



All your cores are
belong to us

Alexander Gounares
Concurix Corporation
March 21, 2013

The manycore era is here.



CATS : ALL YOUR BASE ARE BELONG
TO US.

“Zero Wing”, 1991





The manycore era is here.



The Manycore era is here now

The manycore era is here.



1997: THE FIRST INTEL® TERAFLUP COMPUTER
consisted of:

9,298 INTEL
PROCESSORS

and occupied:
72 SERVER
CABINETS

THE INTEL® XEON® PHI™ COPROCESSOR
will provide:

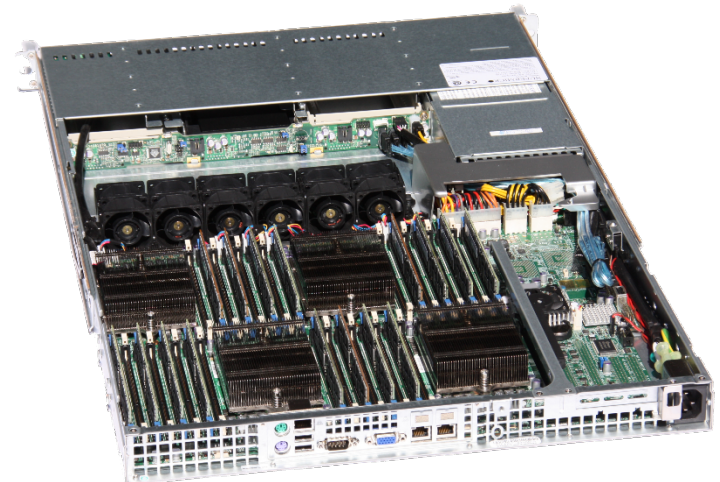
1 TERAFLUP OF
PERFORMANCE | **1** PCIe
SLOT



~\$4.2K

AMD Opteron family 15h
64 cores: 16 per chip x 4 sockets
Streaming SIMD extensions (SSE4)
128 GB RAM—512GB max
8 NUMA Domains with Hypertransport

~\$4.7K



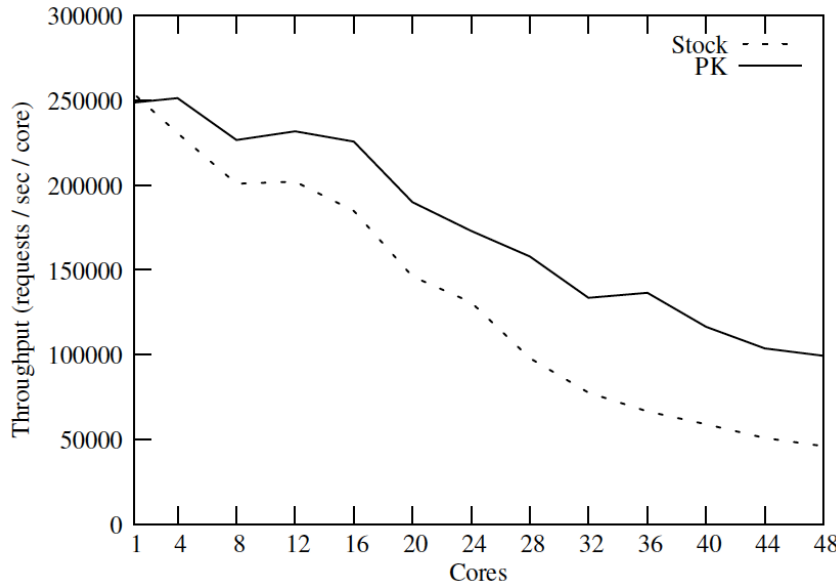


The manycore era is here.



Traditional software does not scale on manycore

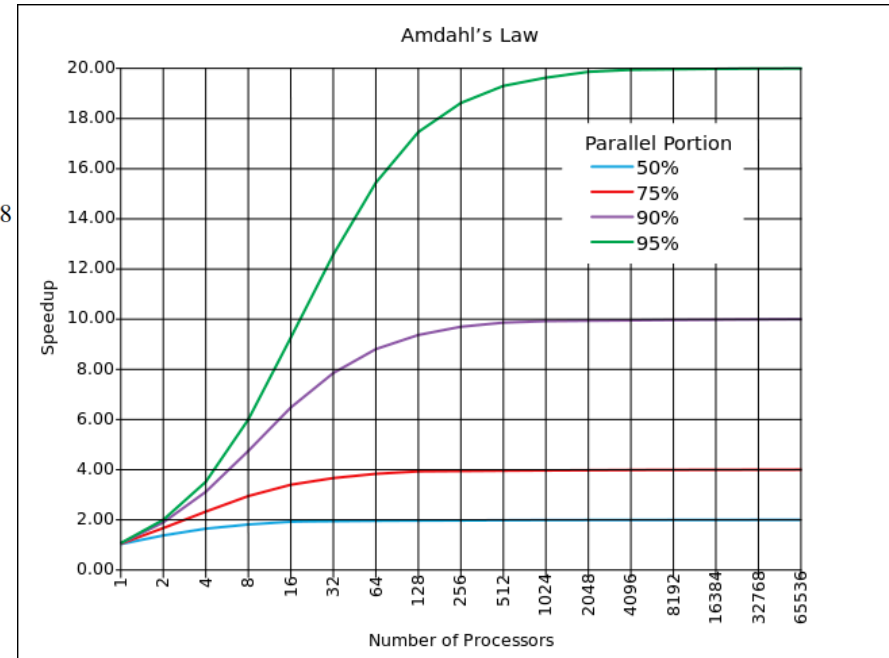
The manycore era is here.



