devez sélectionner une série de tests (soit, WFS 1.0.0 ou WFS 1.1.1) dans la liste des options. Ensuite, vous devez entrer une description identifiant la session. Le système affiche alors un formulaire contenant les options applicables aux tests WFS. Ces options diffèrent selon la version de la spécification qui est testée. Point commun aux deux tests : vous devez entrer une adresse « Capabilities URL » dans le formulaire : `http://hostname:port/path?request=GetCapabilities&service=WFS&version=x.x.x`

Cette information permet à l'analyseur (parser) du script de test d'extraire le document GetCapabilities du serveur testé. Le document GetCapabilities est ensuite utilisé par le script de test pour appliquer la série de tests appropriée. Par exemple, si le document GetCapabilities indique que le serveur n'est pas transactionnel, alors les tests transactionnels ne sont pas appliqués. Une fois les options voulues sélectionnées, il suffit de

cliquer sur le bouton Start pour démarrer le test. Les tests de conformité étant, par nature, exhaustifs, la durée du processus de test peut être relativement longue : des tests WFS d'une durée de 5 à 7 minutes ne sont pas chose rare. Pour s'assurer que le processus de test se poursuit comme prévu, il vaut mieux le surveiller au moyen de la méthode qui convient le mieux à votre navigateur.

Une fois le processus de test terminé, le système affiche un journal contenant les résultats des tests. Ce journal contient une copie du document GetCapabilities du service, les résultats de chaque test et le résultat global de tout le processus de test. Pour une conformité complète, tous les tests doivent indiquer « Passed » comme résultat d'évaluation. La figure 2.4 présente les résultats des tests exécutés avec succès à l'égard d'une instance de la mise en œuvre de référence GeoServer.

```
Test wfs:main (s0068)
Assertion: This WFS is valid
Form w6ac25b3b1b1_1:
  VAR_WFS_CAPABILITIES_URL=http://87.7.38.17:8080/geoserver/ows?service=WFS&request=GetCapabilities&version=1.0.0
  all=all
Request w6ac25b3b1b1_1:
  Method: get
  URL: http://87.7.38.17:8080/geoserver/ows?service=WFS&request=GetCapabilities&version=1.0.0
  Response from parser ::
    <WFS_Capabilities xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.opengis.net/wfs"
      xmlns:ogc="http://www.opengis.net/ogc"

Subtest s0068/w6ac25b3b9d179b1_1 Passed
Subtest s0068/w6ac25b3b9d181b1_1 Passed
Subtest s0068/w6ac25b3b9d183b1_1 Passed

Subtest s0068/w6ac25b3b9e1101b1_1 Passed
Result: Passed
```

Figure 2.4 : Sous-ensemble de résultats d'une session de test réussie

2.3.6 Élaboration de nouveaux tests de conformité

Des tests de conformité n'ont pas été élaborés pour toutes les spécifications OWS. Le test de ces services exige l'élaboration d'une série de tests. La même chose s'applique si un service utilisant un profil ou un schéma d'application doit être testé pour en vérifier l'interopérabilité avec d'autres systèmes.

La portée des normes et des spécifications de l'OGC est étendue et, par conséquent, plusieurs documents XML Schema sont utilisés pour définir les langages XML des OWS. Vous pouvez trouver ces documents XML Schema à l'adresse suivante : <http://schemas.opengis.net/>. À l'aide de ces documents, vous pouvez utiliser le langage CTL pour élaborer de nouveaux tests de conformité. Les scripts de tests exécutables existants ([Executable Test Scripts](#)) sont une référence utile lorsque vous devez élaborer de nouveaux tests de conformité.

2.3.7 Autres outils de validation

Actuellement, le logiciel TEAM Engine prend en charge l'exécution d'un vaste ensemble de tests pour les services de l'OGC sélectionnés. En plus du logiciel TEAM Engine, le programme CITE offre des outils conviviaux pour valider divers types de données géospatiales XML.

