Article
Guillaume Rongier · Avr 6, 2022 10m de lecture

Les globales sont des épées magiques pour stocker des données. Listes éparses (partie 3)

Dans les parties précédentes (1 et 2) nous avons parlé des globales en tant qu'arbres. Dans cet article, nous allons les considérer comme des listes éparses.

Une liste éparse - est un type de liste où la plupart des valeurs ont une valeur identique.

En pratique, vous verrez souvent des listes éparses si volumineuses qu'il est inutile d'occuper la mémoire avec des éléments identiques. Il est donc judicieux d'organiser les listes éparses de telle sorte que la mémoire ne soit pas gaspillée pour stocker des valeurs en double.

Dans certains langages de programmation, les listes éparses font partie intégrante du langage - par exemple, in J, MATLAB. Dans d'autres langages, il existe des bibliothèques spéciales qui vous permettent de les utiliser. Pour le C++, il s'agit de Eigen et d'autres bibliothèques de ce type.

Les globales sont de bons candidats pour la mise en œuvre de listes éparses pour les raisons suivantes :

1. Ils stockent uniquement les valeurs de nœuds particuliers et ne stockent pas les valeurs indéfinies ;
2. L'interface d'accès à une valeur de nœud est extrêmement similaire à ce que de nombreux langages de programmation proposent pour accéder à un élément d'une liste multidimensionnelle.

```
Set ^a(1, 2, 3)=5
Write ^a(1, 2, 3)
```

3. Une structure globale est une structure de niveau assez bas pour le stockage des données, ce qui explique pourquoi les globales possèdent des caractéristiques de performance exceptionnelles (des centaines de milliers à des dizaines de millions de transactions par seconde selon le matériel, voir 1)

   Puisqu'une globale est une structure persistante, il n'est logique de créer des listes éparses sur leur base que dans les situations où vous savez à l'avance que vous disposerez de suffisamment de mémoire pour elles.

   L'une des nuances de la mise en œuvre des listes éparses est le retour d'une certaine valeur par défaut si vous vous adressez à un élément indéfini.

Ceci peut être mis en œuvre en utilisant la fonction $GET dans COS. Prenons l'exemple d'une liste tridimensionnelle.

```
 SET a = $GET(^a(x,y,z), defValue)
```

Quel type de tâches nécessite des listes éparses et comment les globales peuvent-elles vous aider ?

Matrice d'adjacence

Ces matrices sont utilisées pour la représentation des graphiques :

Il est évident que plus un graphe est grand, plus il y aura de zéros dans la matrice. Si nous regardons le graphe d'un réseau social, par exemple, et que nous le représentons sous la forme d'une matrice de ce type, il sera principalement constitué de zéros, c'est-à-dire qu'il s'agira d'une liste éparse.

```
Set ^m(id1, id2) = 1
Set ^m(id1, id3) = 1
Set ^m(id1, id4) = 1
Set ^m(id1) = 3
Set ^m(id2, id4) = 1
Set ^m(id2, id5) = 1
Set ^m(id2) = 2
....
```

Dans cet exemple, nous allons sauvegarder la matrice d'adjacence dans le **^m** globale, ainsi que le nombre d'arêtes de chaque nœud (qui est ami et avec qui et le nombre d'amis).

Si le nombre d'éléments du graphique ne dépasse pas 29 millions (ce nombre est calculé comme 8 * longueur maximale de la chaîne), il existe même une méthode plus économique pour stocker de telles matrices - les chaines binaires, car elles optimisent les grands espaces d'une manière spéciale.

Les manipulations de chaînes binaires sont effectuées à l'aide de la fonction $BIT.

```
; setting a bit
SET $BIT(rowID, positionID) = 1
; getting a bit
Write $BIT(rowID, positionID)
```

Tableau des commutateurs FSM

Le graphe des commutateurs FSM étant un graphe régulier, le tableau des commutateurs FSM est essentiellement la même matrice d'adjacence dont nous avons parlé précédemment.

Automates cellulaires

L'automate cellulaire le plus célèbre est le jeu "Life", dont les règles (lorsqu'une cellule a de nombreux voisins, elle meurt) en font essentiellement une liste éparse.