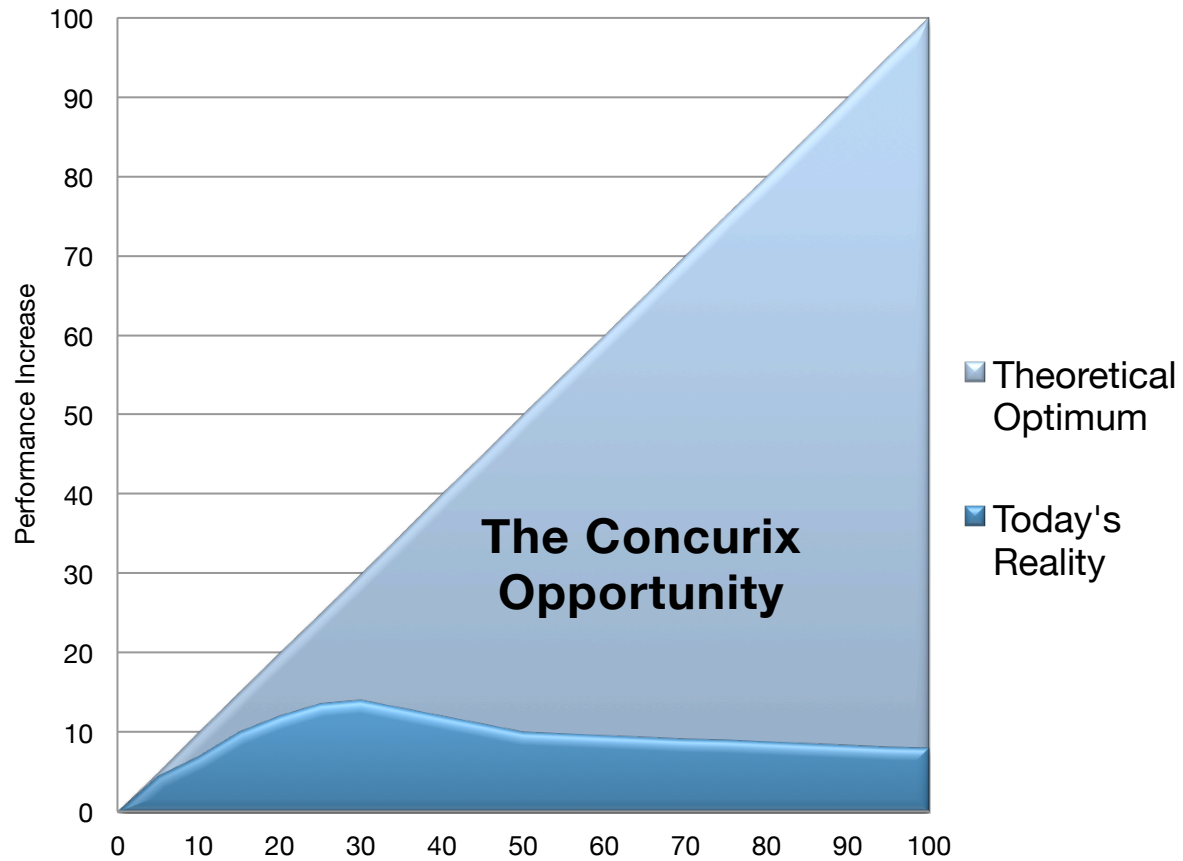
<http://pdos.csail.mit.edu/papers/linux:osdi10.pdf>

Figure 5: memcached throughput.

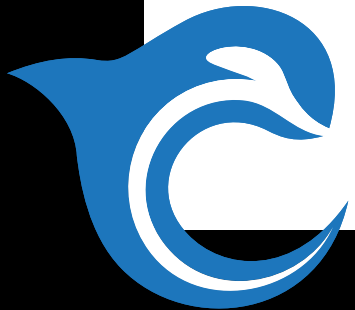
Locks in traditional O-O languages limit scalability per Amdahl's Law



The Concurix Opportunity: Realizing Moore's Law for software



The manycore era is here.



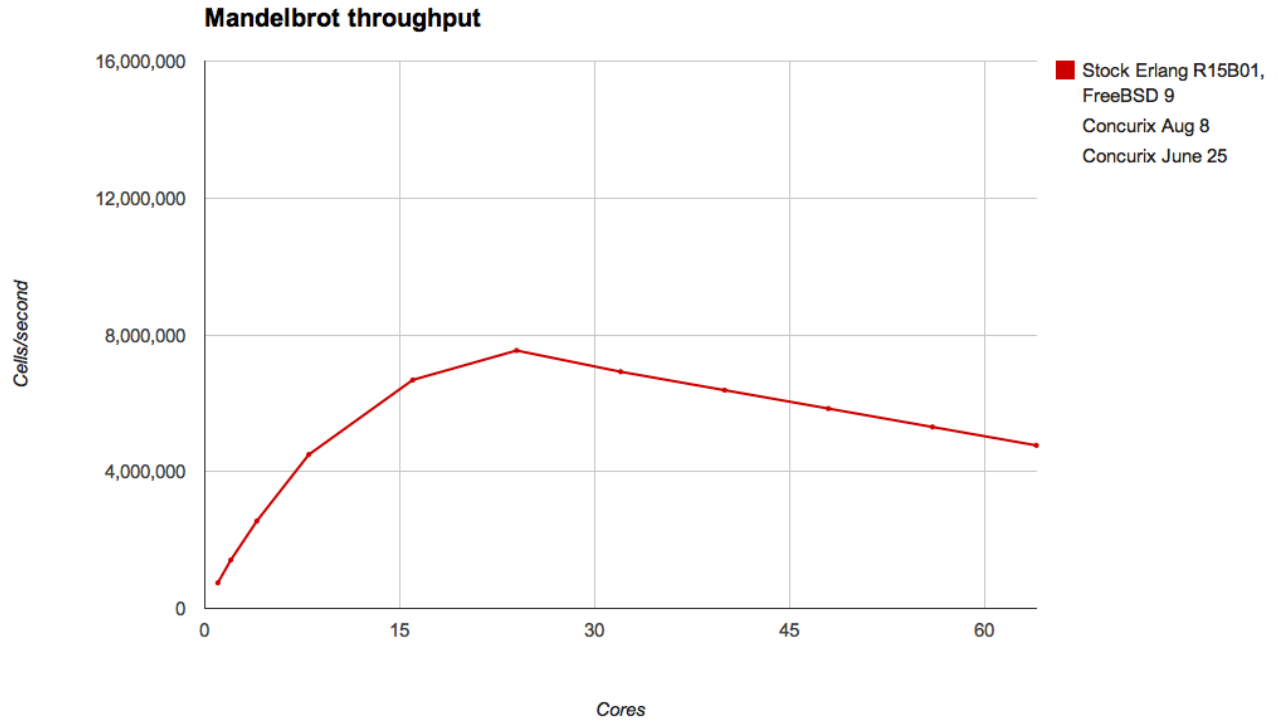
Erlang to the rescue!



The manycore era is here.