[GeoRSS](#) est une spécification simple permettant de baliser des fils de syndication vraiment simples ou des fils RSS au moyen d'informations sur l'emplacement. GeoRSS est un format fondé sur le langage XML qui comprend le codage « simple » et le codage « GML ». Le codage GML prend en charge un plus grand

nombre de fonctions que le codage simple. Les deux types de codage étant fondés sur le langage XML, les documents GeoRSS peuvent être validés par rapport aux documents XML Schema associés. Le portail du programme CITE offre un outil permettant la validation d'un document GeoRSS à codage simple ou GML ([Tool for validating Simple or GML GeoRSS](#)). La case 2.1 présente un exemple d'un document GML/GeoRSS qui peut être sauvegardé et validé au moyen de l'outil de validation GeoRSS du programme CITE. Des outils de validation OWS sont disponibles en plus de ceux du programme CITE.

[KML](#) est un langage XML qui permet la visualisation de données géographiques, y compris le balisage cartographique et le rendu du contenu multimédia associé. KML s'est imposé à l'origine comme le format utilisé par l'outil de visualisation très répandu Google Earth. Le langage KML, version 2.2, a été adopté comme une norme de mise en œuvre de l'OGC. À titre de langage XML, KML peut être validé par rapport à des documents XML Schema associés au moyen de divers analyseurs de validation ou d'un outil en ligne, par exemple [KML Validator](#) de Galdos Systems Inc. L'outil [KML Validator](#) permet de valider des documents KML par rapport au schéma de spécification KLM de l'OGC ([OGC KML Specification](#)). Bien qu'un fichier KML puisse fonctionner avec divers outils logiciels, il est possible qu'il ne soit pas entièrement conforme à la spécification. L'outil KML Validator produit un rapport d'erreur et fournit des suggestions sur la façon de respecter les exigences en matière de conformité.

