

WHITE PAPER

Démantèlement des barrières relationnelles: la gestion des données utilisateur triomphe avec Caché d'InterSystems

Commandité par : InterSystems Corporation

Carl W. Olofson

Février 2003

OPINION D'IDC

Les systèmes de gestion de bases de données relationnelles (SGBDR) sont conçus pour stocker les données selon le mode de catalogage de données le plus performant, qui correspond à celui défini par la théorie des ensembles mathématiques tel qu'il est exprimé dans le paradigme relationnel. Dans de nombreux cas, cependant, le mode de catalogage de données le plus performant ne correspond pas au mode de stockage et d'extraction de données le plus performant. Les bases de données relationnelles fonctionnent parfaitement dans les situations où les données sont gérées de la manière la plus appropriée, c'est-à-dire comme listes plates contenant des types de données simples et mettant en jeu peu d'associations avec les données figurant dans d'autres listes. Lorsque les données à traiter doivent être conservées dans des structures interdépendantes complexes ou lorsqu'il faut extraire les données rapidement en suivant des chemins constitués d'associations au lieu de parcourir de simples listes, les bases de données relationnelles révèlent alors quelques caractéristiques telles que la gestion d'index multiples ainsi que des structures de schémas normalisées complexes et transversales. Ces obstacles, ainsi que les contraintes liées à la longueur des lignes et à la taille des tables, peuvent dans certains cas représenter de lourds handicaps au point d'amener les utilisateurs à considérer un SGBDR comme difficilement applicable à certaines tâches de gestion de données.

Bien que de grands éditeurs de SGBDR ont introduit récemment des fonctionnalités qui permettent à leurs produits de prendre en charge les données situées en dehors du paradigme relationnel, IDC estime que les moyens fondamentaux de gestion des données et l'accès à ces données restent, pour l'essentiel, de nature relationnelle et dépendants de SQL. De ce fait, les SGBDR demeureront des produits inutilement difficiles à configurer et à gérer, et trop peu performants pour certains types de bases de données. À la lumière des différentes expériences des utilisateurs relatées dans ce livre blanc, IDC trouve que Caché offre une excellente alternative dans de nombreux cas.

MÉTHODOLOGIE

IDC a étudié un certain nombre de cas dans le cadre de la préparation de ce document dont les résultats se sont avérés tous relativement homogènes. IDC en a cependant retenu trois à titre d'illustration. Ces cas d'étude, ainsi que des informations de référence sur d'autres cas ont été obtenus par l'intermédiaire d'InterSystems, mais IDC a effectué des interviews sans le concours d'InterSystems et sans lui demander son consentement préalable pour les questions à poser ou les sujets à aborder. Pour les descriptions, les opinions et les conclusions concernant Caché et ses atouts comparatifs par rapport aux SGBDR, IDC s'appuie sur ces cas ainsi que sur d'autres expériences rapportées par les utilisateurs et dont IDC a eu connaissance de façon indépendante.

DANS CE LIVRE BLANC

IDC a comparé les caractéristiques du modèle Caché à ceux des principaux systèmes de gestion de bases de données relationnelles et a passé en revue de nombreux cas d'étude ainsi que des exemples de tests et d'utilisation de Caché, dont plusieurs sont exposés dans ce document.

Le présent document reprend ces informations pour expliquer les différences pratiques clés entre l'utilisation de bases de données relationnelles et l'approche base de données post-relationnelle de Caché. Ce document étudie également les types de bases de données et les contextes d'application dans lesquels l'approche de Caché apparaît supérieure à celle d'un SGBDR.

VUE D'ENSEMBLE DE LA SITUATION

Depuis le début des années 1980, le postulat dans le mode informatique est que tous les problèmes de gestion de données trouvent leur solution dans les bases de données relationnelles. Cette idée a été développée par le Dr E. F. Codd ; elle a ensuite été reprise et largement répandue par IBM et de nombreux universitaires. Toute une industrie s'est construite autour de cette idée, donnant lieu ainsi à la naissance de grandes sociétés de SGBDR, au premier rang desquelles figurent Oracle et Sybase, et à des offres de SGBDR essentielles de la part d'IBM (DB2) et de Microsoft (Microsoft SQL Server), entre autres.