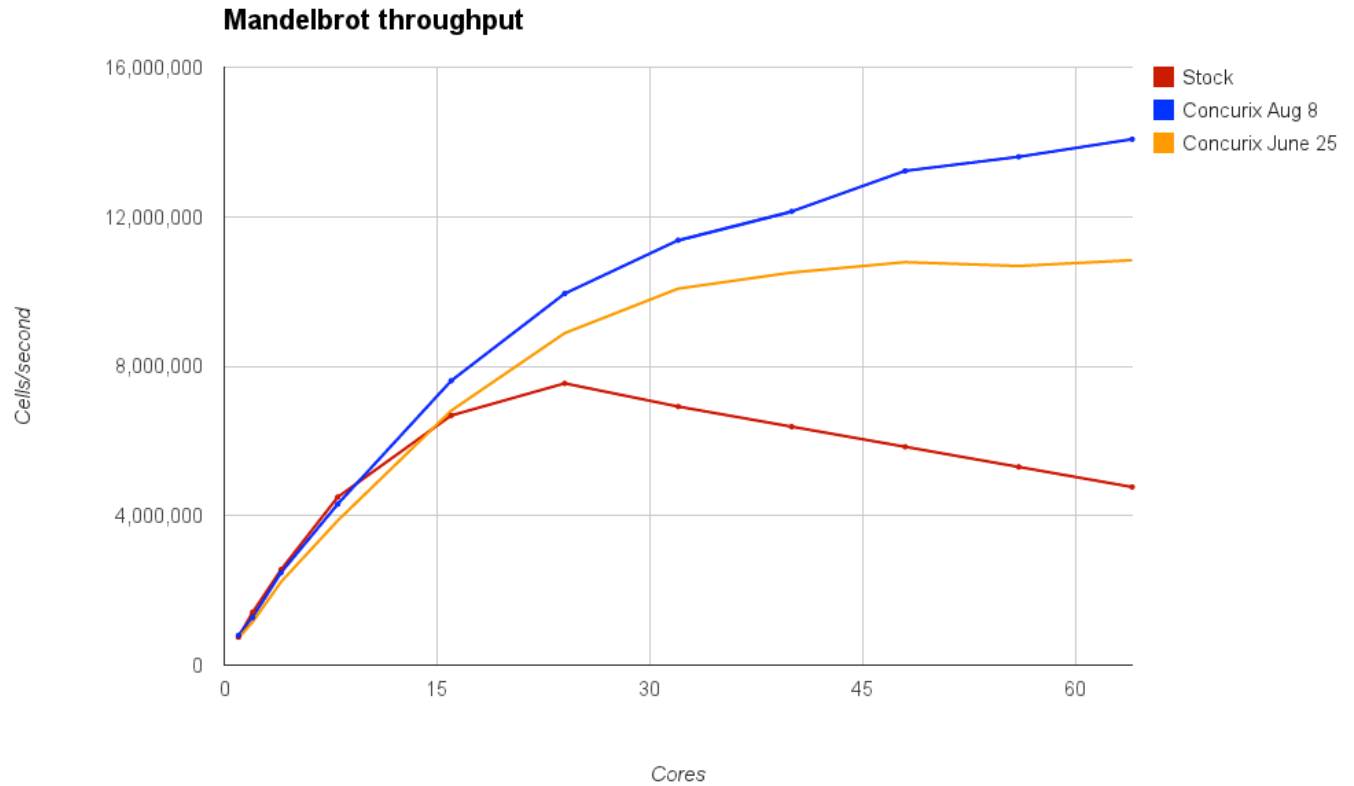
Almost...even very parallelizable workloads had trouble scaling



The manycore era is here.



After a bit of work....

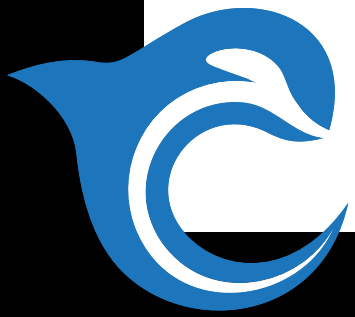
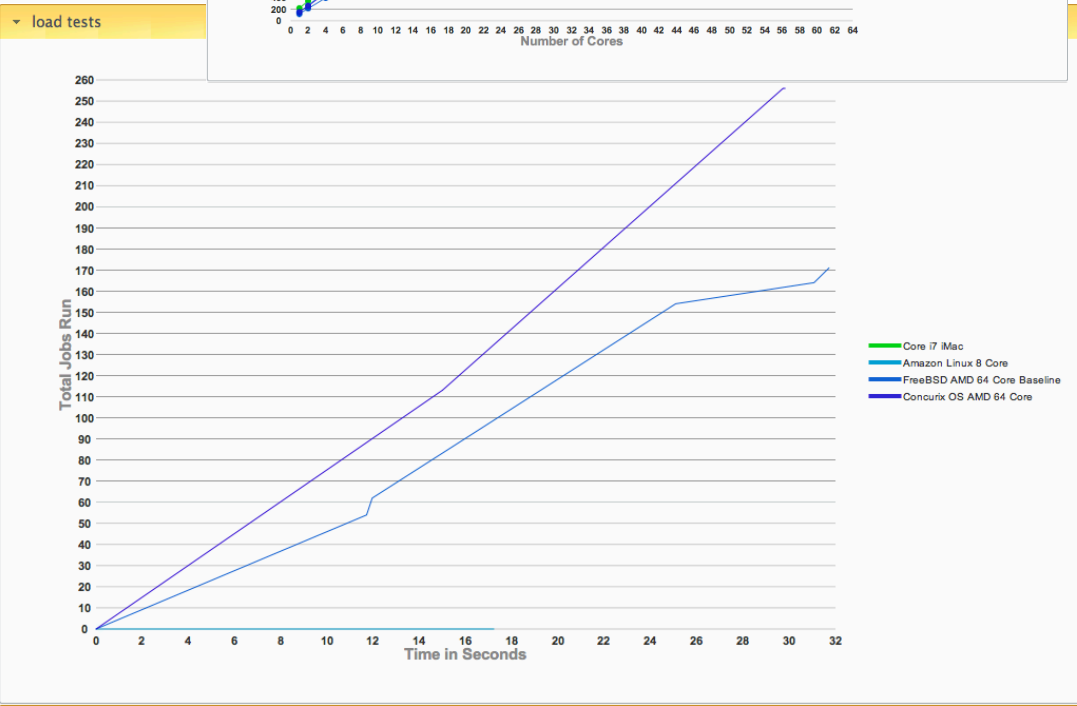
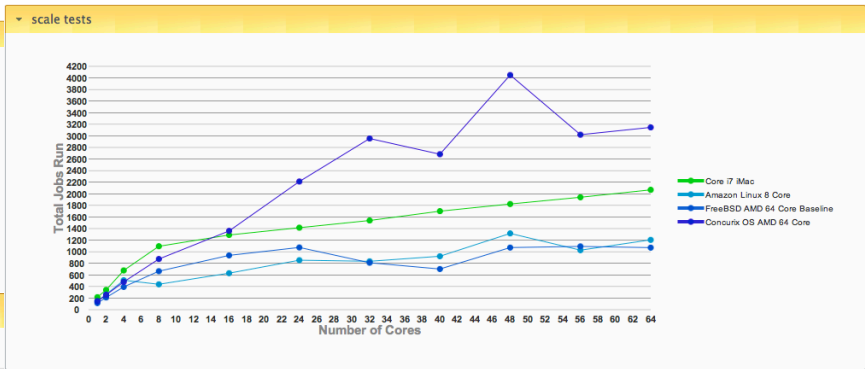
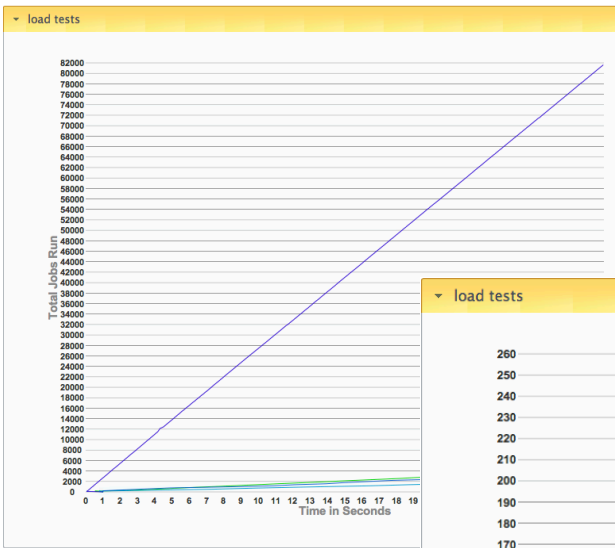


