

Artículo

[Estevan Martinez](#) · 23 jul, 2019 Lectura de 7 min

Estructura Interna de los Bloques de Bases de Datos en Caché (Parte 1)

Los globales de InterSystems Caché proporcionan un conjunto de funciones muy útiles para los desarrolladores. Pero, ¿por qué los globales son tan rápidos y eficientes?

Teoría

Básicamente, la base de datos de Caché es un catálogo con el mismo nombre que la base de datos y contiene el archivo CACHE.DAT. En los sistemas Unix, la base de datos también puede ser una partición normal del disco.

Todos los datos en Caché se almacenan en bloques que, a su vez, se organizan como un árbol-B* balanceado. Si tenemos en cuenta que todos los globales básicamente se almacenan en un árbol, los subíndices de los globales se representarán como ramas, mientras que los valores de los subíndices globales se almacenarán como hojas. La diferencia entre un árbol-B* balanceado y un árbol-B ordinario es que sus ramas también tienen los enlaces apropiados para realizar iteraciones a través de los subíndices (por ejemplo, en nuestro caso los globales) usando rápidamente las funciones [\\$Order](#) y [\\$Query](#) sin necesidad de regresar al tronco del árbol.

De manera predeterminada, cada bloque en el archivo de la base de datos tiene un tamaño fijo de 8,192 bytes. Por lo tanto, no podrá modificar el tamaño de los bloques en una base de datos que ya exista. Mientras crea una nueva base de datos puede elegir entre bloques de 16 KB, 32 KB o incluso bloques de 64 KB, dependiendo del tipo de datos que almacenará. Sin embargo, siempre tenga en cuenta que todos los datos se leen bloque por bloque, en otras palabras, incluso si solicita un solo valor de 1 byte de tamaño el sistema leerá muchos bloques, entre los cuales, el bloque de datos que solicitó será el último. También debe recordar que Caché utiliza búferes globales para almacenar los bloques de las bases de datos en la memoria para utilizarlos después, y que los búferes de bloques tienen el mismo tamaño. No podrá utilizar una base de datos existente o crear una nueva cuando el búfer global, con el tamaño de bloques correspondiente, falte en el sistema. Además, deberá definir la cantidad de memoria que desea asignar para el tamaño específico de los bloques. Es posible utilizar búferes con bloques más grandes que los bloques en las bases de datos, pero en este caso, cada bloque en el búfer solamente almacenará un bloque en dicha base, que será incluso más pequeño.

Menu [Home](#) | [About](#) | [Help](#) | [Logout](#) [System](#) > [Configuration](#) > [Startup Settings](#) > [Edit Startup Settings](#)

Edit: DBSizesAllowed Server: **MBP-Dmitry** Namespace: **%SYS**
User: **UnknownUser** Licensed to: **2016.x Field Test** Instance: **CACHE**

[Save](#) [Cancel](#)

Use the form below to edit a startup setting:

DBSizesAllowed (Check all that apply. 8K is required.)

8K (8192)

16K (16384)

32K (32768)

64K (65536)

NOTE: You must also configure Memory allocation in order to create databases with selected sizes.

Stores a list of allowed database block sizes.

En esta imagen, se asignó memoria para un búfer global con un tamaño de 8 KB, con la finalidad de utilizarla en bases de datos conformadas por bloques de 8 KB. En esta base de datos, los bloques que no están vacíos se definieron en los mapas, de tal modo que alguno de los mapas contenga 62,464 bloques (para bloques con capacidad de 8KB).

Tipos de Bloques

El sistema es compatible con varios tipos de bloques. En cada nivel, los enlaces correctos de un bloque deben señalar un bloque del mismo tipo o un bloque nulo que defina el final de los datos.