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:georss="http://www.georss.org/georss"
  xmlns:gml="http://www.opengis.net/gml">
  <title>Features</title>
  <subtitle>International earthquake observation labs</subtitle>
  <link href="http://example.org"/>
  <updated>2005-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
    <email>jjd@jddomain.com</email>
  </author>
  <id>urn:uuid:60a76c80-d888-11d9-b93C-0003939e0af6</id>
  <entry>
    <title>Feature data</title>
    <link href="http://example.org/2008/08/08/atom01"/>
    <id>urn:uuid:1225c695-cfb8-4add-aaaa-80da344efa6a</id>
    <updated>2005-08-17T07:02:32Z</updated>
    <summary>Interesting feature.</summary>
    <georss:where>
      <gml:Point>
        <gml:pos>45.256 -71.92</gml:pos>
      </gml:Point>
    </georss:where>
  </entry>
</feed>
```

Case 2.1 : Exemple d'un document GML/GeoRSS

2.4 Sommaire : Validation et conformité aux spécifications

La présente section a souligné l'importance des tests de conformité et d'interopérabilité dans le contexte de l'ICDG. L'importance de XML Schema pour les tests et la validation a été établie. Le reste de la section a consisté en directives sur l'établissement d'un environnement de test et a montré et illustré le processus de test pour divers services Web de l'OGC.

3. Optimisation de la performance des services

La présente section contient un aperçu des stratégies et des techniques d'optimisation de la performance des services. Les OWS sont souvent mis en œuvre comme une « superposition » de couches ou de niveaux technologiques physiques et logiques. Cette section présente traite de cinq sujets importants pour l'optimisation de la performance des données aux cinq niveaux courants dans la superposition technologique, soit au niveau des bases de données, des serveurs; des OWS; des réseaux et des applications.

L'optimisation de la performance des applications Web en général, et de celle des OWS en particulier, repose souvent sur la réduction du volume de données nécessaires pour une application. Plusieurs de nos suggestions ont trait à la réduction du volume de données. Pour bien comprendre les nombreux exemples présentés ici et utiliser ces stratégies, vous devez bien connaître votre environnement d'applications.

3.1 Optimisation au niveau des bases de données

La représentation des aspects complexes du monde géographique au moyen d'outils informatiques contemporains peut donner lieu à des bases de données volumineuses. Ces dernières, qui comprennent des comptes rendus détaillés de l'environnement, comme des cartes urbaines ou de l'imagerie satellite à haute définition, peuvent tirer parti des techniques présentées ici.

3.1.1 Généralisation des données : L'application des pratiques de sélection et de simplification reliées à la généralisation cartographique ([Cartographic generalization](#)) peut être une stratégie utile pour optimiser la performance des OWS au niveau des bases de données. La sélection demande de choisir les données les plus importantes pour une application. La simplification demande de modifier le format ou la forme des détails cartographiques, soit habituellement lorsque les données sont transformées d'une utilisation à grande échelle à une utilisation à petite échelle. À la différence d'une carte sur papier, les données spatiales numériques sont stockées, en règle générale, sans une échelle inhérente. Toutefois, il est important de savoir que la précision, l'exactitude et le détail d'un certain ensemble de données conviennent mieux à une échelle d'analyse ou d'affichage particulière. Pour appliquer de façon efficace les techniques de généralisation, vous devez vous familiariser avec votre ensemble de données et ses utilisations prévues, et établir ainsi un équilibre entre l'optimisation de la performance et la conservation du contenu des données nécessaires (p. ex. forme des détails cartographiques et classification).

Les systèmes d'information géographique (SIG) et les bases de données spatiales comprennent des outils qui prennent en charge la généralisation, y compris la classification au moyen de fonctions de regroupement, la simplification au moyen d'algorithmes intégrés (par exemple, [Douglas-Peucker](#)) et la sélection au moyen d'interrogations (par exemple, SQL SELECT). Des explications détaillées sur l'application des techniques de généralisation dépassent la portée du présent document, mais nous invitons le lecteur à consulter le site [GIS Textbooks](#) et la documentation du logiciel sélectionné.

3.1.2 Utilisation d'un système gestion de bases de données spatiales pour une base de données volumineuse ou complexe : Les formats de fichiers de données spatiales, comme ceux popularisés par les grands fournisseurs SIG, fournissent une méthode rapide, pratique et portable de stockage des données spatiales, mais ces systèmes de fichiers peuvent comporter des limites. Les systèmes de fichiers présentent des limites quant à la taille et prennent en charge moins de fonctions inhérentes de gestion et de modélisation des données. Les applications comportant des tables de données volumineuses (soit plus de 1 Go selon le matériel) ou des modèles de données complexes peuvent tirer parti de l'utilisation d'un système de gestion de données relationnelles spatiales, ci-après appelé base de données spatiales. Plusieurs bases de données spatiales sont disponibles aujourd'hui, notamment [Oracle Spatial](#), [ArcGIS](#), [DB2 Spatial Extender](#) et l'extension [PostGIS](#) du système libre bien établi [PostgreSQL](#). Au moyen de stratégies de gestion de bases de données physiques et logiques, les bases de données spatiales peuvent prendre en charge de façon efficace des bases de données très volumineuses. De plus, l'utilisation de langages d'interrogation standard, comme SQL, permet à l'utilisateur d'incorporer des modèles de données généraux

