A simple web templating system for TCL using C

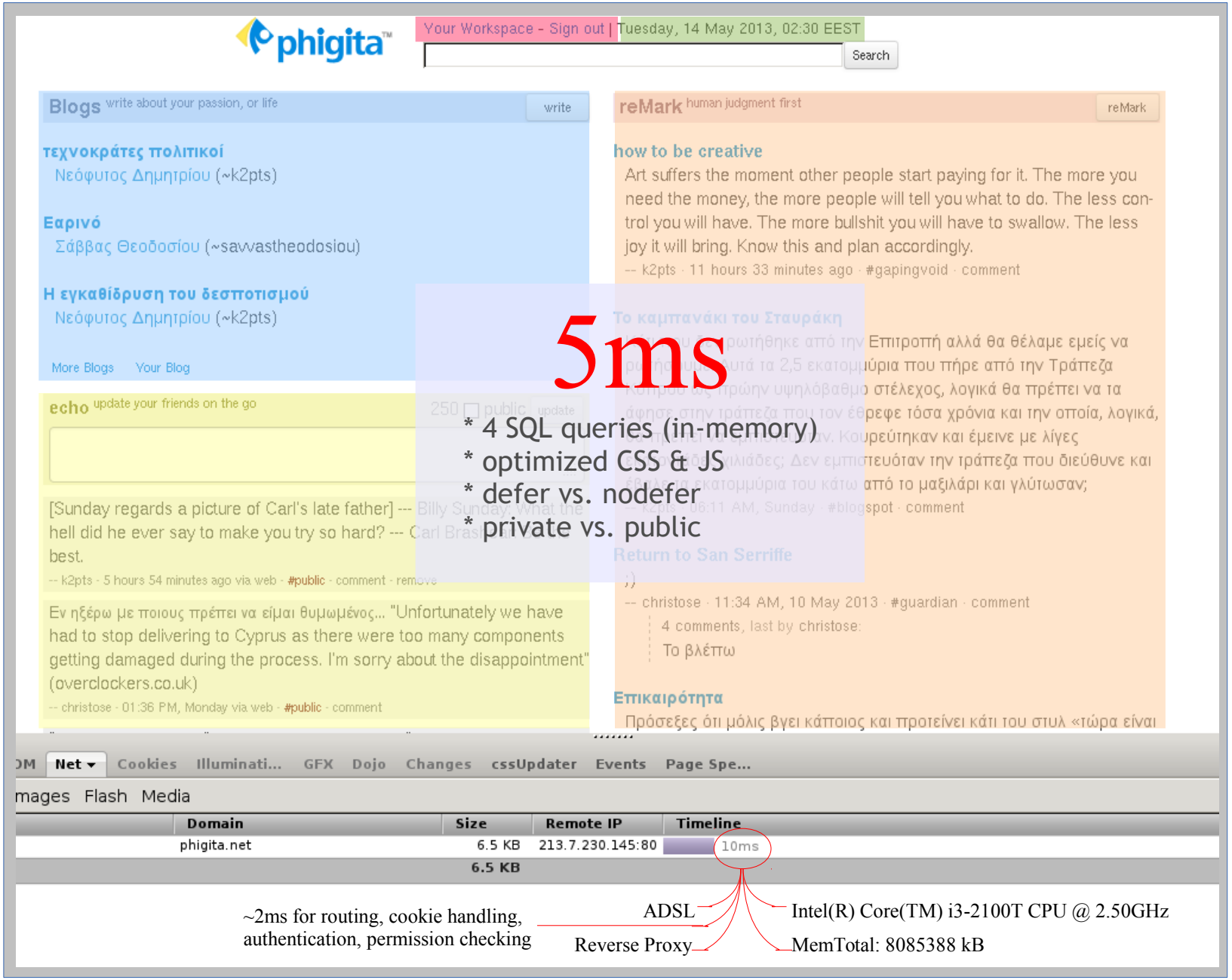
Neophytos Demetriou
neophytos@gmail.com

EuroTCL 2013
(Munich)