- Tipo 9: Bloques de un catálogo de globales. En estos bloques generalmente se describen todos los globales existentes junto con sus parámetros, incluida la compilación de los subíndices globales, el cual es uno de los parámetros más importantes y no puede modificarse una vez que el global se creó
- Tipo 66: Bloques punteros de alto nivel. Solo un bloque de un catálogo de globales puede estar en la parte superior de estos bloques.
- Tipo 6: Bloques punteros de nivel inferior. Únicamente los bloques punteros de alto nivel pueden estar en la parte superior de estos bloques y solamente los bloques de datos pueden colocarse en niveles inferiores.
- Tipo 70: Bloques punteros tanto de alto nivel como de nivel inferior. Estos bloques se utilizan cuando el global correspondiente almacena una pequeña cantidad de valores y, por lo tanto, no es necesario tener varios niveles de bloques. Estos bloques generalmente señalan hacia los bloques de datos, al igual que los bloques de un catálogo de globales.
- Tipo 2: Bloques punteros para almacenar una cantidad relativamente grande de globales. Para asignar valores de manera equitativa entre los bloques de datos, es posible que desee crear niveles adicionales de bloques punteros. Estos bloques generalmente se colocan entre los bloques punteros.
- Tipo 8: Bloques de datos. En estos bloques generalmente se almacenan varios nodos globales en lugar de un solo nodo.
- Tipo 24: Bloques para cadenas largas. Cuando el valor de un solo global es mayor que un bloque, dicho valor se registrará en un bloque especial para cadenas largas, mientras que el nodo del bloque de datos almacenará enlaces en la lista de bloques para cadenas largas junto con la longitud total de este valor.
- Tipo 16: Diagrama de bloques. Estos bloques están diseñados para almacenar información sobre bloques que no han sido asignados.

Por lo tanto, el primer bloque en una base de datos típica de Caché contiene la información de servicio acerca del mismo archivo en la base de datos, mientras que los segundos bloques constituyen un diagrama de bloques. El primer bloque del catálogo va en el tercer lugar (bloque #3), y una sola base de datos puede tener varios bloques en el catálogo. Los siguientes son bloques punteros (ramas), bloques de datos (hojas) y bloques de cadenas largas. Como ya mencioné anteriormente, los bloques en los catálogos globales almacenan información sobre todos los globales que existen en la base de datos o en la configuración global (si no hay datos disponibles en este tipo de globales). En este caso, un nodo que describa a dicho global tendrá un puntero nulo de nivel inferior. Puede consultar la lista de los globales que están disponibles en el catálogo de globales que se encuentra en el portal de administración. En este portal también se le permitirá guardar un global en el catálogo después de haberlo eliminado (por ejemplo, guardar su secuencia de compilación) así como crear un nuevo global con un tipo de compilación ya sea predeterminada o personalizada.

Menu

[Home](#) | [About](#) | [Help](#) | [Logout](#)

[System](#) > [Configuration](#) > [Local Databases](#) > [Globals](#)

Globals

Server: **MBP-Dmitry** Namespace: %S

User: **UnknownUser** Licensed to: 2011

[Create New Global](#)

Globals in database /opt/cache/mgr/samples/:

Globals: System Filter: Page size:

Name	Permission	Empty	Keep			
Aviation.AircraftD	RW	No	No	View	-	Properties
Aviation.AircraftI	RW	No	No	View	-	Properties
Aviation.Countries	RW	No	No	View	-	Properties
Aviation.CrewI	RW	No	No	View	-	Properties
Aviation.EventD	RW	No	No	View	-	Properties
Aviation.EventI	RW	No	No	View	-	Properties
Aviation.States	RW	No	No	View	-	Properties
CacheMsg	RW	No	No	View	-	Properties
Cinema.ReviewD	RW	No	No	View	-	Properties
CinemaooFilmCategoryD	RW	No	No	View	-	Properties
CinemaooFilmCategoryI	RW	No	No	View	-	Properties
CinemaooFilmD	RW	No	No	View	-	Properties
CinemaooFilmI	RW	No	No	View	-	Properties
CinemaooShowD	RW	No	No	View	-	Properties
CinemaooTheaterD	RW	No	No	View	-	Properties
DeepSee.ActiveTasks	RW	Yes	No	View	Delete	Properties
DeepSee.AgentLog	RW	No	No	View	-	Properties
DeepSee.Agents	RW	No	No	View	-	Properties
DeepSee.BucketList	RW	No	No	View	-	Properties
DeepSee.Cache.Results	RW	No	No	View	-	Properties