Dans la mesure où la plupart des problèmes de gestion de données qui étaient présentés de cette façon concernaient les opérations d'archivage de base des entreprises, qui impliquent des ensembles de listes plates relativement simples (à l'exemple des comptes, des transactions, des stocks et des commandes), l'espoir de voir la méthode relationnelle représenter la base de données universelle semblait se confirmer. À mesure que les utilisateurs essayaient de gérer d'autres types de données, tels que les données de nomenclature (qui sont répétitives par nature), les structures de listes intégrées (les adresses de publipostage et d'expédition en sont des exemples simples) et les données régies par nombreuses interdépendances complexes (comme les structures des entrepôts de données ainsi que les structures des données de gestion de l'ingénierie, des pièces de rechange et des réseaux), l'utilité des SGBDR a vite montré ses limites.

LIMITATIONS DU PARADIGME RELATIONNEL

Le paradigme relationnel de gestion de données repose sur la théorie des ensembles mathématiques, qui suppose une structure régulière ou « normalisée » de types de faits organisés sous la forme de listes, ou d'ensembles, simples de valeurs pouvant ou non avoir un rapport entre eux. Cette approche représente une méthode raisonnable pour cataloguer les valeurs et les lister selon la façon dont elles ont été classifiées et collectées. Elle fonctionne moins bien lorsque les valeurs concernées doivent être collectées, manipulées et présentées sous une forme autre que celle d'une liste et quand la classification des faits devient ambiguë, arbitraire ou dépendante de la situation.

Dans le monde réel de l'entreprise, une grande partie des données est de cette nature.

LISTES PLATES

Le paradigme relationnel suppose que toutes les données doivent pouvoir être gérées sous forme de listes et que lorsque les relations entre les données impliquent une structure plus élaborée que celle des listes tabulaires à deux dimensions, les relations entre ces listes, ou tables, utilisant des clés étrangères devraient alors être suffisantes. Cependant, il arrive souvent que les données soient plus désordonnées

que cela. Des instances de faits concernant un objet métier, par exemple, peuvent ou non mettre en jeu les mêmes attributs que d'autres du même type. Des structures telles que les tableaux multidimensionnels ou des tables dans les tables peuvent être sollicitées. Elles peuvent être gérées, mais uniquement par la création d'autres tables et de relations avec des clés étrangères.

INSUFFISANCES ASSOCIÉES AUX INDEX MULTIPLES

Les bases de données contenant plusieurs niveaux de tables liées par des relations de clés étrangères peuvent imposer le parcours de nombreux index de tables avant de pouvoir reconstituer un ensemble de faits relatif à un sujet particulier. Les structures qui sont imbriquées ou de type multidimensionnel, ou celles qui représentent des collections étroitement liées et raisonnablement simples et qui sont susceptibles, sous d'autres méthodes d'organisation de données, d'être extraites avec un ou deux accès d'E/S, peuvent impliquer des recherches dans plusieurs tables et index, des comparaisons et des boucles incessantes pour construire un ensemble de résultats approprié.

PRISE EN CHARGE DE GROS OBJETS

Le paradigme relationnel n'a pas prévu la prise en charge d'objets volumineux. Des objets par exemple de type texte, image, audio, vidéo et géographique (parfois appelés « Binary Large Objects », ou BLOB) ne rentrent pas dans ce paradigme. En termes relationnels, ils sont parfois considérés comme des étendues de bits sans aucune signification pour la base de données, laquelle comprend uniquement les nombres, les chaînes de caractères et les valeurs logiques. Certains éditeurs ont greffé des extensions sur leurs produits SGBDR et fournissent même à leurs utilisateurs et partenaires le moyen d'intégrer la prise en charge de la recherche et l'extraction de tels objets, mais il s'agit de compromis inconfortables d'un système qui ne les incorpore pas dès la phase de conception.

TYPES DE DONNÉES INCORRECTEMENT PRISES EN CHARGE PAR LES SYSTÈMES RELATIONNELS

Outre les BLOB, le paradigme relationnel ne peut pas gérer efficacement ni prendre en charge correctement toute une variété de structures de données moins ésotériques, qui sont détaillées dans les sections suivantes.