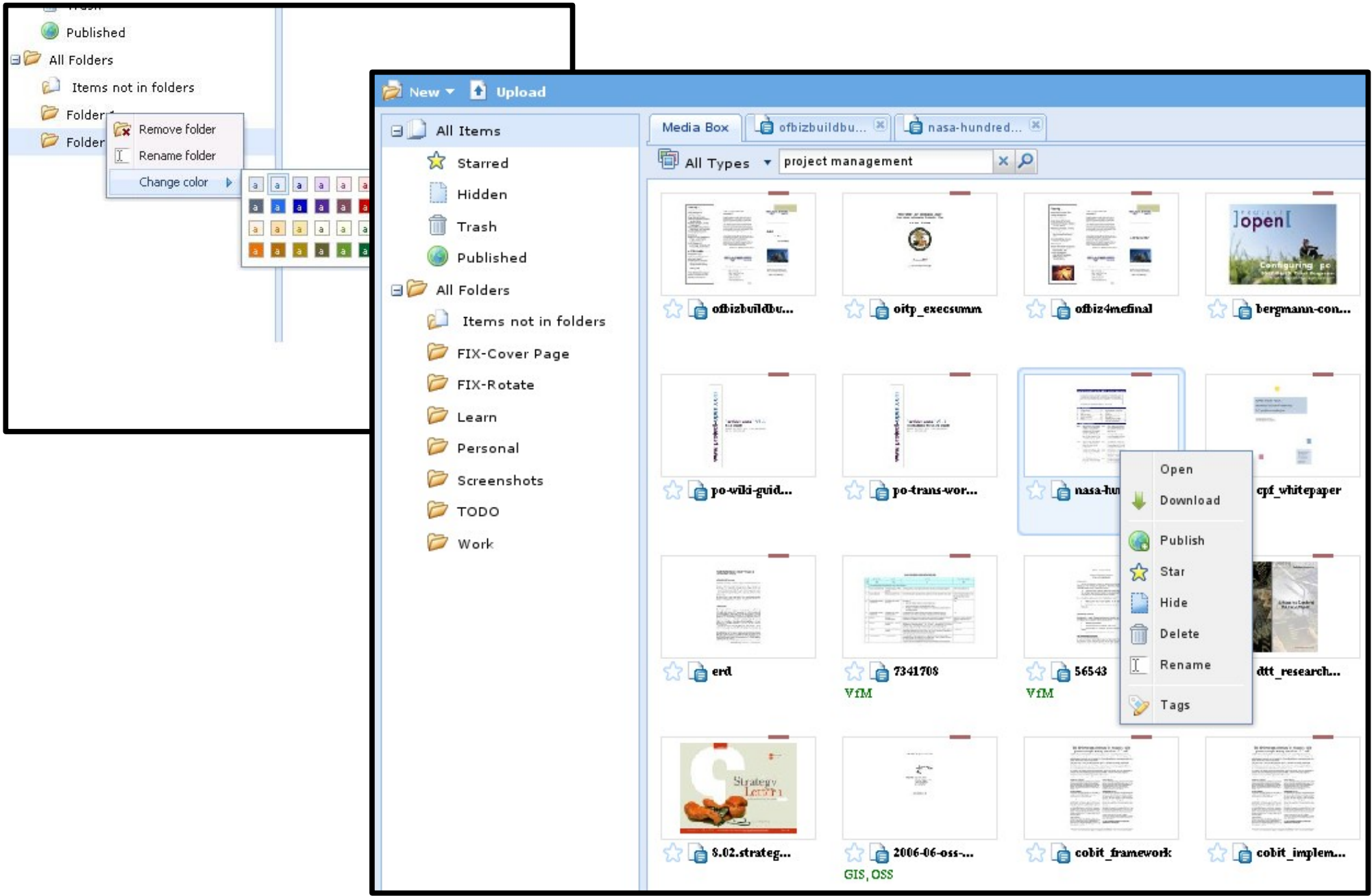
Overview

- NOT a TCL to C compiler
- NOT a reimplementation of the runtime system
- => transforms tDOM pages into C code
- => retains the simplicity of TCL
- => eliminates redundant processing
- Bottomline: It's fast. We'll try to explain why.