[\[Next page\]](#)

Create a new global in /opt/cache/mgr/samples/:

Name:

Collating Sequence: Cache standard

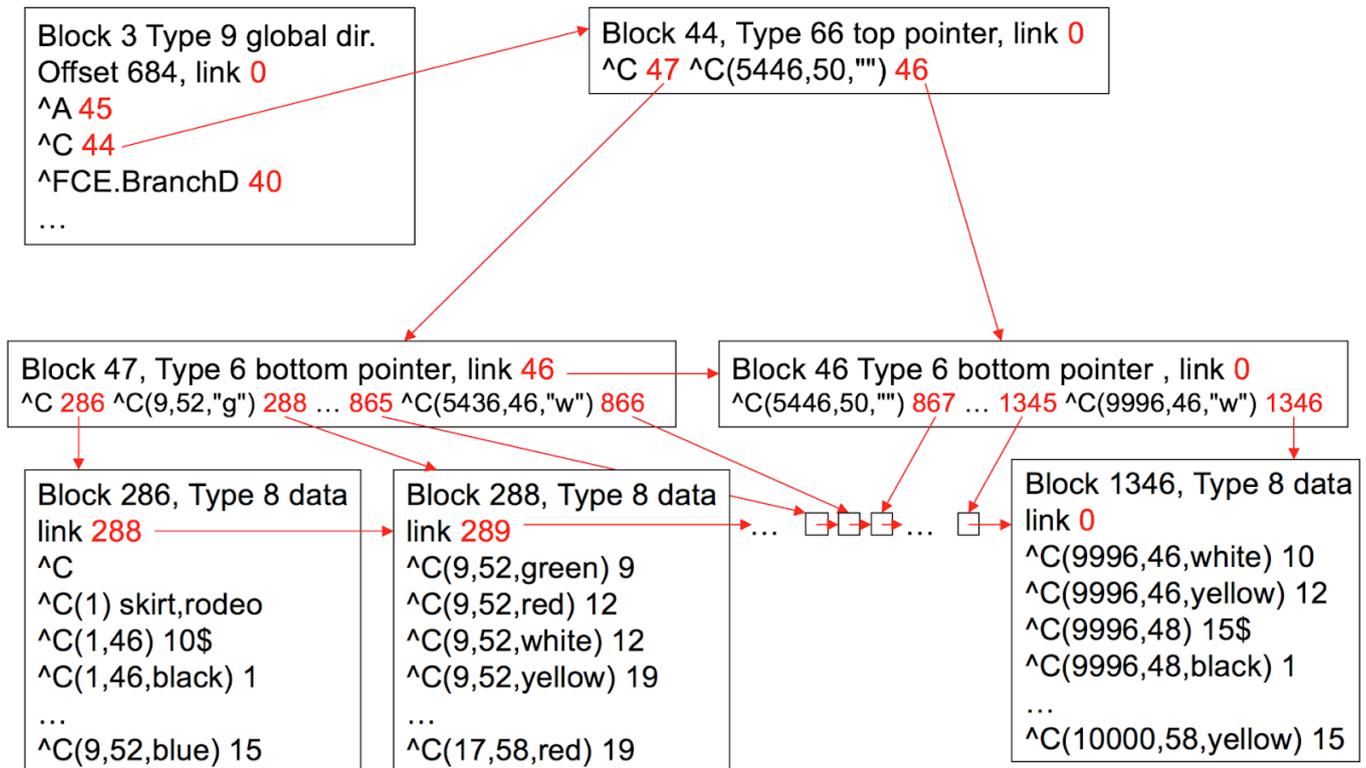
Pointer Block Start at: 16 (1-1664)

Data Block Start at: 50 (1-1664)

Preserve global attributes on delete: No

Save Close

En general, el árbol de bloques puede representarse como la imagen que se muestra a continuación. Tenga en cuenta que los enlaces a estos bloques se muestran en color rojo.