STRUCTURES RÉCURSIVES ET IMBRIQUÉES

Le paradigme relationnel aime les tables qui renvoient à d'autres tables. Il n'est pas très à l'aise avec les tables qui ont des lignes qui renvoient, directement ou indirectement, à d'autres lignes dans la même table.

Ainsi, les structures récursives telles qu'une nomenclature ne peuvent pas être prises en charge directement ; pas plus que le concept selon lequel un employé peut diriger d'autres employés et rendre lui-même compte à un autre employé. Même une imbrication simple, telle que la création d'une adresse dans un enregistrement d'expédition, ne peut pas être effectuée sans violation des règles de la normalisation. Il faut, en effet, créer une table d'« adresses » artificielles pour se conformer à la lettre de la loi. Si des adresses identiques d'un point de vue structurel sont utilisées à d'autres fins, comme la facturation d'un client ou la transmission d'une commande à un fournisseur, il faut alors les placer dans une table (la normalisation exige qu'un ensemble récurrent d'attributs définisse une table et ne se répète pas dans d'autres tables) et ne pas mettre une clé étrangère dans la table ; autrement dit, il faut créer un ID d'adresse artificiel interne auquel les lignes de facturation, d'expédition et d'autres lignes de la table peuvent faire référence.

Cette approche serait en violation d'une autre règle, celle qui proscriit la création d'ID artificiels.

Bien entendu, les administrateurs de bases de données violent en réalité les règles de normalisation tout le temps ; autrement, ils se retrouveraient avec des bases de données horriblement compliquées et peu performantes.

RÉSEAUX DE DONNÉES EN CORRÉLATION

Lorsque les données apparaissent dans des réseaux de structures connexes, telles que celles qui sont requises pour représenter des commandes et des factures, et sont décomposées en ensembles de tables pour le stockage et l'extraction relationnels, il faut parfois des connaissances approfondies en matière de bases de données pour déterminer les tables à interroger et savoir comment combiner les résultats obtenus. Par exemple, pour reconstruire une commande, il faut peut-être interroger la commande, l'article de la commande, le produit, le client et les tables de stock. Rien dans les définitions des tables ne permet de simplifier ce processus ; il faut que l'utilisateur le sache.

Lorsque les données connexes incluent des objets qui ne s'intègrent pas dans les constructions tabulaires, comme le fait d'associer les tickets problème aux messages électroniques de clients mécontents, cela devient un exercice qui sort du cadre du domaine relationnel.

COLLECTIONS DE DONNÉES RELATIVES À DES TYPES D'OBJETS SIMILAIRES MAIS PAS IDENTIQUES

Le paradigme relationnel aime que tous les objets d'un même type possèdent les mêmes attributs. Dans le monde réel, cependant, certains objets varient légèrement dans leur attribution. Nous pouvons traiter ce cas à travers une hiérarchie de types qui nous permet de décrire les variations en cascade de types basiques (par exemple, véhicules automobiles / automobiles / minibus). Le paradigme relationnel force l'utilisation de tables connexes ayant les mêmes variantes d'attributs, en imposant des vues contenant plusieurs niveaux de commandes SELECT imbriquées, même dans le cas de rapport d'objet relativement simples.

SOLUTION DE RECHANGE DU MODÈLE RELATIONNEL

Compte tenu des limitations décrites ci-dessus, diverses solutions de rechange ont été proposées pour le modèle relationnel. Les deux plus courantes sont les SGBD orientés objet (ODBMS) et les SGBD post-relationnels (PDBMS).