The manycore era is here.



Great results on simple benchmarks...

The manycore era is here.



Techniques and improvements

Technique	“Best Case Win” on 64 cores...
Removed locks in the garbage collector	~50% speedup
Tune per process memory usage	~50% speedup
Automatic Memoization	~40X --but extremely dependent on the app
Message-passing aware scheduler	~5% speedup
NUMA memory allocator	~50% speedup on interpreted, ~4% w/ HIPE

The manycore era is here.

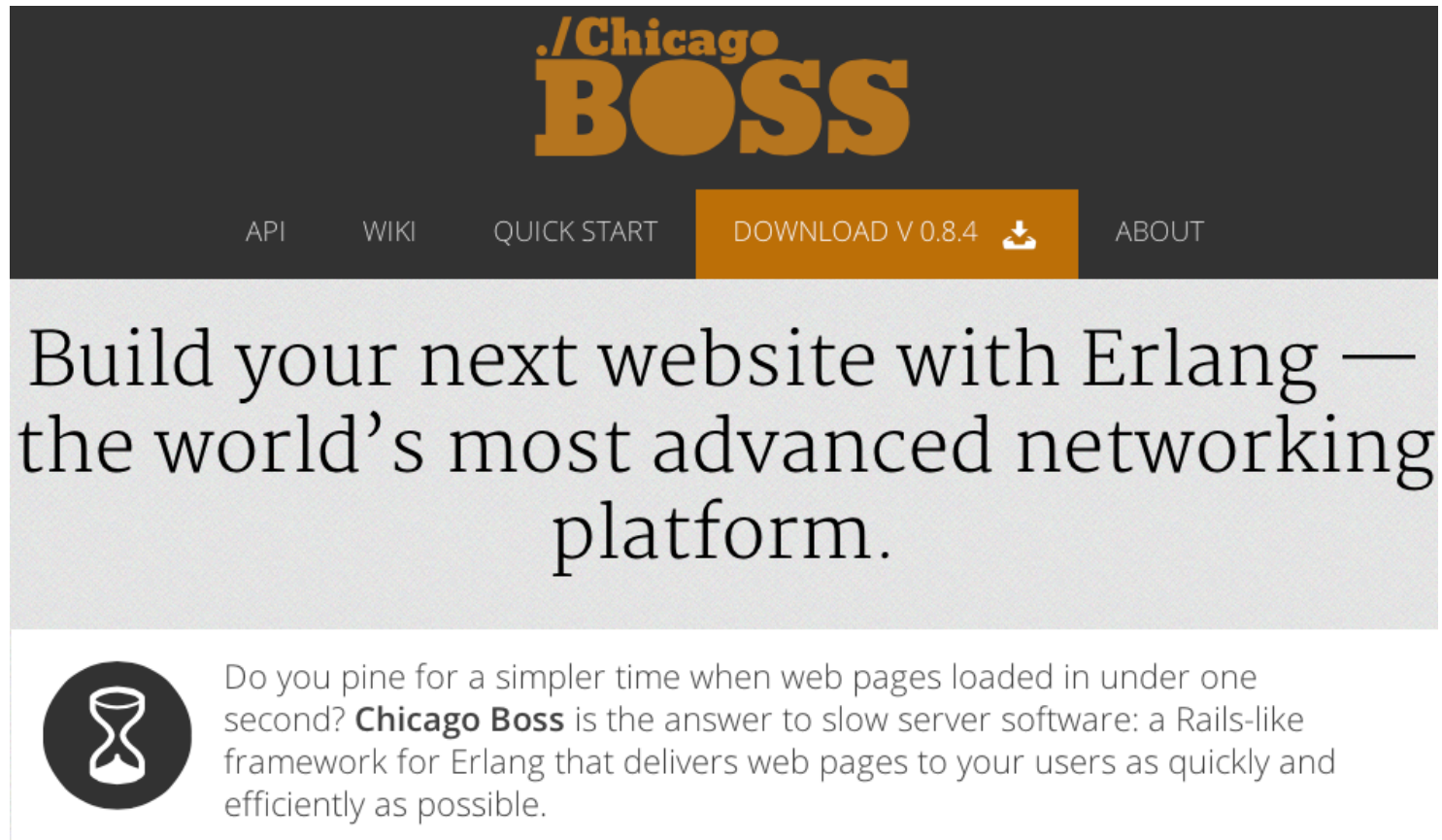


The manycore era is here.