Integridad de la Base de Datos

En la versión actual de Caché, resolvimos los temas y los problemas más importantes con las bases de datos, entonces el riesgo de degradación en las bases de datos es extremadamente bajo. Sin embargo, le recomendamos que realice comprobaciones de la integridad automáticas en las bases de datos de manera regular utilizando nuestra herramienta ^Integrity, la cual puede ejecutarse en el terminal desde el namespace %SYS, mediante nuestro portal de administración, en la página de la Base de datos o desde el administrador de tareas.

Por defecto, la comprobación de la integridad automática ya está configurada y predefinida, por lo tanto lo único que debe hacer es activarla:

The following is a list of databases:

- CACHESYS
- CACHELIB
- CACHETEMP
- CACHE
- CACHEAUDIT
- BLOCKS
- DOCBOOK
- SAMPLES
- USER

Integrity Check

User: UnknownUser
 Namespace: %SYS

This Integrity Check will be run in the background and the result will be saved in a file you designate below.

Please enter a file name for saving the result:

Stop after any error

Please check the databases that you want to perform the Integrity Check:

Name	Directory
<input type="checkbox"/> CACHESYS	/opt/cache/mgr/
<input type="checkbox"/> CACHELIB	/opt/cache/mgr/cache/lib/
<input type="checkbox"/> CACHETEMP	/opt/cache/mgr/cache/temp/
<input type="checkbox"/> CACHE	/opt/cache/mgr/cache/
<input type="checkbox"/> CACHEAUDIT	/opt/cache/mgr/cache/audit/
<input type="checkbox"/> BLOCKS	/opt/cache/mgr/user/blocks/
<input type="checkbox"/> DOCBOOK	/opt/cache/mgr/docbook/
<input type="checkbox"/> SAMPLES	/opt/cache/mgr/samples/
<input type="checkbox"/> USER	/opt/cache/mgr/user/

You may select specific globals for one selected database.

The following is a list of tasks scheduled for execution:

Task Name	Task Type	Namespace	Description	ID	Suspended	Last Finished	Next Scheduled	
Switch Journal	System	%SYS	Switches the journal file at midnight every day	1		2016-09-23 00:17	2016-09-24 00:00	History Run
Purge Journal	System	%SYS	Purges old journal files every night at 12:30 am	2		2016-09-23 02:00	2016-09-24 00:30	History Run
Purge Tasks	System	%SYS	Purges the task history global every night at 1:00 am	3		2016-09-23 02:00	2016-09-24 01:00	History Run
Integrity Check	System	%SYS	Integrity check for databases at 2:00 am every Monday	4	Suspend Reschedule		2016-09-26 02:00	History -
Security Scan	System	%SYS	Scans the security database at midnight every day	5		2016-09-23 00:17	2016-09-24 00:00	History Run
Diagnostic Report	System	%SYS	Send system diagnostic reports to WRC On Demand, and/or on a schedule	6				History Run
Purge Audit Database	System	%SYS	Purges old Audit information after Switch Journal is run	7		2016-09-23 00:18	Runs After #1	History Run
Inventory Scan	System	%SYS	Run a scan of the system inventory on install or upgrade and on demand thereafter	8		2016-09-12 11:25		History Run
Purge errors and log files	System	%SYS	Purges errors and log files at 1:00 am	9		2016-09-23 02:00	2016-09-24 01:00	History Run
Check Logging activity	System	%SYS	Check active application logging at 1:00 am	10		2016-09-23 02:00	2016-09-24 01:00	History Run
Purge Backup Log	System	%SYS	Purges old messages from Cache backup log every night at 1:30 am	11		2016-09-23 02:00	2016-09-24 01:30	History Run
Purge ZEN Reports temp files	System	%SYS	Purges ZEN Reports temp files every night at 1:30 am	12		2016-09-23 02:00	2016-09-24 01:30	History Run

