Improving circulation performance by caching *yaml* file

Requirements:

- Install **memcached**

```
sudo apt-get install memcached
```

- Install **Devel::NYTProf** :

http://wiki.koha-community.org/wiki/Profiling_with_Devel::NYTProf

- [Install **Plack**] Optional, also work on CGI

http://wiki.koha-community.org/wiki/Plack

But at this step :

```
Devel::NYTPROF write an output file, by default in the directory
where the script is executed. You can change the location by
providing a NYTPROF variable. Since the koha scripts are launched
from Apache, you can add the variable in your apache configuration :

...
 SetEnv KOHA_CONF "/etc/koha/koha-conf.xml"
 SetEnv PERL5LIB "/usr/share/koha/lib"
 SetEnv NYTPROF "file=/tmp/nytprof.out:addpid=1:endatexit=1:stmts=0"
 ...
```

Put stmts to 1 to enable profiling line by line.

```
SetEnv NYTPROF "file=/tmp/nytprof.out:addpid=1:endatexit=1:stmts=1"
```

- Add these lines in **koha-httpd.conf** if you use **CGI** or
  in **intranet.psgi** and **opac.psgi** if you use **Plack**

```
SetEnv CACHING_SYSTEM "memcached"
SetEnv MEMCACHED_SERVERS "localhost:11211"
SetEnv MEMCACHED_NAMESPACE "CacheKoha"
SetEnv MEMCACHED_DEBUG=1
```

- Add this line in **koha-httpd.conf** if you use **CGI** or
  in **intranet.psgi** and **opac.psgi** if you use **Plack** if you want debug informations on log file

```
SetEnv DEBUG=1
```

Test Plan (CGI)

Before applying the patch

- In circ/circulation.pl add -d:NYTProf to the shebang

```
#!/usr/bin/perl -d:NYTProf
```

- Go to the circulation page

```
http://admin.koha.local/cgi-bin/koha/circ/circulation.pl
```

- Check out a patron

It will generate a **nytprof.out.XXXX** file in your **/tmp** directory.

- Mine the data collection

```
nytprofhtml -f /tmp/nytprof.out.XXXX --open
```

It should open your browser with the details of NYTProf

- Select **C4/Utils/DataTables/ColumnSettings.pm** from the drop-down list.
- Click on "line"

Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
1	1	1	183µs	515ms	C4::Utils::DataTables::ColumnsSettings::get_columns
1	1	1	100µs	509ms	C4::Utils::DataTables::ColumnsSettings::get_yaml
1	1	1	57µs	66µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@8
1	1	1	29µs	98µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@4
1	1	1	25µs	247µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@3
1	1	1	20µs	31µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@7
1	1	1	16µs	19µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@6
1	1	1	16µs	72µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@5
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:45]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:61]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::get_modules
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::update_columns

Here we can see that **get_yaml** takes ~500ms.

- Click on “get_yaml”

```
10                                          # spent 509ms (100µs+509) within C4::Utils::DataTables::ColumnsSettings::get_yaml which was called:
                                            #    once (100µs+509ms) by C4::Utils::DataTables::ColumnsSettings::get_columns at line 32
                                            sub get_yaml {
11   1   9µs   1   23µs                       my $yml_path =
                                                # spent    23µs making 1 call to C4::Context::config
12                                              C4::Context->config('intranetdir') . '/admin/columns_settings.yml';
13
14   1   7µs   1    8µs                       my $cache  = Koha::Cache->get_instance();
                                                # spent     8µs making 1 call to Koha::Cache::get_instance
15   1   6µs   1  394µs                       my $yaml = $cache->get_from_cache('ColumnsSettingsYaml');
                                                # spent   394µs making 1 call to Koha::Cache::get_from_cache
16   1   8µs                                 unless ($yaml) {
17   2  28µs   1  507ms                           $yaml = eval { YAML::LoadFile($yml_path) };
                                                    # spent   507ms making 1 call to YAML::LoadFile
18   1   2µs                                     warn "############################ Not In Cache ############################" if $ENV{DEBUG};
19   1   1µs                                     warn "ERROR: the yaml file for DT::ColumnsSettings is not correctly formated: $@"
20                                                    if $@;
21   1   9µs   1 1.03ms                          $cache->set_in_cache('ColumnsSettingsYaml', $yaml, { expiry => 3600 });
                                                    # spent  1.03ms making 1 call to Koha::Cache::set_in_cache
22                                              }
23                                              else {
24                                                  warn "############################ In Cache ############################" if $ENV{DEBUG};
25                                              }
26   1  12µs   1   22µs                       return $yaml;
                                                # spent    22µs making 1 call to Cache::Memory::Entry::DESTROY
27                                          }
```

We can see that all the time is spent in that single line :

```
$yaml = eval { YAML::LoadFile($yml_path) };
```

- Apply patch

After applying the patch

On the first run, you are not supposed to see any change (not yet in cache).