SGBD ORIENTÉS OBJET

Les SGBD orientés objet (ODBMS) prennent en charge le stockage de données comme les états, ou les valeurs des propriétés, des objets définis conformément à une structure de classe (généralement appelés modèle objet). Cette structure prend en charge les objets imbriqués, en définissant les objets par classe selon une hiérarchie de classe, et le polymorphisme, ce qui signifie que les opérations (ou « méthodes ») d'un objet et ses propriétés peuvent varier en terme de comportement et de type selon le contexte dans lequel ils sont utilisés, lequel est généralement déterminé par la forme de la liste des paramètres présentée à une méthode. (Cette forme est appelée sa « signature ».) L'avantage de cette approche c'est qu'elle offre une souplesse presque infinie pour la définition et le stockage de données, tel que le fonctionnement de la base de données est entrelacé de façon transparente avec le fonctionnement de l'application, sans requérir une interface de programmation d'application (API) formelle. L'inconvénient c'est que ce résultat est atteint en rendant le modèle objet de l'application identique à celui de la base de données ; cela veut dire que la base de données change en même temps que l'application et donc qu'une base de données ne peut pas être partagée par différentes applications. Ce fait, combiné aux interdépendances typologiques rigides des classes en raison de leurs hiérarchies de classes, fait que les modifications effectuées dans les structures des objets produisent des effets profonds en termes d'effort de maintenance en ce qui concerne aussi bien l'application que la base de données.

SGBD POST-RELATIONNELS

Avant que le paradigme relationnel ne devienne la force dominante, de puissants SGBD pré-relationnels prenaient en charge le stockage des données dans des tables imbriquées ou d'autres structures et incluaient parfois une gestion de listes optimisée et des réseaux de données. Ces SGBD se caractérisaient par une sémantique élaborée qui était destinée à leur permettre d'appliquer les règles des données en fonction de leur sémantique. Cette sémantique a été abandonnée par le monde relationnel, qui ne s'intéressait qu'aux listes plates et à leurs associations. Les SGBD post-relationnels, qui ont souvent été développés à partir d'une technologie pré-relationnelle, regroupent toutes les fonctionnalités décrites ci-dessous en y ajoutant également la prise en charge d'objets complexes. La technologie des SGBD post-relationnels ajoute également des services orientés objet et l'aptitude d'écrire des scripts, bien qu'elle reste distincte de l'application par une API, permettant ainsi un développement et une maintenance séparés des applications et de leurs bases de données. Ils offrent donc les atouts des SGBD orientés objet tout en évitant leurs pièges.

LE SGBD POST-RELATIONNEL CACHÉ D'INTERSYSTEMS

InterSystems propose un SGBD post-relationnel appelé Caché. À l'instar d'autres SGBD post-relationnels, celui-ci a été développé par des ingénieurs qui avaient une grande expérience dans la création de technologie SGBD offrant une vitesse et une souplesse sensiblement supérieures à celles des systèmes relationnels. InterSystems a encapsulé ces fonctionnalités dans un système orienté objet pour la définition et la gestion de données ainsi que la prestation de services relatifs aux données. Ce document ne contient pas une description complète des fonctionnalités et atouts de Caché, mais plutôt une présentation générale de ses points forts sur la base des entretiens menés par IDC avec de nombreux utilisateurs de Caché, et notamment les cas d'étude exposés ci-après.

PRISE EN CHARGE DES DONNÉES MULTIDIMENSIONNELLES

À la différence des SGBD relationnels, qui forcent toutes les données dans des tables connexes à deux dimensions, d'où la nécessité d'effectuer des recherches dans de multiples index et, parfois, de parcourir des tables pour poser des questions sur des données étroitement liées, Caché gère les structures multidimensionnelles grâce à l'indexation multidimensionnelle. Cette fonctionnalité a été mentionnée à plusieurs reprises en tant qu'avantage clé en termes de performance dont dispose Caché par rapport aux systèmes relationnels.

APPROCHE ORIENTÉE OBJET AVEC PRISE EN CHARGE TOTALE DE SQL

Les fonctionnalités orientées objet du produit permettent aux utilisateurs de construire des définitions de données basées sur un modèle objet présentant des hiérarchies de classe, l'imbrication, le polymorphisme, etc. Ces objets sont accessibles via une API de services orientée objet mais n'imposent pas un modèle objet au programme d'application. Dans le même temps, Caché assure une prise en charge totale de SQL, ce qui fait que les outils d'interrogation basés sur SQL et les programmes externes qui exigent SQL peuvent facilement accéder aux données. Cette prise en charge de SQL ne joue pas le rôle de « parent pauvre » par rapport à l'API orientée objet et n'enregistre pas de baisse de performances de ce fait.