How fast

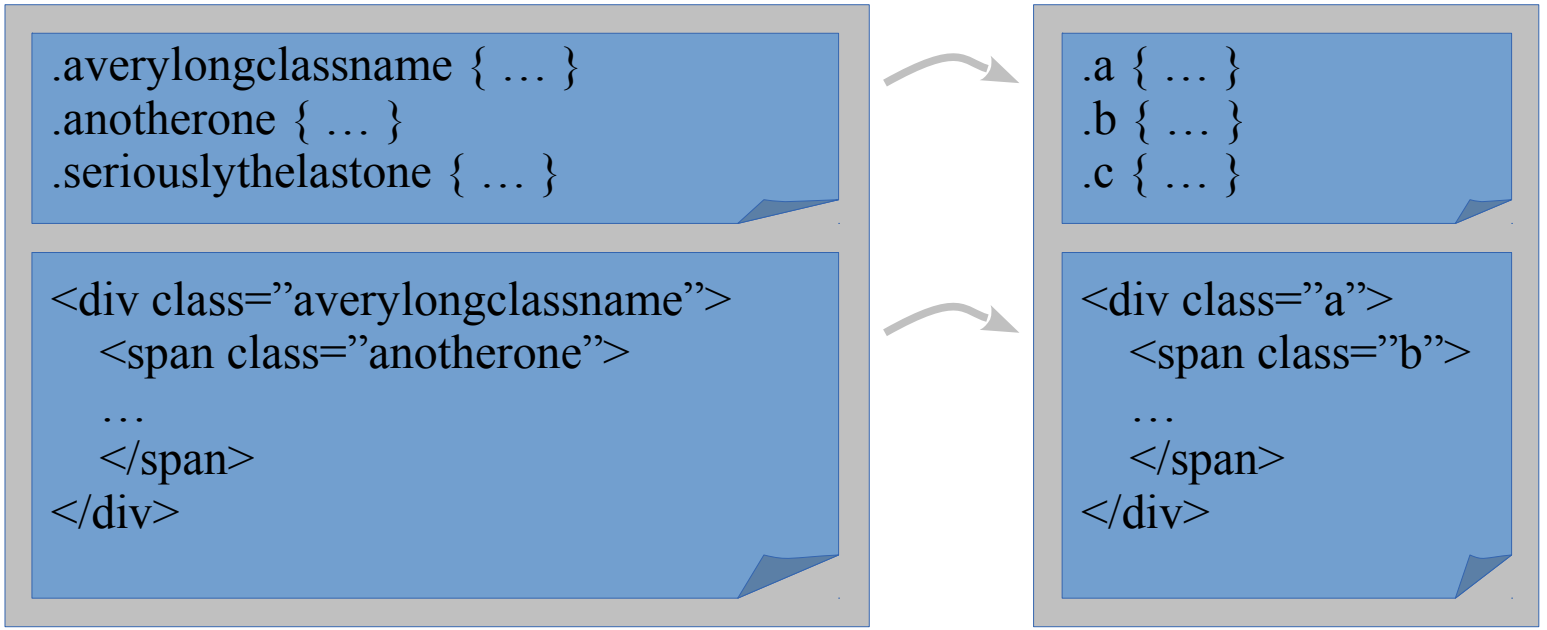

phigita
Your Workspace - Sign out | Tuesday, 14 May 2013, 02:30 EEST
Search
Blogs write about your passion, or life
write
τεχνοκράτες πολιτικοί
Νεόφυτος Δημητρίου (~k2pts)
Εαρινό
Σάββας Θεοδοσίου (~savvastheodosiou)
Η εγκαθίδρυση του δεσποτισμού
Νεόφυτος Δημητρίου (~k2pts)
More Blogs
Your Blog
echo update your friends on the go
[Sunday regards a picture of Carl's late father] --- Billy Sunday, what the hell did he ever say to make you try so hard? --- Carl Brashear best.
-- k2pts · 5 hours 54 minutes ago via web · #public · comment · remove
Εν ηξέρω με ποιους πρέπει να είμαι θυμωμένος... "Unfortunately we have had to stop delivering to Cyprus as there were too many components getting damaged during the process. I'm sorry about the disappointment" (overclockers.co.uk)
-- christose · 01:36 PM, Monday via web · #public · comment
reMark human judgment first
reMark
how to be creative
Art suffers the moment other people start paying for it. The more you need the money, the more people will tell you what to do. The less control you will have. The more bullshit you will have to swallow. The less joy it will bring. Know this and plan accordingly.
-- k2pts · 11 hours 33 minutes ago · #gapingvoid · comment
Το καμπανάκι του Σταυράκη
Επιτροπή αλλά θα θέλαμε εμείς να
μύρια που πήρε από την Τράπεζα
στέλεχος, λογικά θα πρέπει να τα
ρεφε τόσα χρόνια και την οποία, λογικά,
ρεύτηκαν και έμεινε με λίγες
τευόταν την τράπεζα που διεύθυνε και
από το μαξιλάρι και γλύτωσαν;
spot · comment
Return to San Serriffe
;)
-- christose · 11:34 AM, 10 May 2013 · #guardian · comment
4 comments, last by christose:
Το βλέπω
Επικαιρότητα
Πρόσεξες ότι μόλις βγει κάποιος και προτείνει κάτι του στυλ «τώρα είναι
5ms
* 4 SQL queries (in-memory)
* optimized CSS & JS
* defer vs. nodefer
* private vs. public
Net
Cookies
Illuminati...
GFX
Dojo
Changes
cssUpdater
Events
Page Spe...
mages
Flash
Media
Domain
Size
Remote IP
Timeline
phigita.net
6.5 KB
213.7.230.145:80
10ms
6.5 KB
~2ms for routing, cookie handling, authentication, permission checking
ADSL
Reverse Proxy
Intel(R) Core(TM) i3-2100T CPU @ 2.50GHz
MemTotal: 8085388 kB

Previous Effort: XOTCL to JS



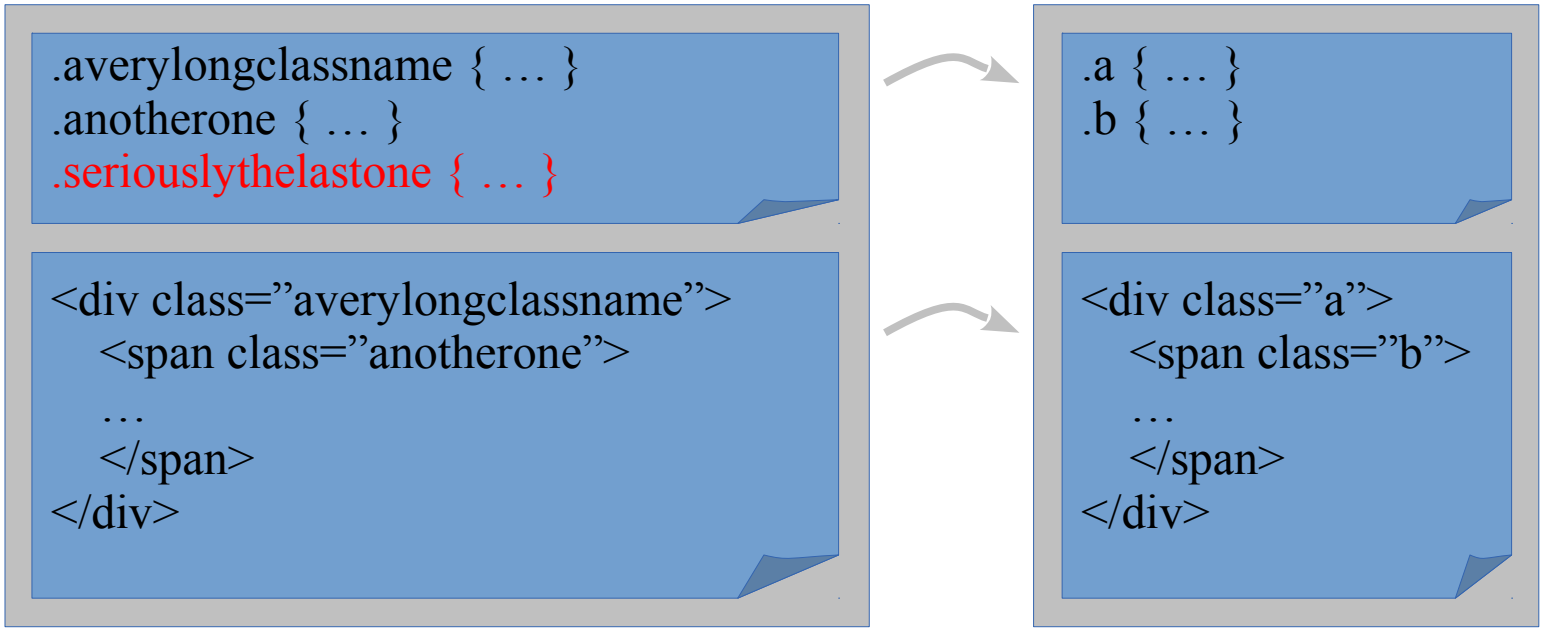
Motivating Example: Renaming CSS class names