et la logique au niveau des bases de données (plutôt que des applications). Si vous utilisez une base de données volumineuse ou complexe, examinez la possibilité d'utiliser une base de données spatiales pour améliorer la performance et l'efficacité de la maintenance des applications.

3.1.3 Tri et indexation des données : Que vous utilisiez un système de fichiers ou une base de données spatiales, le tri et l'indexation des données peuvent améliorer considérablement le temps de réponse des interrogations de données. Le tri des données peut se révéler efficace pour améliorer les temps d'accès et d'affichage si une application exige d'accéder aux fonctions de façon constante dans un ordre particulier. Par exemple, l'application libre [MapServer](#) contient l'outil [sortshp](#) qui permet de trier un fichier ESRI Shapefile au moyen d'un attribut particulier.

Bien que le tri des données puisse se révéler utile, plusieurs applications exigent plus de flexibilité quant à la façon d'accéder aux données et de les interroger. Dans ce cas, diverses stratégies d'indexation peuvent être appliquées. Les méthodes d'indexation de bases de données standard peuvent être utilisées pour améliorer la performance lorsque des valeurs non spatiales d'attributs sont utilisées pour définir les interrogations. Pour améliorer la rapidité des recherches d'une base de données au moyen d'interrogations spatiales, un index spatial doit être utilisé. Il existe plusieurs méthodes d'indexation spatiale ([Spatial index](#)), dont la méthode R-tree qui est privilégiée par plusieurs applications de bases de données spatiales, y compris l'extension [PostGIS](#). De plus, les données spatiales fondées sur des fichiers peuvent être indexées, par exemple, un fichier ESRI Shapefile peut être indexé au moyen du programme [shptree](#).

3.1.4 Optimisation des données matricielles : Les méthodes de représentation des données matricielles peuvent procurer plusieurs avantages, y compris la représentation efficace de surfaces continues, le traitement algébrique rapide des cartes, l'affichage des données de surface et la représentation efficace des données images. Cependant, les représentations de données matricielles peuvent donner lieu à des ensembles de données très volumineux, en comparaison avec les ensembles de données vectorielles relativement homogènes représentant des phénomènes semblables. Toutefois, certains phénomènes ne sont pas représentés efficacement au moyen du modèle vectoriel et c'est pourquoi, en règle générale, les données matricielles font partie des mises en œuvre contemporaines des OWS.

Si vous utilisez des ensembles volumineux de données matricielles (par exemple, l'imagerie satellitaire à haute définition ou des modèles numériques d'altitude), de nombreuses stratégies peuvent être utilisées pour optimiser la performance. Les deux principales techniques utilisées sont les suivantes : a) les pyramides d'images ([Image Pyramids](#)) et b) la compression d'images ([Image Compression](#)), y compris la compression par ondelettes ([Wavelet Compression](#)). Des outils libres, comme ceux compris dans la distribution [GDAL](#) peuvent être utilisés pour générer des pyramides d'images et convertir ([gdal.translate](#)) l'imagerie non compressée en format compressé. En règle générale, la compression par ondelettes s'effectue au moyen de logiciels propriétaires.

La plupart des communautés d'applications logicielles fournissent des directives précises sur la façon d'optimiser les données matricielles et le niveau d'optimisation (par exemple, [MapServer](#), [GeoServer](#)).

3.2 Optimisation au niveau des serveurs

La plupart des « superpositions » technologiques des OWS comprennent au moins deux serveurs, soit un serveur propre aux OWS (par exemple, WMS, WFS) et un serveur Web au moyen duquel les demandes et les réponses HTTP sont gérées. Un serveur de base de données peut également être utilisé pour assurer le soutien de l'application. La présente section traite de l'optimisation des serveurs Web et des serveurs de bases de données tandis que la section suivante porte sur l'optimisation des serveurs OWS.

Tel qu'il a été indiqué précédemment, un serveur Web est généralement utilisé pour gérer les demandes et les réponses HTTP en relation avec l'OWS que vous hébergez ou que vous utilisez dans votre application. Si vous êtes un utilisateur du service, nous vous recommandons de discuter des questions liées à la performance du serveur Web avec votre fournisseur de services. Si vous hébergez le service, nous vous recommandons d'effectuer une recherche des meilleures pratiques pour optimiser la performance de votre serveur Web. De nombreux documents et de nombreuses ressources Web contenant des conseils sur

l'optimisation du serveur Web sont disponibles. Par exemple, la documentation standard du populaire serveur libre Apache Web contient un document sur l'optimisation de la performance ([Apache Performance Tuning](#)).

