

Lua Performance Tips What can I do to increase the performance of a Lua program?

Avoid globals

Local variables are very fast as they reside in virtual machine registers, and are accessed directly by index. Global variables on the other hand, reside in a lua table and as such are accessed by a hash lookup.” – Thomas Jefferson

```
local find = string.find
local sin = math.sin
```

Use table preallocation

Inserting element to table requires reallocation. For small tables better is to preallocate:

```
local t = { 0, 0, 0, 0, 0 } -- allocate 5 element table
```

Avoid creating tables or closures in loops

Creating or extending table requires heap allocation.

Use short inline instad of function calls

Also true for LUAjit:

```
t[#t+1] = 0 is faster than table.insert(t, 0).
```

Avoid new string creation

Lua hashes all strings on creation, this makes comparison and using them in tables very fast and reduces memory use since all strings are stored internally only once. But it makes string creation more expensive.

```
local ret = {};
for i=1, C do
    ret[#ret+1] = foo();
end
ret = table.concat(ret);
```

From:

<https://niziak.spox.org/wiki/> - **niziak.spox.org**

Permanent link:

<https://niziak.spox.org/wiki/programming:lua:optimisation>

Last update: **2020/07/03 09:48**