- Renaming CSS class names
  - helps reduce the size of the CSS
  - class names must also be renamed in HTML and JS
  - waste of processing time if on-the-fly



Motivating Example: Removing unused CSS

- Removing unused CSS selectors
  - most tools inspect generated HTML
  - selector marked as used if it matches an element
  - waste of processing time if on-the-fly



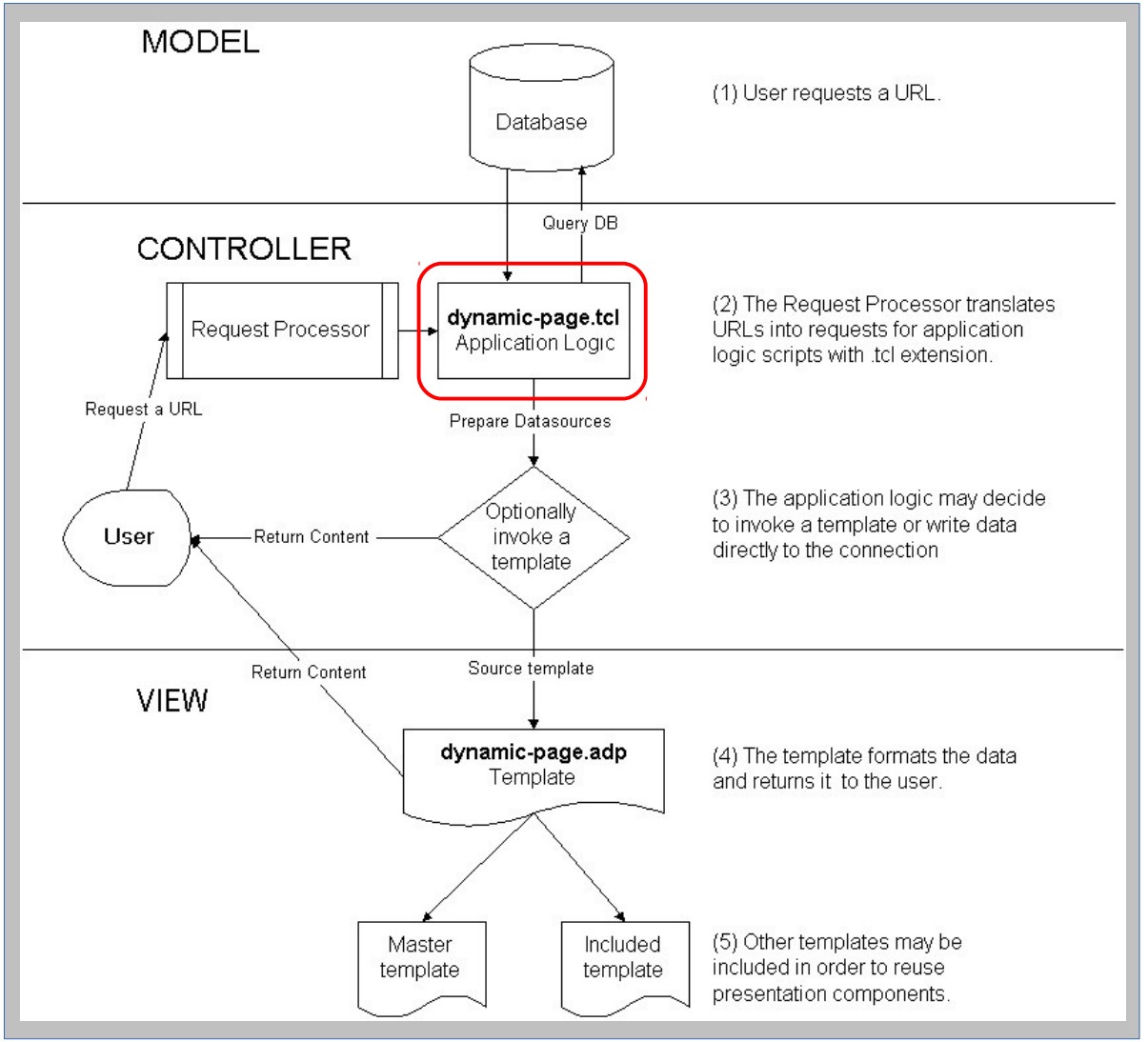
The Gist

- savings to be made at runtime if pre-processed
- something vs. nothing
  - on-the-fly compression takes time (at runtime)
  - compile-time compression saves time (at runtime)
- not convinced?
  - inline vs. external style sheet

```
# Why do this at runtime?

if { [string bytelength ${css}] < 8192 } {
    append html \
        "<style type=text/css>${css}</style>"
} else {
    append html \
        "<link rel=stylesheet type=text/css href=${url}>"
}
```

Problem is...