Stephen Wolfram estime que les automates cellulaires représentent un nouveau domaine de la science. En 2002, il a publié un livre de 1280 pages intitulé "A New Kind of Science", dans lequel il affirme que les réalisations dans le domaine des automates cellulaires ne sont pas isolées, mais sont plutôt stables et importantes pour tous les domaines de la science.

Il a été prouvé que tout algorithme qui peut être traité par un ordinateur peut également être mis en œuvre à l'aide d'un automate cellulaire. Les automates cellulaires sont utilisés pour simuler des environnements et des systèmes dynamiques, pour résoudre des problèmes algorithmiques et à d'autres fins.

Si nous avons un champ considérable et que nous devons enregistrer tous les états intermédiaires d'un automate cellulaire, il est logique d'utiliser les globales.

Cartographie

La première chose qui me vient à l'esprit lorsqu'il s'agit d'utiliser des listes éparses est la cartographie.

En règle générale, les cartes comportent beaucoup d'espace vide. Si nous imaginons que la carte du monde est composée de grands pixels, nous verrons que 71 % de tous les pixels de la Terre seront occupés par le réseau creux de l'océan. Et si nous ajoutons uniquement des structures artificielles à la carte, il y aura plus de 95 % d'espace vide.

Bien sûr, personne ne stocke les cartes sous forme de tableaux bitmap, tout le monde utilise plutôt la représentation vectorielle.
Mais en quoi consistent les cartes vectorielles ? C'est une sorte de cadre avec des polylignes et des polygones. En fait, il s'agit d'une base de données de points et de relations entre eux.

L'une des tâches les plus difficiles en cartographie est la création d'une carte de notre galaxie réalisée par le télescope Gaia. Au sens figuré, notre galaxie est un gigantesque réseau creux : d'immenses espaces vides avec quelques points lumineux occasionnels - des étoiles. C'est 99,999999.....% d'espace absolument vide. Caché, une base de données basée sur des globales, a été choisie pour stocker la carte de notre galaxie.

Je ne connais pas la structure exacte des globales dans ce projet, mais je peux supposer que c'est quelque chose comme ça :

```
Set ^galaxy(b, l, d) = 1; le numéro de catalogue de l'étoile, s'il existe
Set ^galaxy(b, l, d, "name") = "Sun"
Set ^galaxy(b, l, d, "type") = "normal" ; les autres options peuvent inclure un trou
noir, quazar, red_dwarf et autres.
Set ^galaxy(b, l, d, "weight") = 14E50
Set ^galaxy(b, l, d, "planetes") = 7
Set ^galaxy(b, l, d, "planetes", 1) = "Mercure"
Set ^galaxy(b, l, d, "planetes", 1, weight) = 1E20
...
```

Où b, l, d représentent coordonnées galactiques*: la latitude, la longitude et la distance par rapport au Soleil.*

La structure flexible des globales vous permet de stocker toutes les caractéristiques des étoiles et des planètes, puisque les bases de données basées sur les globales sont exemptes de schéma.

Caché a été choisi pour stocker la carte de notre univers non seulement en raison de sa flexibilité, mais aussi grâce à sa capacité à sauvegarder rapidement un fil de données tout en créant simultanément des globales d'index pour une recherche rapide.

Si nous revenons à la Terre, les globales ont été utilisées dans des projets axés sur les cartes comme OpenStreetMap XAPI et FOSM, un branchement d'OpenStreetMap.

Récemment, lors d'un hackathon Caché, un groupe de développeurs a mis en œuvre des index géospatiaux en utilisant cette technologie. Pour plus de détails, consultez l'article.

Mise en œuvre d'index géospatiaux à l'aide de globales dans OpenStreetMap XAPI

Les illustrations sont tirées de cette présentation.

Le globe entier est divisé en carrés, puis en sous-carrés, puis en encore plus de sous-carrés, et ainsi de suite. Au final, nous obtenons une structure hiérarchique pour laquelle les globales ont été créées.

À tout moment, nous pouvons instantanément demander n'importe quelle case ou la vider, et toutes les sous-carrés seront également retournées ou vidées.

Un schéma détaillé basé sur les globales peut être mis en œuvre de plusieurs façons.