PAS D' « ERREUR D'IMPÉDANCE » AVEC CACHÉ OBJECTSCRIPT

Les applications orientées objet, telles que celles qui peuvent être écrites en langage Java ou C++, trébuchent souvent ou rencontrent une « erreur d'impédance » lorsqu'elles essaient d'agir sur une base de données relationnelle, car une couche de traitement est nécessaire pour faire entrer des structures plates dans les tables relationnelles ou pour les en faire sortir. Cette erreur d'impédance n'existe pas dans

le cas de Caché non seulement parce que celui-ci peut prendre en charge des structures complexes, mais également parce qu'en raison de l'utilisation de Caché ObjectScript, le langage d'écriture de scripts orienté objet de Caché, des objets de données de Caché peuvent avoir le genre de caractéristiques de comportement qui leur permet de se fondre dans une approche de programmation orientée objet. Caché ObjectScript est également considéré par les clients contactés comme un moyen d'intégrer de l'intelligence et de l'automatisation dans la gestion de données sur le serveur de bases de données, ce qui permet d'avoir des applications plus simples, d'assurer l'intégrité des données et de bénéficier d'une efficacité opérationnelle plus grande.

DÉFIS/OPPORTUNITÉS

Bien que Caché ait beaucoup de qualités, IDC note qu'InterSystems doit vaincre à la fois le parti pris relationnel du marché et le handicap que constituent sa petite taille et son manque de notoriété. Ce sont là des motifs de préoccupation pour les clients existants et potentiels.

Néanmoins, la longévité et la stabilité de la compagnie au fil des ans lui permettront de vaincre ces réticences une fois que les clients seront familiarisés avec l'organisation mise en place.

IDC remarque également que l'accent est de plus en plus mis sur la nécessité d'intégrer les applications ainsi que sur la gestion de combinaisons de données structurées et non structurées. IDC relève par ailleurs l'utilisation grandissante du langage XML pour l'intégration des applications et les interactions inter-entreprise, notamment dans l'évolution des services Web.

Caché, surtout dans la mesure où ce produit développe une prise en charge plus directe du langage XML, se tient prêt à exploiter toutes les opportunités que représentent les services Web et à tirer parti de toutes les futures initiatives dans le domaine du calcul distribué.

CONCLUSION

Selon IDC, le paradigme relationnel continuera d'être la force dominante en matière d'archivage de base pour les entreprises. Lorsque la gestion des données sort des limites de ces fonctions informatiques classiques d'entreprise, elle commence à s'écrouler. Malgré toutes les greffes de déclencheurs et de procédures stockées, d'extensions d'objet, de prise en charge de BLOB et récemment d'une certaine prise en charge de XML (tous ces éléments étant destinés à compenser les insuffisances inhérentes au modèle relationnel), les SGBDR se révèlent inadéquats pour les nouvelles classes d'applications qui requièrent un accès rapide aux structures de données complexes, la prise en charge d'analytique commerciale en temps réel et les données qui fonctionnent avec des systèmes effectuant des modifications réelles dans les opérations tout au long de la journée.

D'après IDC, Caché représente une puissante solution de remplacement pour les entreprises qui ont ces besoins. Les cas d'étude qui suivent illustrent les avantages que l'indexation multidimensionnelle, les scripts côté serveur orientés objet ainsi que l'insertion et l'extraction à haute vitesse de données complexes offrent aux scénarios critiques et réels des entreprises. Sur la base des discussions que IDC a eues avec les utilisateurs de Caché, ces exemples sont loin d'être inhabituels et illustrent parfaitement les caractéristiques qui ont permis à InterSystems de se constituer une communauté loyale et profondément engagée d'utilisateurs qui savent qu'aucune base de données relationnelle ne peut répondre à leurs besoins critiques de la façon dont le fait Caché.

CAS D'ÉTUDE

A M E R I T R A D E

Ameritrade, dont le siège se trouve à Bellevue (dans le Nebraska), utilise Caché pour son système de compensation, qui fonctionne sur un cluster Sun ES10000 haute disponibilité et gère 250 Go de données sur une baie de stockage EMC. Le système permet de « compenser », ou de vérifier de façon absolue, les transactions réalisées toute la journée, tous les jours. Aucune transaction ne peut être considérée comme étant terminée tant qu'elle n'a pas été publiée dans la base de données Caché et validée correctement. Ce système communique avec les autres systèmes associés et gère 2,8 millions de comptes.