- Rewriting at compile-time is tricky, e.g.:




```
# .tcl file

if { [registered_user_p] }  {
  set msg “<div class=blue>hi</div>”
} else {
  set msg “<div class=red>bye</div>”
}
```

```
<!-- .adp file -->

<html>
  <body class=”something”>
    Message: @msg@
  </body>
</html>
```

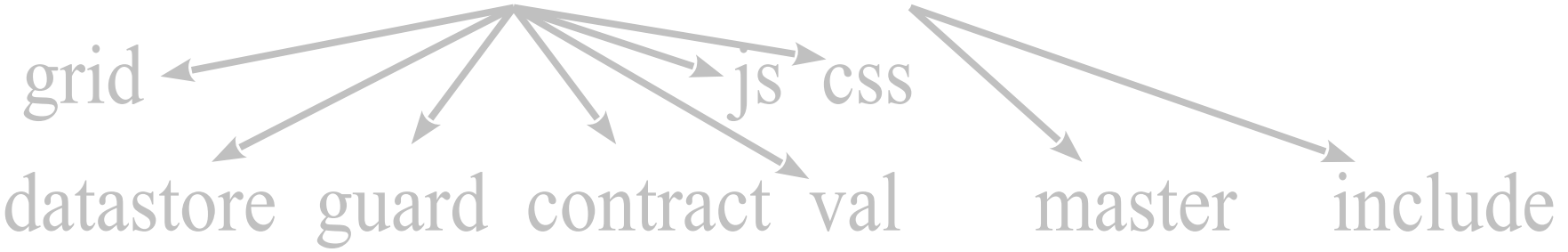
OpenACS Templating System Diagram

Structure of a Page: phigita templating system

page = application logic + presentation
= tDOM document, i.e. tree of widgets
= HTML + special markup tags + bindings

---

widget = reusable or common components



---

special tag = vars + conditional insertion + iteration

Composite Page: what it looks like

The root page includes the main content and wraps itself in the master.

Tabs inserted with the <include> tag.

Active tab passed as attribute to <include>

The root page specifies a descendant node (via x-master-renderTo) to be rendered by the master template.

Multiple blog posts that are not visible in the screenshot.

Composite Page: spec as tDOM script

```
val -id “context_user_id” {
    return [ad_conn context_user_id]
}

datastore \
    -id “userdata” \
    -singleton “true” \
    -from “users” \
    -where “user_id = :context_user_id”

guard {
    return @{userdata:rowcount}
}

datastore \
    -id “blogdata” \
    -scope “context_user” \
    -from “blog_posts” \
    -order “entry_date desc”

set pagetitle “@{userdata.full_name} > Blog”
```

```
master -src “shared-master.inc” -title $pagetitle {
    include -src “shared-tabs.inc” -active “blog” {
        tpl -for “blogdata” {
            div -class “post” -href “@{id}” {
                a -class “title” {
                    t @{title}
                }
                div -class “content” {
                    t @{content:noquote}
                }
            }
        }
    }
}
```

Composite Page: spec as tDOM script - explained

set ::__data__(context_user_id) [ad_conn context_user_id]

```
val -id “context_user_id” {
    return [ad_conn context_user_id]
}

datastore \
    -id “userdata” \
    -singleton “true” \
    -from “users” \
    -where “user_id = :context_user_id”
WHERE user_id=[dbquote $::__data__(context_user_id)]
guard {
    return @{userdata:rowcount}
}
$::__data__(userdata,rowcount)
COMMA
datastore \
    -id “blogdata” \
    -scope “context_user” \
    -from “blog_posts” \
    -order “entry_date desc”

set pagetitle “@{userdata.full_name} > Blog”
```

VALUE

retrieves single item

bind vars

SQL

ABORT if false

SQL

CONST

```
master -src “shared-master.inc” -title $pagetitle {
    include -src “shared-tabs.inc” -active “blog” {
        tpl -for “blogdata” {
            div -class “post” -href “@{id}” {
                a -class “title” {
                    t @{title}
                }
                div -class “content” {
                    t @{content:noquote}
                }
            }
        }
    }
}
```

What if table does not exist?
Why generate the SQL at runtime?
How caching works?

set ::__data__(blogdata) [$conn exec “SELECT * FROM **xo__u[ad_conn context_user_id].xo__blog_posts** ORDER BY entry_date desc”]

Composite Page: spec as markup

```
<val id="context_user_id">
   return [ad_conn context_user_id]
</val>

<datastore
   id="userdata"
   singleton="true"
   from="users"
   where="user_id = :context_user_id" />

<guard>return @{userdata:rowcount}</guard>

<datastore
   id="blogdata"
   scope="context_user"
   from="blog_posts"
   order="entry_date desc" />
```

```
<master src="shared-master.inc"
        title="@{userdata.full_name} > Blog">
  <include src="shared-tabs.inc" active="blog">
    <tpl for="blogdata">
      <div class="post" href="@{id}">
        <a class="title">
          @{title}
        </a>
        <div class="content">
          @{content:noquote}
        </div>
      </div>
    </tpl>
  </include>
</master>
```