Les serveurs de bases de données peuvent être mis au point de plusieurs façons pour optimiser la performance, par exemple, comme nous l'avons souligné précédemment, au moyen de l'indexation au niveau des bases de données. Il est également possible d'optimiser la performance au niveau des serveurs de bases de données. Ceci comprend notamment l'optimisation de la configuration des dispositifs de stockage (par exemple, le partitionnement des disques durs), la définition du « format de la page » utilisé pour le stockage des données (par exemple, [PostgreSQL Method](#)), la gestion efficace de la mémoire et les tâches régulières de maintenance des bases de données. Chaque grande base de données spatiales comporte ses propres approches qui doivent être documentées par le fournisseur, l'utilisateur et la communauté des éditeurs. Par exemple, le projet PostgreSQL fournit un document sur la configuration des serveurs ([Server Configuration](#)) qui traite de nombreuses questions liées à l'optimisation des serveurs.

Il est important de souligner que l'optimisation au niveau des serveurs repose sur l'optimisation des logiciels et du matériel. Le temps et le travail consacrés à l'optimisation au moyen de la configuration des logiciels peuvent ne donner aucun résultat si le serveur s'exécute sur du matériel inférieur aux normes (par exemple, mémoire vive (RAM) insuffisante, unité centrale (UC) ou lecteur de disque dur trop lent). Si vous utilisez un service dans votre application, il pourrait être utile d'en discuter avec votre fournisseur de services. De la même façon, si vous hébergez un service, directement ou par l'intermédiaire d'un service d'hébergement, une bonne compréhension de la capacité de votre système au niveau des serveurs pourrait être utile pour l'évaluation de la performance globale.

3.3 Optimisation au niveau des OWS

Tout comme pour les serveurs Web et de bases de données, les applications serveur des OWS peuvent être optimisées de différentes façons. Les techniques d'optimisation sont propres à chaque mise en œuvre logicielle et aux OWS concernés. Toutefois, dans plusieurs cas, plusieurs approches générales sont utiles.

3.3.1 Utilisation de profils : Dans le contexte des OWS, un profil est réservé à un sous-ensemble d'un langage XML (par exemple, GML) qui limite le nombre de types d'objets qui peuvent figurer dans un schéma conforme. Ceci a pour résultat des instances de documents XML plus faciles à traiter. Un profil bien connu dans la communauté de l'OGC est le Geography Markup Language Simple Feature Profile (OGC 2006). L'utilisation d'un tel profil peut améliorer la performance d'un service (complexité réduite) et, potentiellement, augmenter le volume des données de réponse en raison de structures de données plus simples.

3.3.2 Utilisation d'un autre codage des données : Par défaut, les OWS comme WFS et CSW produisent des documents XML en format texte. Bien que ces documents soient d'une simplicité élégante, les flux de données résultants peuvent être très volumineux compte tenu de la nature relativement verbeuse de XML. Pour éviter de perdre la puissance expressive de XML, la compression peut être utilisée pour réduire la taille des données XML au moment de leur transmission. Cette méthode est bien documentée pour WFS dans une étude de 2005 réalisée par des membres de l'OGC (OGC 2005). Si vous traitez de grands volumes de données XML et que votre logiciel serveur prend en charge un autre codage de données, la performance pourrait s'en trouver améliorée.

Le même raisonnement peut s'appliquer à l'utilisation de WMS. Un WMS peut produire une vaste gamme de formats d'images, y compris des formats non compressés comme TIFF. À moins d'un motif impérieux d'utiliser des formats non compressés, nous proposons d'utiliser ou de demander un format compressé comme JPEG ou PNG pour réduire la consommation de bande passante et le temps de réponse respectivement.

3.3.3 WMS – Antémémoire et mosaïque : Dans le cas où les données sont relativement statiques, la mise en antémémoire côté serveur des mosaïques WMS est une façon très efficace d'améliorer la performance des services. Au moment de la réception d'une demande, le serveur répond au moyen d'un résultat en mosaïque prérendu, réduisant ainsi de façon importante le temps de traitement requis pour donner suite à la demande. Un bon exemple d'une application de mise en mémoire de mosaïques WMS est [GeoWebCache](#). De plus,

l'OGC a publié dernièrement un article instructif sur ce sujet intitulé *OpenGIS Tiled WMS Discussion Paper* (OGC 2009c).

Il existe plusieurs autres approches d'optimisation des OWS au niveau des serveurs. Par exemple, [Anderson & Deoliveira](#) (2009) présentent des recommandations très utiles pour optimiser MapServer et GeoServer. Le site du projet GeoServer présente un dossier complet sur les techniques d'optimisation ([Optimization Techniques](#)), tout comme le site de [MapServer](#). D'autres ressources, comme le rapport *OGC Canadian Geospatial Data Infrastructure WFS and GML Best Practices* (OGC 2008), présentent les enjeux et des recommandations sur les défis reliés à la mise en œuvre et la performance des OWS. L'examen des ressources présentées ici à titre de références ainsi que la documentation logicielle et d'autres ressources pertinentes vous fournira une base solide pour établir une stratégie d'optimisation au niveau des OWS.