Now something more complex....



ChicagoBoss!



The screenshot shows the ChicagoBoss website. At the top, the logo reads **./Chicago BOSS** in orange and white. Below the logo is a navigation bar with links for [API](#), [WIKI](#), [QUICK START](#), [DOWNLOAD V 0.8.4](#) (with a download icon), and [ABOUT](#). The main content area features a large heading: "Build your next website with Erlang — the world's most advanced networking platform." Below this is a section with an hourglass icon and the text: "Do you pine for a simpler time when web pages loaded in under one second? **Chicago Boss** is the answer to slow server software: a Rails-like framework for Erlang that delivers web pages to your users as quickly and efficiently as possible."

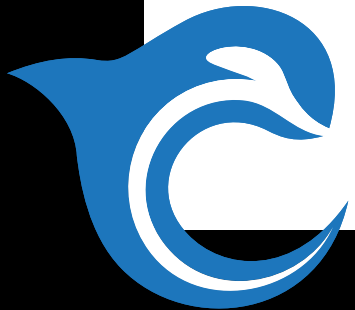
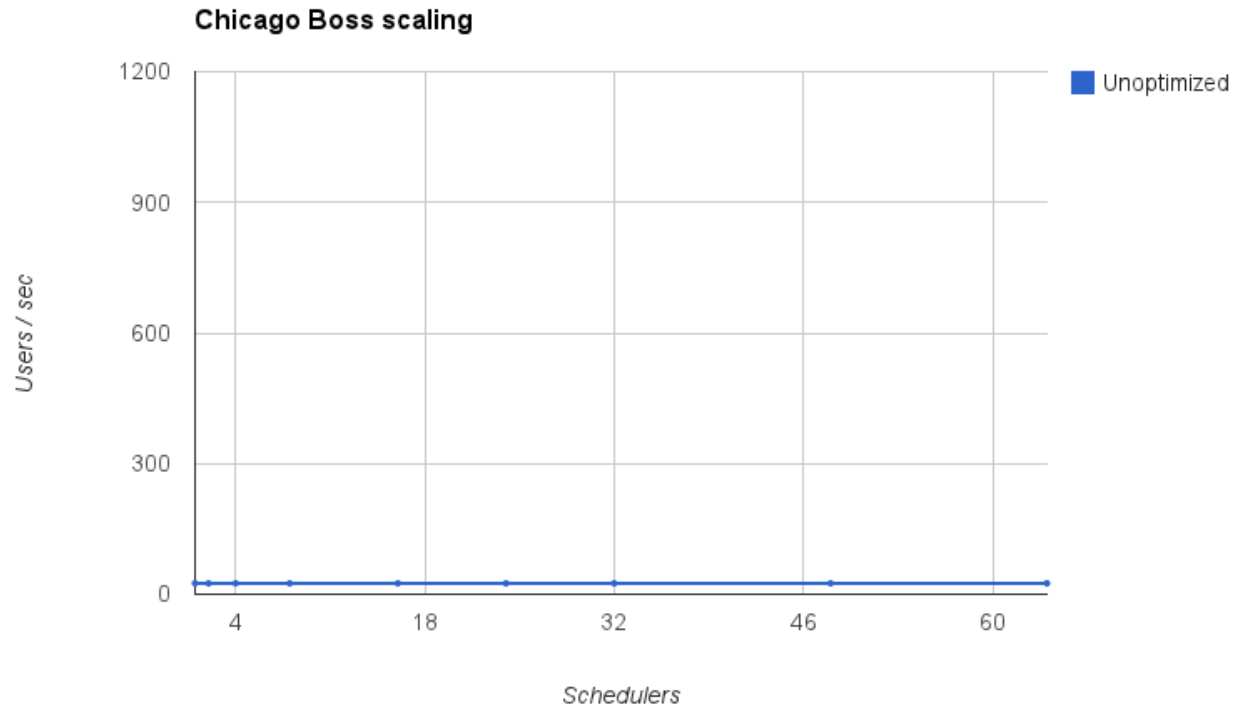
Check out Evan Miller's Chicago Boss talk on Friday 4pm

The manycore era is here.



Real world Chicago Boss application (concurix.com) December 2012

The manycore era is here.