1. Result of processing the tDOM script is a tDOM document
2. Widgets get converted into basic building blocks, i.e. tpl tags
3. tDOM document is transformed into C code

Transformation Order

- process <include> and <master> tags
- rewrite <contract> and <param> tags using <guard>
- prepare the rest of the widgets for the transformation
  - CSS/JS
  - Generate SQL queries for <datastore> tags
- transform resulting tDOM document to C code

Transformation Files

index.{mode}.{ino}.{mtime}.{c,so}

```
 70574 Jun 29 18:42 index.0.12993353.1372520506.c
 76941 Jun 29 18:42 index.0.12993353.1372520506.so
  5559 Jun  5 22:01 index.private_defer-compiled-0ED4375E66E0F774B02714B90D22AB6C7C3CEC4E.js
 16307 Jun  5 22:01 index.private_defer-map.js
 21247 Jun  5 22:01 index.private_defer-source-0ED4375E66E0F774B02714B90D22AB6C7C3CEC4E.js
  5229 Jun  5 22:01 index.private_nodefer-compiled-E0ECA20E03C929D6FE051C74FD122A47EF8F83F4.js
 14213 Jun  5 22:01 index.private_nodefer-map.js
 26075 Jun  5 22:01 index.private_nodefer-source-E0ECA20E03C929D6FE051C74FD122A47EF8F83F4.js
130388 Jun 29 18:42 index.tar.bz2
 68744 Jun 29 18:42 index.tdp_c
189398 Jun 29 18:42 index.tdp_css
    98 Jun 29 18:42 index.tdp_css_keep
  2846 Jun 29 18:42 index.tdp_css_map
125085 Jun 29 18:42 index.tdp_css_min
256419 Jun 29 18:42 index.tdp_css_min_dropped
 10923 Jun 29 18:42 index.tdp_css_min_final
   966 Jun 29 18:42 index.tdp_dep
   305 Jun 29 18:42 index.tdp_html
 21246 Jun 29 18:42 index.tdp_js_private_defer
  5585 Jun 29 18:42 index.tdp_js_private_defer_min
 26074 Jun 29 18:42 index.tdp_js_private_nodefer
  5255 Jun 29 18:42 index.tdp_js_private_nodefer_min
   726 Jun 29 18:42 index.tdp_spec
 30274 Jun 29 18:42 index.tdp_spec_final
 14192 Jun 29 18:42 index.tdp_spec_inc
 18216 Jun 29 18:42 index.tdp_spec_ini
```

- Twitter Bootstrap CSS (190 kB)
- renaming map of CSS class names
- CSS3 rules expanded
- removed unused CSS selectors
- minimized CSS (11 kB)

- some JS is deferred
- other JS is not deferred
- minimized JS
- aware of CSS class renaming
- no JS for non-registered users here

- spec as markup after each phase

Transformation Notes

- Concatenation
  - static and dynamic parts
- Contextual data source
  - each dynamic part is assigned a data source
  - global context is a TCL array ::__data__
  - current and parent context
- Error Checking
- Developer Mode
  - No need to restart to load changes
  - Debugging Symbols and Info
- Production Mode
  - Restart required to load changes

- Test Setup
  - NaviServer 4.99.6
  - TCL 8.5.14
  - NSF 2.0b5 for Persistence Layer
  - tDOM 0.8.3-20120716
  - Critcl v1
  - Closure Stylesheets for CSS
  - Closure Compiler for JS
  - PostgreSQL 9.0.4
- Build Files
  - distribute without source possible

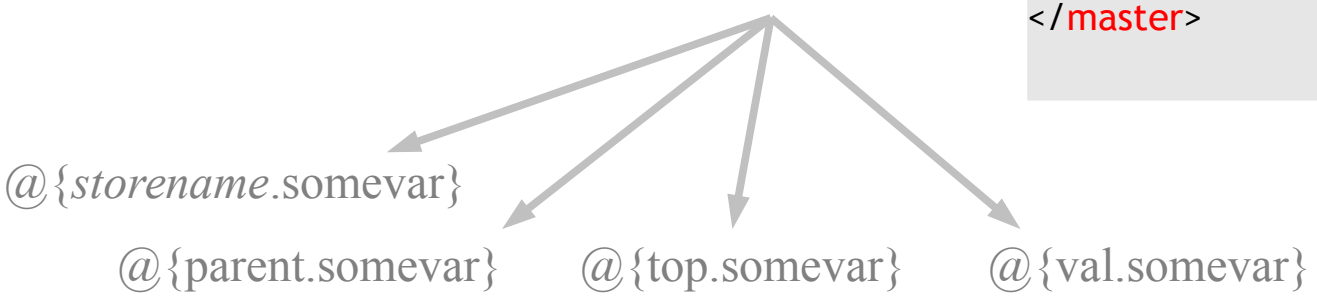
Transformation Result: C code sample - edited

```
Tcl_DStringAppend(dsPtr,"</head><body><div class=\"r\">
                         <a href=\"http://www.phigita.net/\">[...]</a></div>",  220);

if ( getbool_0(interp, global_objects, block0_o0, global_objects[OBJECT_VARNAME_registered_p]) ) {
   Tcl_DStringAppend(dsPtr,"<a href=\"[...]\">Your Workspace</a> - <a href=\"[...]\">Sign out</a>",117);
}
else {
   Tcl_DStringAppend(dsPtr,"<a href=\"[...]\">Sign in</a>",55);
}

Tcl_DStringAppend(dsPtr," | ",3);

append_0(interp, global_objects, block0_o0, global_objects[OBJECT_VARNAME_localtime], dsPtr,0);

 Tcl_DStringAppend(dsPtr,"[...]",256);
```