La opción "integrity check" incluye la verificación de los enlaces en los niveles inferiores, la validación de los tipos de bloques, el análisis de los enlaces correctos y la correspondencia de los nodos globales con la secuencia para la aplicación de la compilación. Si el programa encuentra algún error durante la comprobación de la integridad, puede ejecutar nuestra herramienta ^REPAIR desde el namespace %SYS. Al utilizar esta herramienta, podrá

visualizar cualquier bloque y modificarlo según sea necesario, por ejemplo, para reparar su base de datos.

Práctica

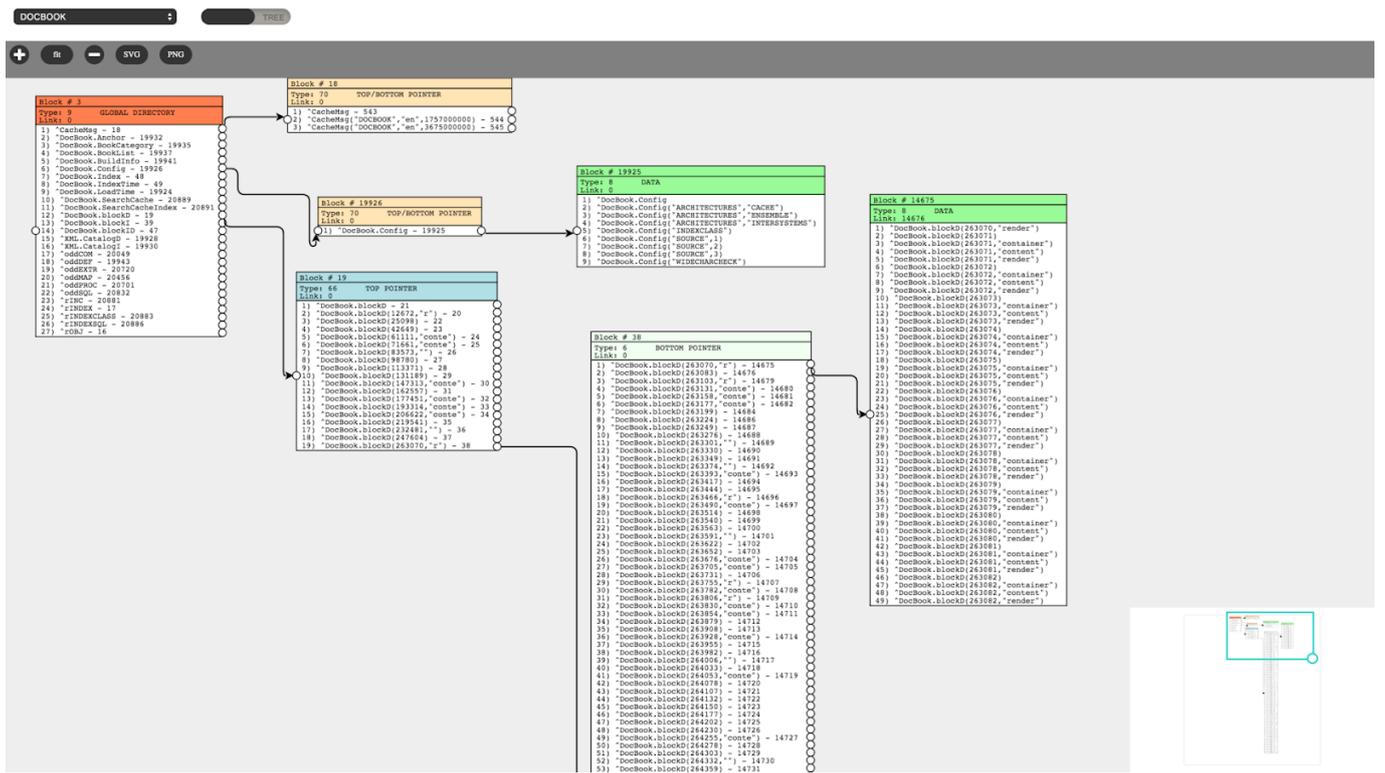
Esto fue solo la teoría. Todavía es difícil comprender qué hace un global y cómo se ven en realidad sus bloques. Actualmente, la única manera en que podemos visualizar los bloques es utilizando nuestra herramienta ^REPAIR, que mencionamos anteriormente. A continuación, se muestra la forma característica en que se visualizan los resultados de este programa:

```
Map Blocks: 2
Entering Block Repair Menu
Block #: 3
Block # 3                               Type: 9 GLOBAL DIR
Link Block: 0                            Offset: 2516
Count of Nodes: 111                      Big String Nodes: 0
Pointer Length: 0                         Next Pointer Length: 0   Diff Byte: Hex 0
Pointer Reference:                        ^Aviation.AircraftD
Next Pointer Reference:
Next pointer stored? No
```