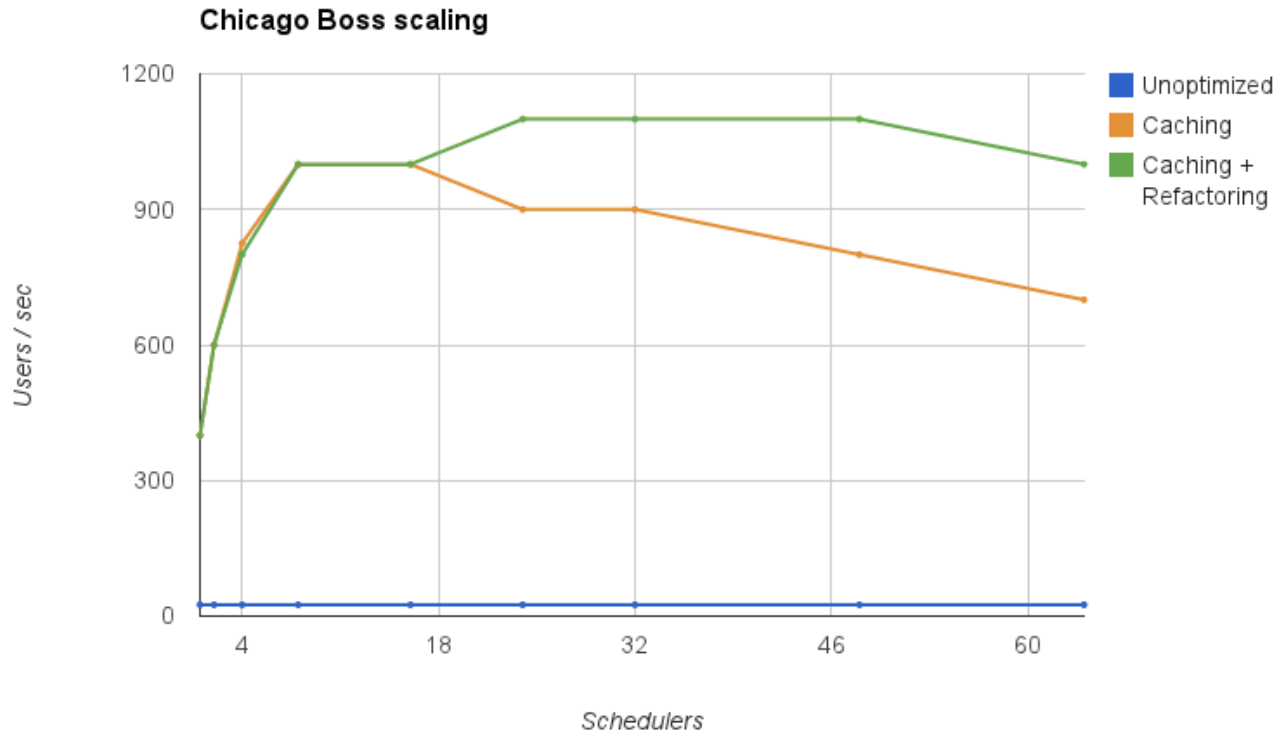


G. TARPLEY

The manycore era is here.



Where we are now... 45x!!!



The manycore era is here.



How we did it...

The manycore era is here.



Back to Amdahl's Law—it's all about the locks

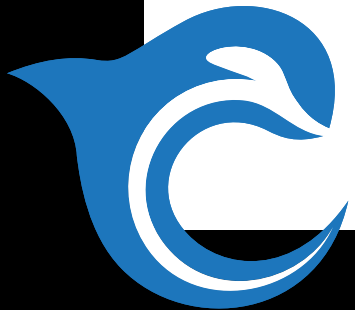


The manycore era is here.

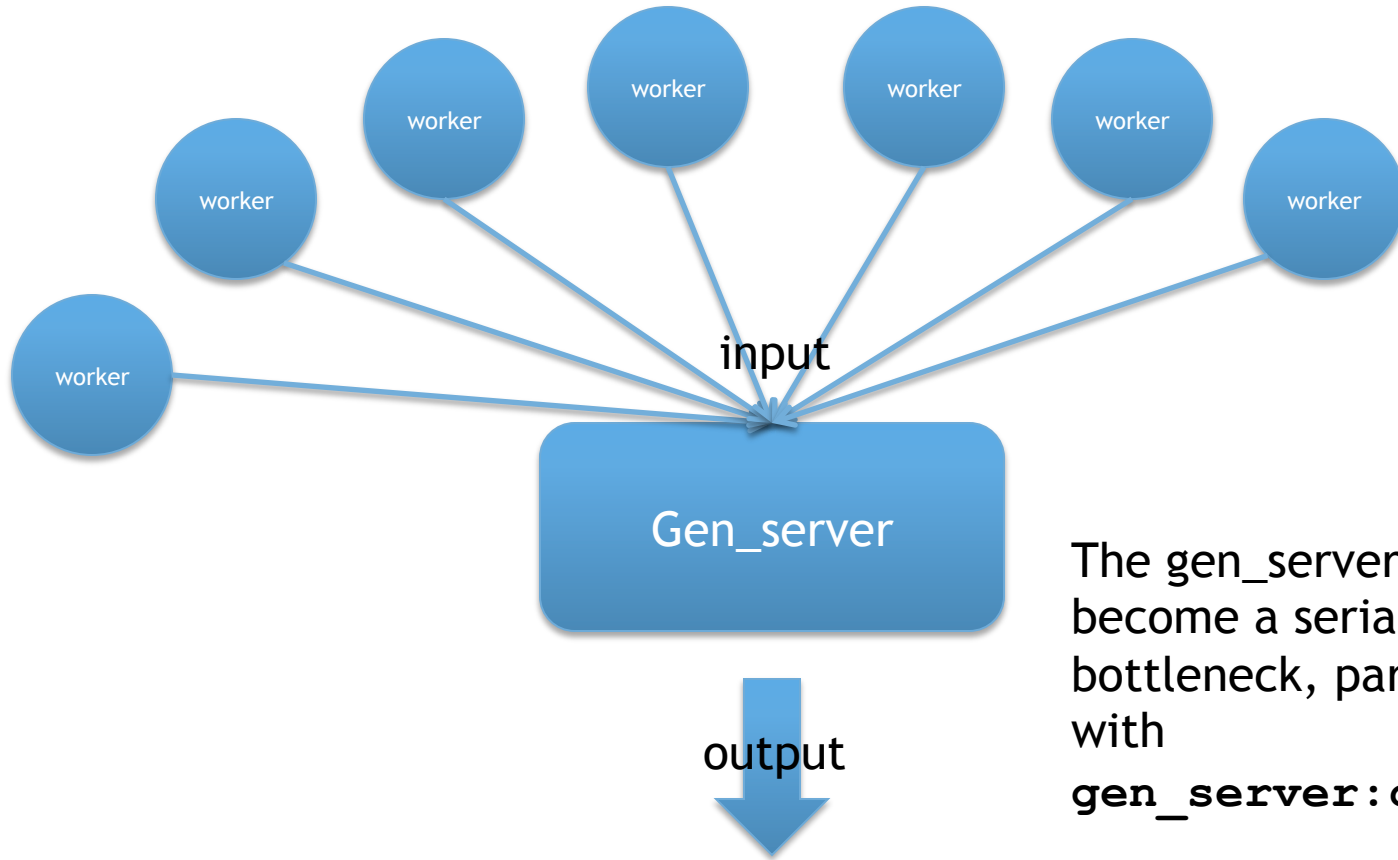


gen_server –the hidden lock in Erlang

```
1 %% -*- erlang-indent-level: 4;indent-tabs-mode: nil -*-
2 %% ex: ts=4 sw=4 et
3 %% @author {{author}} <{{author_email}}>
4 %% @copyright {{copyright_year}} {{copyright_holder}}
5
6 -module({{name}}).
7
8 -behaviour(gen_server).
9
10 -export([start_link/0]).
11
12 %% gen_server callbacks
13 -export([init/1,
14         handle_call/3,
15         handle_cast/2,
16         handle_info/2,
17         terminate/2,
18         code_change/3]).
19
20 -record(state, {}).
21
22 start_link() ->
23     gen_server:start_link({local, ?MODULE}, ?MODULE, [], []).
24
25 init([]) ->
26     {ok, #state{}}.
27
28 handle_call(_Request, _From, State) ->
29     {reply, ignored, State}.
30
31 handle_cast(_Msg, State) ->
32     {noreply, State}.
33
34 handle_info(_Info, State) ->
35     {noreply, State}.
36
37 terminate(_Reason, _State) ->
38     ok.
39
40 code_change(_OldVsn, State, _Extra) ->
41     {ok, State}.
42
43 %% Internal functions
```

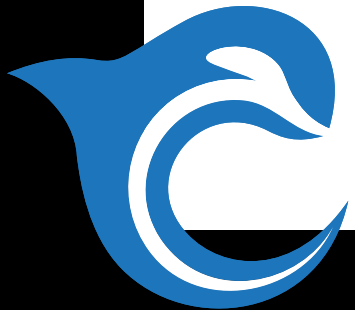


Worker processes can bottleneck on a gen_server



The gen_server can become a serialization bottleneck, particularly with `gen_server:call(...)`

The manycore era is here.