Transformation Example: variable substitution

- placeholders for dynamic data
- scoped variables
  - global context
  - current context
  - parent context

```
<master src="shared-master.inc"
        title="@{userdata.full_name}'s Blog">
  <include src="shared-tabs.inc" active="blog">
    <tpl for="blogdata">
      <div class="post" href="@{id}">
        <a class="title">
          @{title}
        </a>
        <div class="content">
          @{content:noquote}
        </div>
      </div>
    </tpl>
  </include>
</master>
```



Transformation Example: variable substitution

@{userdata.full_name}

```
append_1 /* nsfvar */ (
    interp,
    global_objects,
    Tcl_ObjGetVar2(
        interp,
        global_objects[OBJECT_DATA],
        global_objects[OBJECT_VARNAME_userdata],
        TCL_GLOBAL_ONLY
    ),
    global_objects[OBJECT_VARNAME_full_name],
    dsPtr,
    0
);
```

::__data__(userdata)

Fetch 'userdata' from the global data array.

In this case, 'userdata' is an XOTCL object.

Alternative definition of append_1exists that works with TCL dictionaries.

Then, get 'full_name' from the 'userdata' object, quote it, and append it to dsPtr (a pointer to a dstring that holds the resulting HTML).

Whether to quote the HTML or not.

Transformation Example: variable substitution

```
<tpl for=”registered_users”>
   @{full_name}
</tpl>
```

```
append_1 /* nsfvar */ (
    interp,
    global_objects,
    block1_o1,
    global_objects[OBJECT_VARNAME_full_name],
    dsPtr,
    0
);
```

an object in the list of 'registered_users'

Transformation Example: variable substitution

```
<val id="msg">
    return "something"
</val>
---
Hello World! Message is @{msg}
```

```
Tcl_DStringAppend(dsPtr,"Hello World! Message is ",24);

append_0 /* tclvar */ (
    interp,
    global_objects,
    block0_o0,
    global_objects[OBJECT_VARNAME_msg],
    dsPtr,
    0
);
```

global context
::__data__

append html "Hello World! Message is "
append html $::__data__(msg)

Transformation Example: iteration

```
<tpl for="blogdata">
   ...
</tpl>
```

```
Tcl_Obj *block1_listPtr0
   = getvar_0 /* tclvar */ (
        interp,
        global_objects,
        block0_o0,
        global_objects[OBJECT_VARNAME_blogdata]);
int rowcount_block1_o1, rownum_block1_o1;
Tcl_ListObjLength(interp, block1_listPtr0, &rowcount_block1_o1);
Tcl_Obj *block1_o1;
for (rownum_block1_o1=0; rownum_block1_o1<rowcount_block1_o1; rownum_block1_o1++) {
  Tcl_ListObjIndex(interp, block1_listPtr0, rownum_block1_o1, &block1_o1);
   /* … */
}
```

```
set block1_list $::__data__(blogdata)
```

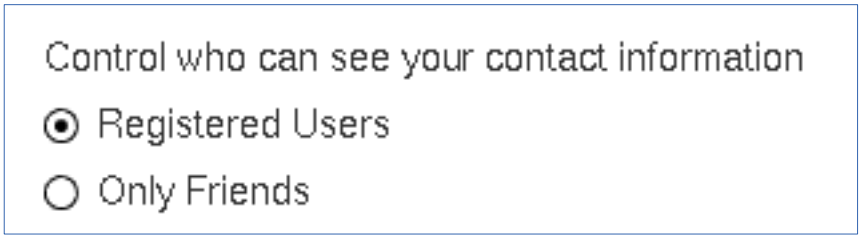
```
foreach block1_o1 ${block1_list} {
   # do something with $block1_o1
}
```

Transformation Example: conditional insertion

```
<tpl for="navigation_tabs">
  <tpl if="@{val.selectedtab} eq @{_.0}">
    <a class="selectedtab" href="@{_.1}">
      @{_.2:noquote}
    </a>
  </tpl>
</tpl>
```

```
if (  strcmp_eq (
        Tcl_GetString(
           getvar_0 /* tclvar */(
                interp,
                global_objects,
                global_objects[OBJECT_DATA],
                global_objects[OBJECT_VARNAME_selectedtab])
        ),
        Tcl_GetString(getvar_obj_element(interp,block2_o1,0)))
) {
                Tcl_DStringAppend(dsPtr,"<a class=\"yf\" href=\"",20);
                append_obj_element(interp,block2_o1, 1, dsPtr, 0);
                Tcl_DstringAppend(dsPtr,"\">",2);
                append_obj_element(interp,block2_o1, 2, dsPtr, 1);
                Tcl_DStringAppend(dsPtr,"</a>",4);

}
```