But at second run, it's supposed to be cached. We are going to check it.

- Reload the page without changing anything else.

- Re-open NYTProf :

Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
1	1	1	154µs	7.03ms	C4::Utils::DataTables::ColumnsSettings::get_columns
1	1	1	54µs	470µs	C4::Utils::DataTables::ColumnsSettings::get_yaml
1	1	1	28µs	98µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@4
1	1	1	26µs	278µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@3
1	1	1	20µs	31µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@7
1	1	1	18µs	26µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@8
1	1	1	16µs	71µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@5
1	1	1	16µs	19µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@6
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:45]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:61]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::get_modules
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::update_columns

```
10                                  # spent 470µs (54+416) within C4::Utils::DataTables::ColumnsSettings::get_yaml which was called:
                                    #    once (54µs+416µs) by C4::Utils::DataTables::ColumnsSettings::get_columns at line 32
                                    sub get_yaml {
11  1   17µs    1   22µs                my $yml_path =
                                        # spent    22µs making 1 call to C4::Context::config
12                                        C4::Context->config('intranetdir') . '/admin/columns_settings.yml';
13
14  1   10µs    1   8µs                 my $cache  = Koha::Cache->get_instance();
                                        # spent     8µs making 1 call to Koha::Cache::get_instance
15  1   6µs     1   386µs               my $yaml = $cache->get_from_cache('ColumnsSettingsYaml');
                                        # spent   386µs making 1 call to Koha::Cache::get_from_cache
16  1   3µs                             unless ($yaml) {
17                                          $yaml = eval { YAML::LoadFile($yml_path) };
18                                          warn "############################ Not In Cache ############################" if $ENV{DEBUG};
19                                          warn "ERROR: the yaml file for DT::ColumnsSettings is not correctly formated: $@"
20                                            if $@;
21                                          $cache->set_in_cache('ColumnsSettingsYaml', $yaml, { expiry => 3600 });
22                                      }
23                                      else {
24  1   2µs                                 warn "############################ In Cache ############################" if $ENV{DEBUG};
25                                      }
26  1   9µs                             return $yaml;
27                                  }
```

We can see that we saved a lot of time (almost ~500ms).

Test Plan (Plack)

Before applying the patch

- Launch Plack :

ADMIN:

```
plackup --access-log /tmp/plack_intra_access.log --reload --port 5001
~/Documents/Koha/intranet.psgi
```

- Go to the circulation page

```
http://localhost:5001/cgi-bin/koha/circ/circulation.pl
```

- Check out a patron

At first launch you can see in **/tmp/plack_intra_access.log** that you are not using cache.

```
############################## Not In Cache ##############################
```

- Open NYTProf

- Select **C4/Utils/DataTables/ColumnSettings.pm** from the drop-down list.

- Click on “line”

Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
1	1	1	187µs	287ms	C4::Utils::DataTables::ColumnsSettings::get_columns
1	1	1	171µs	281ms	C4::Utils::DataTables::ColumnsSettings::get_yaml
1	1	1	69µs	90µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@4
1	1	1	62µs	207µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@3
1	1	1	26µs	35µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@7
1	1	1	26µs	33µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@8
1	1	1	24µs	59µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@5
1	1	1	24µs	29µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@6
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:45]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:61]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::get_modules
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::update_columns

Here we can see that **get_yaml** takes ~300ms.