La nuit, il compense des millions de transactions qui ont été regroupées par lots, à partir de 18:00 heures (heure de la côte Est) ; le traitement du batch doit être terminé correctement au plus tard à 04:00 heures (heure de la côte Est).

Chaque transaction menée dans la base de données requiert une série de lectures complexes puis une opération d'écriture qui met à jour un certain nombre de champs et de tables. Ameritrade attribue le mérite du haut débit assuré par Caché à sa capacité à représenter les structures de données complexes utilisées dans les tableaux multidimensionnels, ce qui simplifie grandement les opérations de lecture et d'écriture. Le système gère actuellement plus de 900 utilisateurs simultanés pendant les heures de courtage, chacun d'eux effectuant ces transactions complexes à un très haut débit et ne pouvant pas tolérer des retards, comme ceux causés par un temps de réponse lent ou une annulation de transaction suivie d'un redémarrage (une opération qu'exigent certains systèmes pour fonctionner correctement compte tenu de conflits de ressources ou d'interblocages). Ameritrade a révélé à IDC qu'elle a essayé d'utiliser un grand nom du SGBDR pour ce système il y a cinq ans avant d'abandonner car « il ne pouvait pas suivre ».

Outre la base de données de production d'un demi-téraoctet, Ameritrade utilise une autre base de données d'édition d'un demi-téraoctet et une base de données historique de 190 Go, toutes sur Caché. La société reconnaît que Caché ObjectScript permet de simplifier grandement l'exécution de ses transactions complexes, et gère les relations entre ces trois bases de données en programmant de l'intelligence dans les bases de données elles-mêmes. Dans la mesure où Caché ObjectScript joue un rôle clé dans le traitement du système, une mise à niveau récente de la dernière version de Caché a offert un bonus à Ameritrade : un gain de 15,20% dans le temps nécessaire pour traiter certains travaux par lots.

Ameritrade considère comme une fonctionnalité clé la prise en charge en continue de XML par Caché.

Ameritrade reconnaît également que Caché ObjectScript simplifie le développement et le déploiement des applications, ce qui permet à la compagnie de tout réussir du premier coup. De ce fait, tous les travaux de développement actuels portent presque exclusivement sur l'ajout d'une nouvelle fonctionnalité ou l'amélioration de fonctionnalités existantes et non pas sur la correction de bogues ou le remaniement du code.

ANALYSE

Par rapport à la technologie SGBDR, Caché semble se distinguer en l'occurrence par les avantages clés suivants:

- La prise en charge de tableaux multidimensionnels, ce qui permet à des structures de données plus simples de gérer des données qui exigeraient autrement un grand nombre de tables connexes dans un environnement relationnel.

- ☒ Sa capacité à maintenir une structure de données orientée objet neutre par rapport aux applications, ce qui permet d'utiliser un code ObjectScript simple et propre.
- ☒ Son coût total de possession, qui dérive non seulement d'une redevance de licence et de maintenance moindre par rapport à celle des produits SGBDR pour des installations de taille équivalente, mais également de l'utilisation performante du système et de la simplicité de gestion, qui permettent tous deux de réaliser des économies au niveau des coûts du système et du personnel par rapport aux principaux produits SGBDR du marché.

Comme cela a été le cas avec chaque utilisateur de Caché avec lequel nous avons pu nous entretenir, Ameritrade n'était pas avare d'éloges au sujet du professionnalisme et de la structure de remontée de l'information, parfaitement organisée, du support technique d'InterSystems.