These can be hard to spot without studying the code

```
17 reload_routes() ->
18     gen_server:call(boss_web, reload_routes).
19
20 reload_translation(Locale) ->
21     gen_server:call(boss_web, {reload_translation, Locale}).
22
23 reload_all_translations() ->
24     gen_server:call(boss_web, reload_all_translations).
25
26 reload_init_scripts() ->
27     gen_server:call(boss_web, reload_init_scripts).
28
29 get_all_routes() ->
30     gen_server:call(boss_web, get_all_routes).
31
32 get_all_models() ->
33     gen_server:call(boss_web, get_all_models).
34
35 get_all_applications() ->
36     gen_server:call(boss_web, get_all_applications).
37
38 base_url(App) ->
39     gen_server:call(boss_web, {base_url, App}).
40
41 domains(App) ->
42     gen_server:call(boss_web, {domains, App}).
43
44 static_prefix(App) ->
45     gen_server:call(boss_web, {static_prefix, App}).
46
47 translator_pid(AppName) ->
48     gen_server:call(boss_web, {translator_pid, AppName}).
49
50 router_pid(AppName) ->
51     gen_server:call(boss_web, {router_pid, AppName}).
52
53 application_info(App) ->
54     gen_server:call(boss_web, {application_info, App}).
```



The Bet



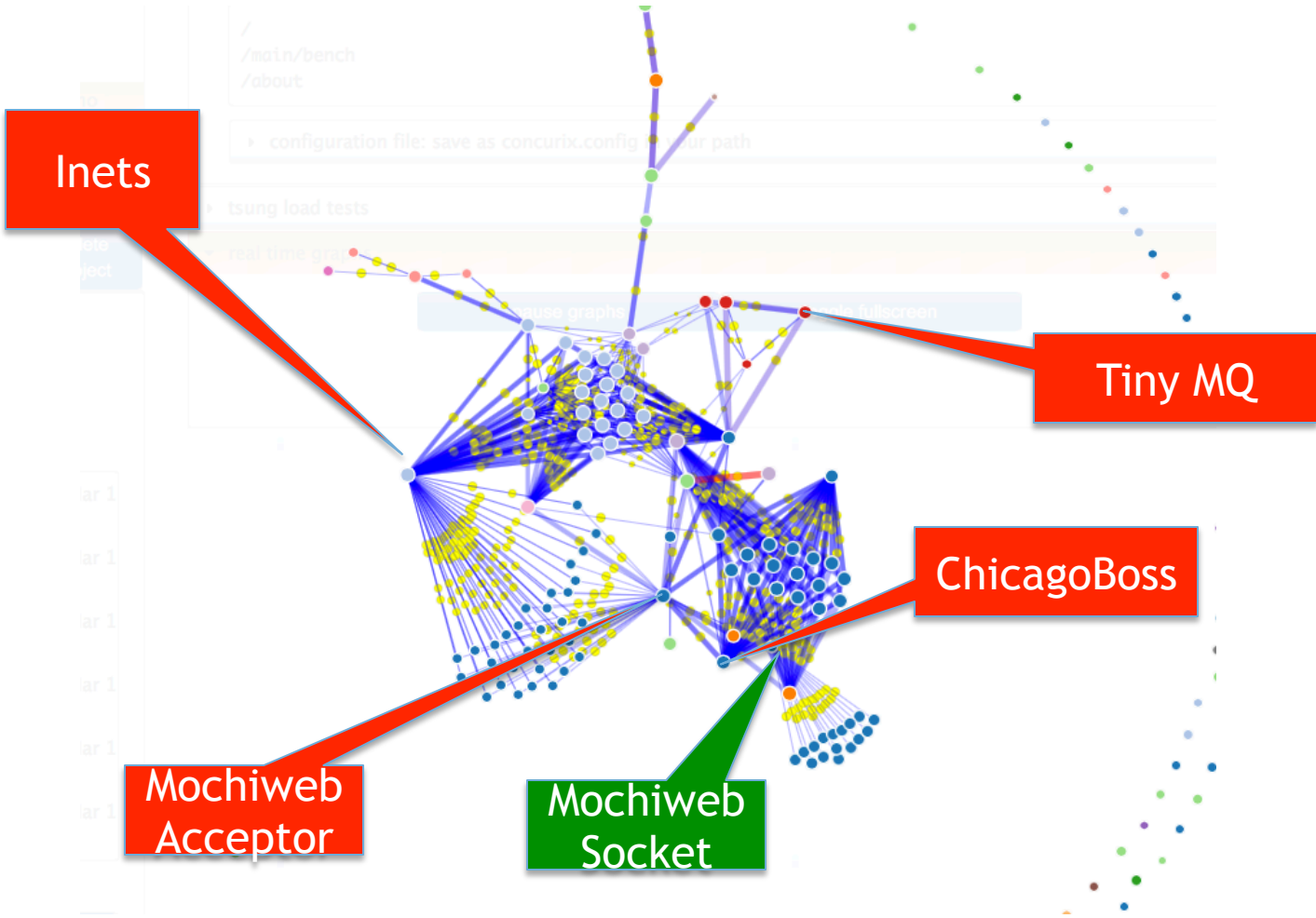
The manycore era is here.