Variante 1:

```
Set ^m(a, b, a, c, d, a, b,c, d, a, b, a, c, d, a, b,c, d, a, 1) = idPointOne
Set ^m(a, b, a, c, d, a, b,c, d, a, b, a, c, d, a, b,c, d, a, 2) = idPointTwo
...
```

Variante 2:

```
Set ^m('abacdabcdabacdabcda', 1) = idPointOne
Set ^m('abacdabcdabacdabcda', 2) = idPointTwo
...
```

Dans les deux cas, il ne sera pas très difficile dans COS/M de demander des points situés dans un carré de n'importe quel niveau. Il sera un peu plus facile de dégager des segments d'espace carrés de n'importe quel niveau dans la première variante, mais cela est rarement nécessaire.

Un exemple de carré de bas niveau :

Et voici quelques globales du projet XAPI : représentation d'un index basé sur des globales :

La globale **^voie** est utilisé pour stocker les sommets des polylines (routes, petites rivières, etc.) et des polygones (zones fermées : bâtiments, bois, etc.).

Une classification approximative de l'utilisation des listes éparses dans les globales.

1. Nous stockons les coordonnées de certains objets et leur état (cartographie, automates cellulaires).
2. Nous stockons des matrices creuses.

Dans la variante 2), lorsqu'une certaine coordonnée est demandée et qu'aucune valeur n'est attribuée à un élément, nous devons obtenir la valeur par défaut de l'élément de la liste éparse.

Les avantages que nous obtenons en stockant des matrices multidimensionnelles dans les globales

Suppression et/ou sélection rapide de segments d'espace qui sont des multiples de chaînes, de surfaces, de cubes, etc. Pour les cas avec des index intégraux, il peut être pratique de pouvoir supprimer et/ou sélectionner rapidement des segments d'espace qui sont des multiples de chaînes, de surfaces, de cubes, etc.

La commande Kill permet de supprimer un élément autonome, une chaîne de caractères et même une surface entière. Grâce aux propriétés de la globale, elle se produit très rapidement, mille fois plus vite que la suppression élément par élément.

L'illustration montre un tableau tridimensionnel dans la globale **^a** et différents types d'enlèvements.

Pour sélectionner des segments d'espace par des indices connus, vous pouvez utiliser la commande Merge.

Sélection d'une colonne de la matrice dans la colonne Variable :

```
; Définissons un tableau tridimensionnel creux 3x3x3
Set ^a(0,0,0)=1,^a(2,2,0)=1,^a(2,0,1)=1,^a(0,2,1)=1,^a(2,2,2)=1,^a(2,1,2)=1
Colonne de fusion = ^a(2,2)
; Produisons la colonne Variable
Zwrite colonne
```

Produit :

```
Column(0)=1
Colonne(2)=1
```

Ce qui est intéressant, c'est que nous avons obtenu un tableau épars dans la colonne Variable que vous pouvez adresser via $GET puisque les valeurs par défaut ne sont pas stockées ici.

La sélection de segments d'espace peut également se faire à l'aide d'un petit programme utilisant la fonction $Order. Ceci est particulièrement utile pour les espaces dont les indices ne sont pas quantifiés (cartographie).

Conclusion

Les réalités d'aujourd'hui posent de nouveaux défis. Les graphes peuvent comporter des milliards de sommets, les cartes peuvent avoir des milliards de points, certains peuvent même vouloir lancer leur propre univers basé sur des automates cellulaires (1, 2).

Lorsque le volume de données dans les listes éparses ne peut pas être comprimé dans la RAM, mais que vous devez quand même travailler avec elles, vous devriez envisager de mettre en œuvre de tels projets en utilisant des globales et des COS.

Clause de non-responsabilité :: *cet article et les commentaires le concernant reflètent uniquement mon opinion et n'ont rien à voir avec la position officielle de la société d'InterSystems.*

#Débutant #Globals #Indexation #Modèle de données #Performances #Tables relationnelles #Valeur clé #Caché #InterSystems IRIS

URL de la source:https://fr.community.intersystems.com/post/les-globales-sont-des-%C3%A9p%C3%A9es-magiques-pour-stocker-des-donn%C3%A9es-listes-%C3%A9parses-partie-3