Bindings




```
<form action=”save-settings”
        x-bind=”{
            formdata userdata
        }”>
    Control who can see your contact information
    <label class="radio">
        <input type=”radio” name="priv_contact_info"
           value=”5”>

         Registered Users
    </label>
    <label class="radio">
        <input type=”radio” name="priv_contact_info"
           value=”2”>

          Only Friends
        </label>
</form>
```

```
<form action=”save-settings”>
    Control who can see your contact information
    <label class="radio">
        <input type=”radio” name="priv_contact_info"
           value=”5”
           x-bind=”{
               checked {
                  ! @{userdata:rowcount}
                  || @{userdata.priv_contact_info} eq {5}
               }
           }”>

         Registered Users
    </label>
    <label class="radio">
        <input type=”radio” name="priv_contact_info"
           value=”2”
           x-bind=”{
               checked {
                  @{userdata:rowcount}
                  && @{userdata.priv_contact_info} eq {2}
               }
           }”>

          Only Friends
        </label>
</form>
```

Conclusion

- Templating System
  - simple and fast
  - clean syntax and clear semantics
- tDOM script
  - trivial to define, develop and maintain a DSL
- Persistence Layer
  - encapsulation of database logic
- Future Work
  - sub-page entry points
  - better error checking
  - insert/update/delete to C
  - alt ways of rendering

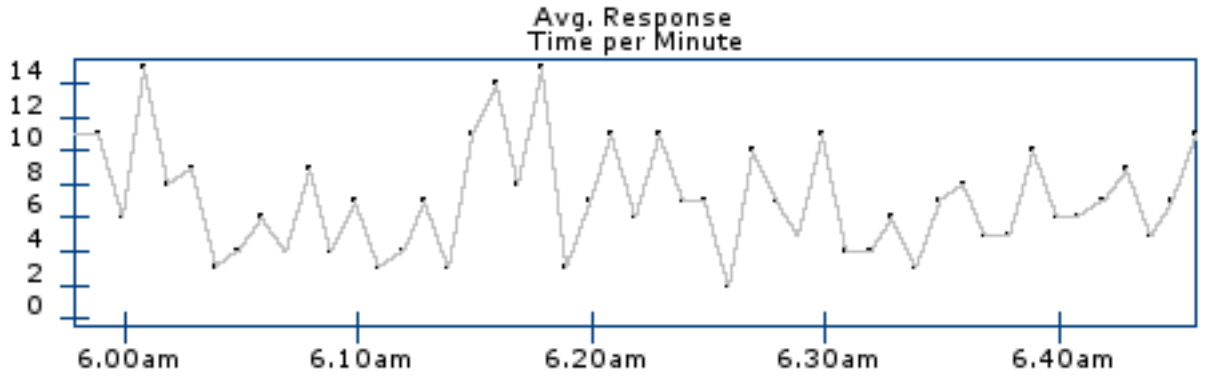
Also:
-cache “blog_posts_@{context_user_id}”
-where_if “...”
-extend { … }

Acknowledgements

- Univ. Prof. Dr. Gustaf Neumann
  Institute for Information Systems and New Media
  Vienna University of Economics and Business

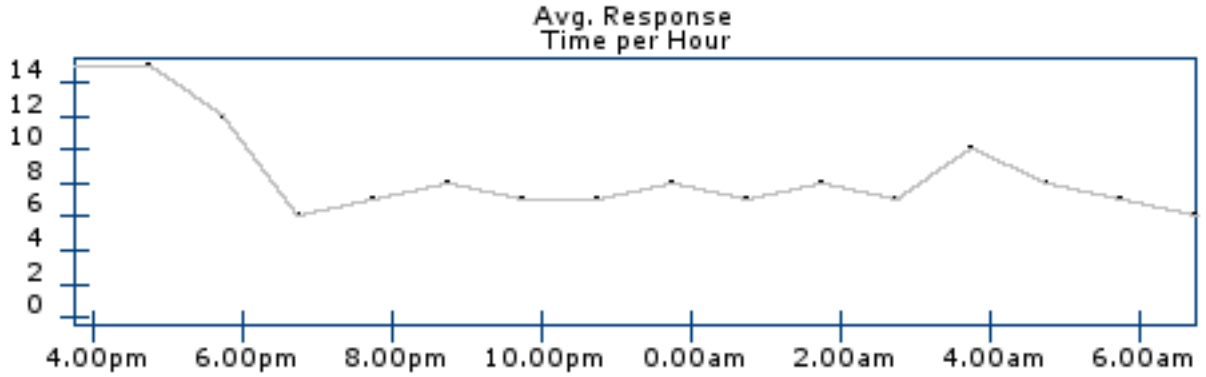
Thank You!

Avg. Response Time in milliseconds



Max

Mon Jul 01 15:32:57 UTC 2013	169
Tue Jul 02 03:11:57 UTC 2013	100
Mon Jul 01 16:37:57 UTC 2013	70
Mon Jul 01 23:08:57 UTC 2013	54
Mon Jul 01 16:35:57 UTC 2013	41
Mon Jul 01 20:25:57 UTC 2013	39



Max

Mon Jul 01 16:31:57 UTC 2013	15
Mon Jul 01 17:31:57 UTC 2013	12
Tue Jul 02 03:31:57 UTC 2013	10
Mon Jul 01 20:31:57 UTC 2013	8
Mon Jul 01 23:31:57 UTC 2013	8
Tue Jul 02 01:31:57 UTC 2013	8