M E R A L C O

Meralco est une compagnie d'électricité opérant aux Philippines qui gère un énorme entrepôt de données (data warehouse) à l'aide d'un grand produit SGBDR. Soucieuse de réduire le coût de gestion de son entrepôt de données, la compagnie a fait appel à Digital Dimensions, une firme locale affiliée à InterSystems, en vue de réaliser un test portant sur la définition de bases de données d'entrepôt de données identiques sur Caché et sur une instance du SGBDR ainsi que le chargement d'une partie des données utilisées dans l'entrepôt de données de production dans chacune d'elles. C'était la première étape d'un processus d'évaluation qui est toujours en cours afin de déterminer si Meralco transférera ou non son entrepôt de données sur Caché. Le but de ce test est de déterminer si Caché est susceptible d'améliorer la durée de chargement par rapport au SGBDR.

Pour réaliser le test dans des conditions équivalentes, les ingénieurs ont dû réécrire les procédures stockées du SGBDR en Caché ObjectScript. C'est la phase du projet qui a demandé le plus de temps car il s'agissait de passer d'un paradigme relationnel à un paradigme objet pour la gestion de données. Le système de test, qui était utilisé à la fois pour la base de données Caché et la base de données du SGBDR, était un système IBM NetFinity 7000 M10 SMP équipé de deux processeurs Intel 450 MHz, de 1 Go de mémoire RAM et d'un disque de 300 Go (y compris les fichiers autres que ceux de la base de données) sur quatre disques durs.

Lors de la conduite du test, les ingénieurs ont noté que le SGBDR exigeait le chargement de l'une des tables brutes (un total de 13 396 510 enregistrements) puis son filtrage (pour la ramener à 90 349 lignes), Caché filtrait les données au fur et à mesure qu'il les chargeait, ne demandant aucune étape distincte et permettant de gagner 1 heure et 37 minutes en durée. Par ailleurs, le SGBDR chargeait les données puis reconstruisait les index, alors que Caché ajustait dynamiquement les index au fur et à mesure du chargement, permettant ainsi des gains de temps extraordinaires au niveau du chargement. Une fois que le SGBDR a chargé puis filtré les données, l'étape suivante consistait à générer les cubes multidimensionnels. Dans la mesure où les données de Caché sont multidimensionnelles par nature, cette étape était inutile, d'où de nouveau un gain de 2 heures.

Il convient de relever que ce test a été effectué à l'aide d'une version récente et non de la toute dernière version du produit SGBDR. Les représentants de Meralco ont suggéré qu'avec la toute dernière version, les résultats auraient peut-être été moins spectaculaires. Cependant, le filtrage et le chargement des données de test sous la version actuelle du SGBDR utilisée par Meralco ont pris 26 heures et 9 minutes, alors qu'il n'a fallu que 4 heures et 46 minutes avec Caché. Le rapport de performance obtenu entre Caché et le SGBDR atteint ainsi presque 6:1.

ANALYSE

Le temps de chargement est l'un des critères clés à prendre en compte dans la gestion d'un entrepôt de données en raison de ses implications en termes de coûts du personnel et du système. Meralco exécute son entrepôt de données sur un serveur Sun qui est beaucoup plus puissant que le système de test Intel utilisé aux fins de la présente comparaison. Compte tenu des résultats de ce test, il existe de fortes probabilités qu'en adoptant Caché, Meralco pourrait déployer son entrepôt de données sur des systèmes moins coûteux à l'avenir. D'autre part, à la lumière du rapport de performance, Meralco pourrait s'attendre à faire des économies en déployant son entrepôt de données avec moins de ressources système en utilisant Caché.

IDC trouve que ces résultats démontrent incontestablement que Caché est beaucoup plus apte à gérer de hauts volumes de données complexes qu'un système de base de données relationnel. D'autre part, la fonction de gestion d'index souple de Caché semble suggérer que ces bénéfices se traduiront en avantages croissants en termes d'efficacité pendant le chargement des données au fil du temps car les entrepôts de données ne font généralement que se développer.