3.4 Optimisation au niveau des réseaux

Le temps d'attente peut se définir comme le temps qui s'écoule entre le moment où un traitement est lancé et le moment où l'un de ses effets est perçu. Le temps d'attente est un concept important dans un environnement de services Web où la performance d'une application est fonction des demandes et des réponses résultantes transmises sur un réseau de communication (soit Internet). L'optimisation d'un OWS aux niveaux des bases de données, des serveurs et des serveurs OWS peut aboutir à une performance toujours relativement piètre des services (et, par voie de conséquence, des applications dépendantes) si l'on ne tient pas compte de tous les facteurs en jeu, comme la bande passante moyenne disponible du réseau. Certes, la bande passante nécessaire est propre au contexte des applications, mais les applications de données spatiales sont généralement exigeantes en bande passante. Les WMS produisent des données image relativement volumineuses tandis que les services, CSW, GeoRSS, KML etc. produisent des documents XML verbeux. En plus d'optimiser la performance aux autres niveaux de la superposition, vous devez vous assurer que la bande passante disponible au niveau des réseaux est suffisante pour répondre aux besoins de l'application. Ceci s'applique aux fournisseurs de services, aux développeurs d'applications et aux utilisateurs finaux des applications.

3.5 Optimisation côté client

Dans son document *High Performance Web Sites* (Souders 2009), Steve Souders indique que 10 % à 20 % seulement du temps de réponse des utilisateurs finaux est consacré au téléchargement du document HTML, alors que 80 % à 90 % du temps restant est consacré au téléchargement de toutes les composantes dans la page (p. 5). Il s'agit d'un point important dont il faut tenir compte pour les applications spatiales Web qui sont, en règle générale, riches en composantes (par exemple, des images cartographiques) et qui sont de plus en plus développées au moyen d'approches de développement côté client (JavaScript, AJAX, etc.). C'est pourquoi les développeurs et les utilisateurs doivent examiner sérieusement l'impact de la conception et de l'optimisation côté client. En plus de ce document indiqué à titre de référence, vous pouvez prendre connaissance des nombreuses recommandations de Souders en ligne ([Best Practices for Speeding Up your Web Site](#)). Veuillez noter que certaines recommandations, comme « Gzip Components », nous sont déjà connues. Le développement Web ne cessant d'évoluer, il est de plus en plus important de tenir compte de la composante côté client pour la performance globale.

Sommaire : Optimisation de la performance des services

L'amélioration des services peut être le résultat d'un ou de deux changements ayant un impact majeur ou la combinaison de plusieurs changements ayant des impacts mineurs cumulatifs. Certaines stratégies sont applicables aux systèmes logiciels tandis que d'autres sont propres à un environnement logiciel, matériel ou de services en particulier. Les utilisateurs de services Web sont invités à définir de la façon la plus détaillée possible les services utilisés dans une application donnée. L'utilisation des renseignements présentés ici comme base de recherche sur la performance logicielle peut contribuer à l'élaboration de commentaires constructifs pour un fournisseur des services.

Références

- OGC (2005) *OWS 3 GML Investigations - Performance Experiment by Galdos Systems*. Document 05-101 de l'Open Geospatial Consortium, accessible aux fins de téléchargement à l'adresse suivante : <http://www.opengeospatial.org/standards/gml>. Dernier accès : le 10 mars 2009.
- OGC (2006) *GML 3.1.1 simple features profile*. Document 06-049r1 de l'Open Geospatial Consortium, accessible aux fins de téléchargement à l'adresse suivante : <http://www.opengeospatial.org/standards/profile>. Dernier accès : le 10 mars 2009.
- Anderson & Deoliveira (2009) *WMS Performance: Mapserver vs. Geoserver*. Présentation effectuée dans le cadre de la Free and Open Source Software for Geospatial 2009 Conference, Victoria, Colombie-Britannique, 24 au 27 septembre 2009. Accessible à l'adresse suivante : http://www.foss4g2009.org/presentations/view.php?abstract_id=120. Dernier accès : le 17 mars 2009.
- OGC (2009a) *OGC Compliance Testing Portal*. Ressource en ligne accessible à l'adresse suivante : <http://cite.opengeospatial.org/instructions>. Dernier accès : le 12 février 2009.
- OGC (2009b) *Reference Implementations*. Ressource en ligne accessible à l'adresse suivante : <http://cite.opengeospatial.org/reference>. Dernier accès : le 12 février 2009.
- OGC (2009c) *OpenGIS Tiled WMS Discussion Paper*. Document 07-057r2 de l'Open Geospatial Consortium, accessible aux fins de téléchargement à l'adresse suivante : <http://www.opengeospatial.org/standards/dpd>. Dernier accès : le 10 mars 2009.
- Souders, S. (2009) *High Performance Web Sites: Essential Knowledge for Front-End Engineers*. Sebastapol, CA: O'Reilly Media Inc. 168 p.
- OGC (2008) *OGC® Canadian Geospatial Data Infrastructure WFS and GML Best Practices*. Document 08-002 de l'Open Geospatial Consortium, accessible aux fins de téléchargement à l'adresse suivante : <http://www.opengeospatial.org/standards/dpd>. Dernier accès : le 10 mars 2009.