--more--

#	Node	PTR,NEW	GROWTH	COLL	TYPE	PROT
1	^Aviation.AircraftD	8804	50	5	0	195
2	^Aviation.AircraftI	8869	50	5	0	195
3	^Aviation.Countries	8874	50	5	0	195
4	^Aviation.CrewI	8876	50	5	0	195
5	^Aviation.EventD	8882	50	5	0	195
6	^Aviation.EventI	10036	50	5	0	195
7	^Aviation.States	10040	50	5	0	195
8	^CacheMsg	49	50	5	0	195
9	^Cinema.ReviewD	10136	50	5	0	195

Hace tres años, Dmitry comenzó un nuevo proyecto para desarrollar una herramienta que nos permitiera iterar a través de un árbol de bloques sin ningún riesgo de que la base de datos se dañara, visualizar estos bloques en una IU web y proporcionara opciones para guardar dichas visualizaciones en formatos SVG o PNG. El proyecto se llama CacheBlocksExplorer, y su código fuente está disponible para que lo descargue en [GitHub](#).



A partir del momento de la redacción del artículo original en 2016.

Las características que se implementaron incluyen:

- Visualizar cualquier base de datos que se haya configurado o simplemente instalado en el sistema;
- Visualizar la información por bloque, los tipos de bloques, el puntero correcto o una lista de nodos con sus enlaces;
- Visualizar a detalle la información sobre cualquier nodo que señale hacia un bloque de nivel inferior;
- Ocultar bloques al eliminar los enlaces hacia ellos (sin dañar los datos almacenados en estos bloques).

Lista de cosas pendientes:

- Mostrar los enlaces correctos: en la versión actual los enlaces correctos se muestran en la información sobre los bloques, pero lo mejor sería que pudiéramos visualizarlos como flechas,
- Mostrar los bloques de las cadenas largas: en la versión actual simplemente no se muestran,
- Mostrar todos los bloques que se encuentran en el catálogo de globales en lugar de que solamente se visualice el tercero

Asimismo, me gustaría mostrar el árbol completo, pero todavía no encuentro ninguna biblioteca capaz de renderizar rápidamente los cientos de miles de bloques junto con sus enlaces. Con la biblioteca actual es posible renderizarlos en los navegadores web, pero mucho más lentamente que el tiempo que le toma a Caché leer toda la estructura.

En el próximo artículo, intentaré describir con más detalle cómo funciona, proporcionar algunos ejemplos sobre su uso y mostrar cómo se recuperan muchos datos que pueden procesarse sobre los globales y los bloques mediante mi Cache Block Explorer.

[#Bases de datos](#) [#Administración del sistema](#) [#Caché](#)

URL de
fuente: <https://es.community.intersystems.com/post/estructura-interna-de-los-bloques-de-bases-de-datos-en-cach%C3%A9-parte-1>