- Click on “get_yaml”

```
10                                                  # spent 281ms (171µs+280) within C4::Utils::DataTables::ColumnsSettings::get_yaml which was c
                                                    #    once (171µs+280ms) by C4::Utils::DataTables::ColumnsSettings::get_columns at lin
                                                    sub get_yaml {
11   1   32µs   1   12µs                               my $yml_path =
                                                        # spent    12µs making 1 call to C4::Context::config
12                                                        C4::Context->config('intranetdir') . '/admin/columns_settings.yml';
13
14   1   21µs   1   6µs                                my $cache  = Koha::Cache->get_instance();
                                                        # spent     6µs making 1 call to Koha::Cache::get_instance
15   1   1.38ms   1   1.36ms                           my $yaml = $cache->get_from_cache('ColumnsSettingsYaml');
                                                        # spent  1.36ms making 1 call to Koha::Cache::get_from_cache
16   1   6µs                                          unless ($yaml) {
17   2   272µs   1   275ms                                $yaml = eval { YAML::LoadFile($yml_path) };
                                                            # spent   275ms making 1 call to YAML::LoadFile
18   1   1.22ms   1   1.18ms                               warn "############################ Not In Cache ############################" if $EI
                                                            # spent  1.18ms making 1 call to CGI::Compile::__ANON__
19   1   1µs                                              warn "ERROR: the yaml file for DT::ColumnsSettings is not correctly formated: $@"
20                                                            if $@;
21   1   2.40ms   1   2.38ms                               $cache->set_in_cache('ColumnsSettingsYaml', $yaml, { expiry => 3600 });
                                                            # spent  2.38ms making 1 call to Koha::Cache::set_in_cache
22                                                      }
23                                                      else {
24                                                          warn "############################ In Cache ############################" if $ENV{DI
25                                                      }
26   1   10µs                                         return $yaml;
27                                                  }
```

We can see that all the time is spent in that single line :

```
$yaml = eval { YAML::LoadFile($yml_path) };
```

- Apply patch

After applying the patch

On the first run, you are not supposed to see any change (not yet in cache).

But at second run, it's supposed to be cached. We are going to check it.

- Reload Plack to make NYTProf reload

```
plackup --access-log /tmp/plack_intra_access.log --reload --port 5001
~/Documents/Koha/intranet.psgi
```

- Reload the page without touching anything else.

Now in **/tmp/plack_intra_access.log** we can see that we are using cache.

```
############################## In Cache ##############################
```

- Re-open NYTProf :

Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
1	1	1	154µs	6.27ms	C4::Utils::DataTables::ColumnsSettings::get_columns
1	1	1	70µs	690µs	C4::Utils::DataTables::ColumnsSettings::get_yaml
1	1	1	40µs	55µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@4
1	1	1	26µs	102µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@3
1	1	1	20µs	32µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@7
1	1	1	17µs	20µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@8
1	1	1	16µs	38µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@5
1	1	1	15µs	17µs	C4::Utils::DataTables::ColumnsSettings::BEGIN@6
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:45]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::__ANON__[:61]
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::get_modules
0	0	0	0s	0s	C4::Utils::DataTables::ColumnsSettings::update_columns

10					`# spent 690µs (70+620) within C4::Utils::DataTables::ColumnsSettings::get_yaml which was calle`
					`#  once (70µs+620µs) by C4::Utils::DataTables::ColumnsSettings::get_columns at line`
					`sub get_yaml {`

11	1	28µs	1	10µs	`  my $yml_path =` `  # spent  10µs making 1 call to C4::Context::config`
12					`    C4::Context->config('intranetdir') . '/admin/columns_settings.yml';`
13					
14	1	21µs	1	6µs	`  my $cache  = Koha::Cache->get_instance();` `  # spent  6µs making 1 call to Koha::Cache::get_instance`
15	1	599µs	1	588µs	`  my $yaml = $cache->get_from_cache('ColumnsSettingsYaml');` `  # spent  588µs making 1 call to Koha::Cache::get_from_cache`
16	1	3µs			`  unless ($yaml) {`
17					`    $yaml = eval { YAML::LoadFile($yml_path) };`
18					`    warn "############################ Not In Cache ############################" if $EN`
19					`    warn "ERROR: the yaml file for DT::ColumnsSettings is not correctly formated: $@"`
20					`      if $@;`
21					`    $cache->set_in_cache('ColumnsSettingsYaml', $yaml, { expiry => 3600 });`
22					`  }`
23					`  else {`
24	1	33µs	1	16µs	`    warn "############################ In Cache ############################" if $ENV{DE` `    # spent  16µs making 1 call to CGI::Compile::__ANON__`
25					`  }`
26	1	10µs			`  return $yaml;`
27					`}`

We can see that we save a lot of time (almost ~300ms).