SCHLUMBERGER SEMA

SchlumbergerSema est l'un des principaux fournisseurs mondiaux de systèmes SMS aux opérateurs de réseaux téléphoniques mobiles. Face aux besoins de ses clients, la filiale Sud-africaine de SchlumbergerSema a entrepris le développement d'un système CIMS (Consolidated Information Management System), un produit complémentaire du central SMS de base (SMSC) afin de permettre le suivi des messages et la collecte de statistiques détaillées. CIMS enregistre chaque événement de trafic SMS et le tient à la disposition de la fonction de suivi des messages. SchlumbergerSema a essayé dans un premier temps d'implémenter ce système à l'aide d'un produit SGBDR leader pour gérer sa base de données d'événements. L'étude initiale a révélé qu'avec la technologie SGBDR classique, ce système pouvait exiger un matériel aussi puissant (et aussi coûteux) que le matériel SMSC de base, rendant l'application de suivi coûteuse, et non plus un outil complémentaire bon marché pour le système SMSC de base. Les performances, notamment le fait d'obtenir le taux d'insertion le plus rapide possible, étaient le principal souci de SchlumbergerSema, car la société voulait construire un système taillé pour faire face à la croissance future avec une cible de 8 000 événements de trafic par seconde (comme les insertions dans la base de données) aux heures de pointe. L'autre impératif était que le taux d'insertion ne devait pas se détériorer considérablement lorsque des requêtes sélectionnées étaient exécutées par les utilisateurs en même temps que les insertions.

Recherchant une solution technologique pouvant répondre à ses besoins, SchlumbergerSema a décidé d'effectuer un banc d'essai afin de déterminer le taux d'insertion qui serait réalisable avec Caché sur une plate-forme matérielle modeste. SchlumbergerSema a mené le test sur un système simple bi-processeur Intel de 1,0 GHz, doté de 1 Go de mémoire RAM et de cinq disques SCSI de 36 Go configuré en agrégat par bande. Elle a testé les opérations en monoflux (un seul flux d'événements chargeant la base de données) et multiflux (plusieurs flux chargeant la base de données) dans des conditions marquées par une absence de conflits de requêtes et de charge. Elle a constaté que ce système était capable de charger 7 700 enregistrements d'événement par seconde sans charge et, bizarrement, 8 800 sous la charge d'une requête. (La différence peut s'expliquer par les bons rendements tenant à la gestion de la mémoire tampon.) Satisfaite de ces résultats, la compagnie a depuis sauté le pas et décidé la mise en production du système sur le premier de ses sites clients. Le système est actuellement implémenté sur une baie de serveurs bi-processeur ProLiant exécutant Linux sur des processeurs Intel 1,4 GHz, avec 1,5 Go de mémoire RAM, chacun équipé d'un système multidisque composé de quatre disques de 72 Go chacun.

SchlumbergerSema a exprimé sa satisfaction quant à la vitesse de la base de données. De fortes augmentations du trafic sont prévues à Noël et pour le Jour de l'an, ainsi que dans les cas où le vote par SMS est introduit. SchlumbergerSema attend encore quelques réponses à certaines questions avant d'être complètement assurée que le système pourra gérer le trafic en rafale, comme l'évaluation de l'application toute entière et l'optimisation des procédures de gestion interne. L'optimisation des performances du système est une tâche difficile pour laquelle une connaissance de haut niveau de Caché ObjectScript est essentielle. SchlumbergerSema se fie ici à l'expertise et au support technique d'InterSystems ; la société se déclare satisfaite du niveau de service dont elle a bénéficié jusqu'à présent.

ANALYSE

Cet exemple montre comment les limites des systèmes relationnels, qui n'avaient jamais été destinés à gérer de gros volumes de données à haute vitesse, peuvent les rendre complètement inadaptés pour certains types d'applications. Dans ce cas, le besoin concernait une base de données pour une table très volumineuse ayant de très longues lignes et des relations avec un grand nombre de petites tables qui ne pouvaient tout simplement pas être prises en charge par un grand SGBDR.

Pourtant Caché a pu non seulement gérer ce type de structure de données mais également offrir un débit qui dépassait de très loin les attentes des utilisateurs.

COPYRIGHT

Publication externe d'IDC Information and Data . Toute information d'IDC utilisée dans une publicité, des communiqués de presse ou des supports promotionnels requiert une autorisation écrite délivrée par le vice-président ou le responsable local d'IDC. Un avant-projet du document proposé doit accompagner une telle demande. IDC se réserve le droit de refuser l'autorisation d'une utilisation externe pour tout motif.

Copyright 2003 IDC.

Toute reproduction sans consentement écrit est illicite.