Allocation policy enforcement

Svyatoslav Feldsherov

Allocation is 1ns to 50ns

```
Benchmark                           Time             CPU   Iterations
---------------------------------------------------------------------
BM_Glibc/64/threads:1            9.30 ns         9.30 ns     78626554
BM_Glibc/64/threads:4            2.90 ns         11.6 ns     61778116
BM_Glibc/64/threads:8            3.28 ns         25.9 ns     25718152
BM_Glibc/256/threads:1           9.12 ns         9.11 ns     74287685
BM_Glibc/256/threads:4           3.01 ns         12.0 ns     48275676
BM_Glibc/256/threads:8           3.30 ns         26.3 ns     26199608
BM_Glibc/4096/threads:1          36.3 ns         36.3 ns     19100403
BM_Glibc/4096/threads:4          11.1 ns         44.3 ns     14262944
BM_Glibc/4096/threads:8          10.1 ns         79.8 ns      8929152
BM_Glibc/1048576/threads:1       36.4 ns         36.4 ns     19090817
BM_Glibc/1048576/threads:4       11.2 ns         44.8 ns     14419624
BM_Glibc/1048576/threads:8       10.4 ns         82.4 ns      9245784
```

Allocation is 1ns to 500ns

```
Benchmark                                   Time             CPU   Iterations
-----------------------------------------------------------------------------
BM_TCMalloc/64/threads:1                 6.10 ns         6.10 ns    116268227
BM_TCMalloc/64/threads:4                 1.77 ns         7.10 ns     82612188
BM_TCMalloc/64/threads:8                 1.67 ns         12.9 ns     51157976
BM_TCMalloc/256/threads:1                5.99 ns         5.99 ns    115207011
BM_TCMalloc/256/threads:4                1.91 ns         7.64 ns     94367140
BM_TCMalloc/256/threads:8                1.61 ns         12.7 ns     56211648
BM_TCMalloc/4096/threads:1               6.19 ns         6.19 ns    112071810
BM_TCMalloc/4096/threads:4               1.84 ns         7.36 ns     89468952
BM_TCMalloc/4096/threads:8               1.77 ns         13.7 ns     53497208
BM_TCMalloc/1048576/threads:1            44.0 ns         44.0 ns     15957359
BM_TCMalloc/1048576/threads:2             147 ns          293 ns      2999688
BM_TCMalloc/1048576/threads:8             429 ns         3051 ns       219592
```

This is lot if you tune micros

- Wait! Micros, why, what for?
  - High frequency trading systems
  - User space networking
  - Large scale databases in per row code

Ok, we optimized all allocations!
How avoid addition of new ones?

Perf testing!

Does not work :(

What do we want?

```
> thread_local allow_allocations;
>
> int main() {
>   allow_allocations = false;
>   auto ac = std::make_unique<AwesomeClass>(); // <- boom!
> }
```

Custom allocator

```
> void *malloc(size_t size) noexcept {
>   if (!allow_allocations) {
>     DoBoom();
>   }
>   return __libc_malloc(size);
> }
```

Malloc hooks

Available until glibc 2.34:

- **__malloc_hook**
- **__realloc_hook**
- **__free_hook**
- **__memalign_hook**

https://www.gnu.org/software/libc/manual/html_node/Hooks-for-Malloc.html

A __malloc_hook

```c
void* my_malloc_hook(size_t size, const void *caller)
{
    void *result;

    __malloc_hook = old_malloc_hook; //  <- Unprotected write

    result = malloc(size);

    old_malloc_hook = __malloc_hook;

    DoBoom();
    __malloc_hook = my_malloc_hook;

    return result;
}
```

Malloc hooks

TCMalloc

```
> thread_local allow_allocations;
>
> int main() {
>   allow_allocations = false;
>   MallocHook_AddNewHook(&new_hook);
>   auto ac = std::make_unique<AwesomeClass>(); // <- boom!
> }
```

https://github.com/gperftools/gperftools

A TCMalloc hook

```
> thread_local bool in_hook = false;
>
> void new_hook(const void *caller, unsigned long size)
> {
>   if (in_hook || allocations_allowed) {
>     return;
>   }
>
>   in_hook = true;
>   DoBoom()
>   in_hook = false;
> }
```

Thank you for your attention!

- linkedin.com/in/svyat/
- svyat.dev