The table Command: Basics

spark, 2010-13-01

The BIG-IP is a powerful tool when it comes to managing your application traffic. One of the reasons is F5's CMP technology: the ability to scale across many CPUs by running independently on each of them. As any advanced iRule writer could tell you, though, this independence can sometimes make it difficult or impossible to treat the system as one cohesive whole. The problem is that maintaining a consistent set of data spread across processors like that is a long-standing one in computer science. There's just no way to do it in a general enough way for something like Tcl global variables to work and still get good performance.

In v10, the session table was the one small escape hatch for any iRule to have a globally available set of shared data and not be pinned to a single processor. Although it has existed in iRules for some time, the `session` command was never designed to be a general purpose tool for this sort of thing: the interface has a lot of unnecessary complexity, and there is significant missing functionality. But when you get down to brass tacks, it works; it solves the problem and maintains the speed we need.

With version 10.1, we've given the session table some long-sought functionality, and revamped its iRules interface completely to give users a new, cleaner, full-featured way to keep track of global data. We are very proud to introduce the `table` command...

There's quite a lot to cover, so we'll take it slowly.

The Basics

At its heart, the `table` command is just another way to access the session table, and the fundamental concepts are not much different than they were before. The session table is a global table shared across all processing units in the BIG-IP system. It stores values that are indexed by keys. An entry in the session table consists of a key, a value, and some associated metadata (which will be covered later). The `session` command allows you to set a key/value pair, lookup a value for a given key, or delete a key/value pair. It looks like this:

```
session add <mode> <key> <data> [<timeout>]
session lookup <mode> <key>
session delete <mode> <key>
```

where `<mode>` is a value that no longer has any meaning in v10, and `<key>` can take some baroque syntax, which is also meaningless in v10. When those commands are used, it might look like:

```
session add uie "$key any pool any virtual" $data 180
session lookup uie "$key any pool"
session lookup uie "$key" any pool
session delete uie "$key"
```

To someone unfamiliar with iRules, this might not make a lot of sense. What's "uie"? Why the "any pool" business? And what does any of this have to do with sessions?

When we set out to improve this interface, the first thing we did was to come up with a new command name, one that might hint that this is something general purpose that we're dealing with: hence, the `table` command. A new command name also ensures that any existing iRules using the `session` command wouldn't be interfered with. The second thing we did was get rid of all those bulky unused parameters. The `table` command's basic form looks like:

```
table set $key $data 180
table lookup $key
table delete $key
```

That's already an improvement! But we didn't stop there. Not by a long shot.

Continue reading part two in the series: New Ways To Set Data