DEMO

Message passing between processes



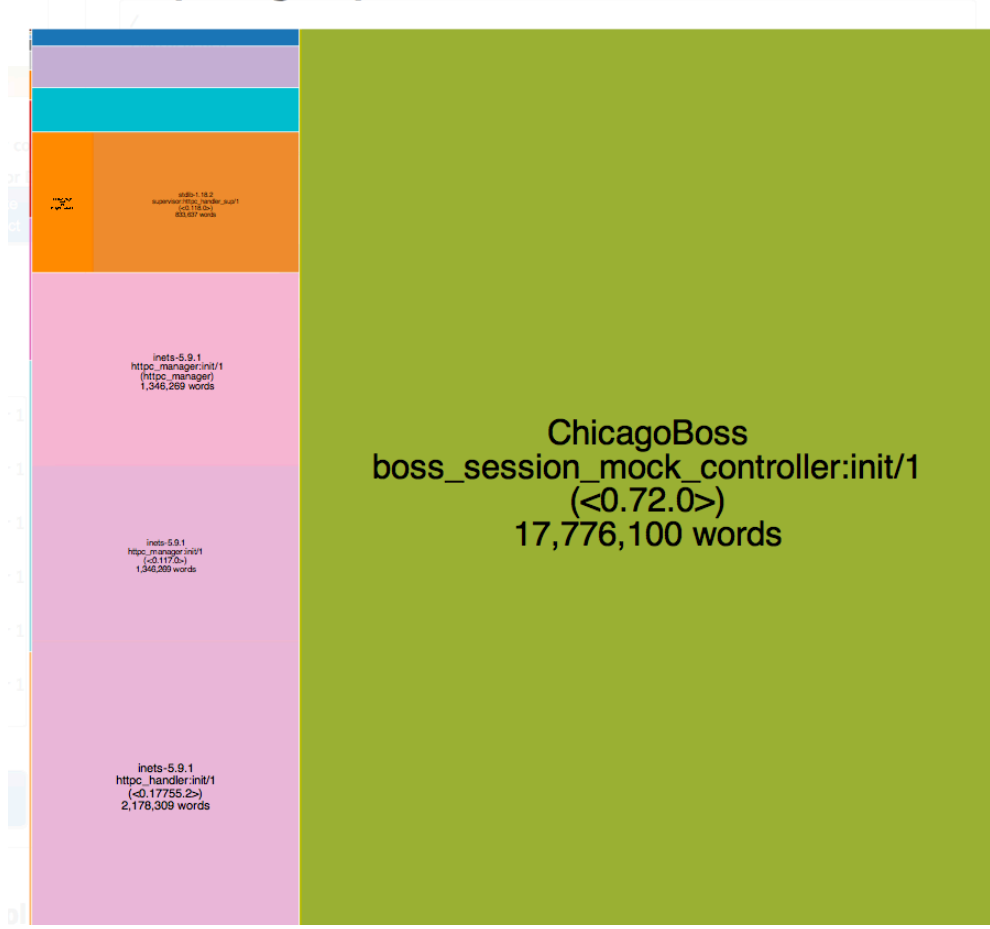
The manycore era is here.



Excessive Memory Usage

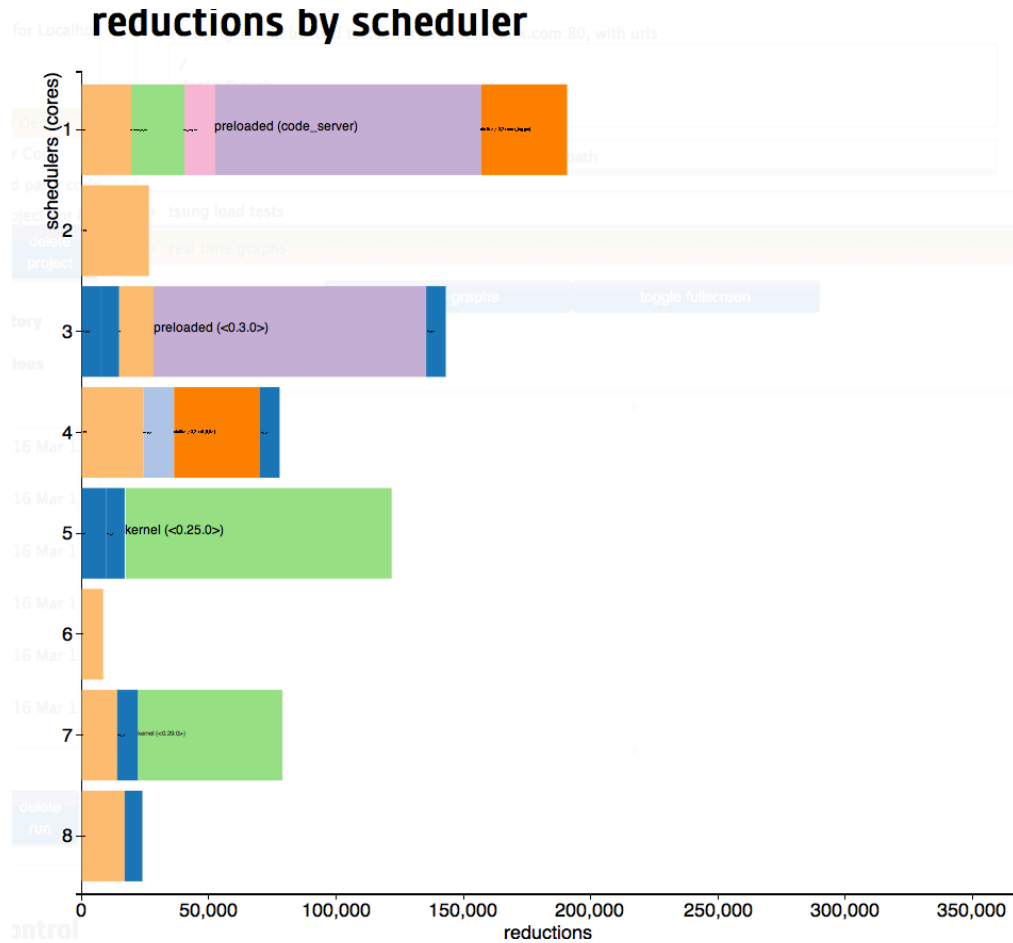
The manycore era is here.

memory usage by service



Uneven CPU Core utilization

The manycore era is here.



Use the `+sbt nnts` flag to lock threads to schedulers!



Try it yourself!

1. Add `concurix_runtime` to your `rebar.config` file:

```
{concurix_runtime, ".*",  
  {git, "https://github.com/Concurix/cx_runtime.git"}}
```

2. Start the `concurix_runtime` system:

```
concurix_runtime:start()
```

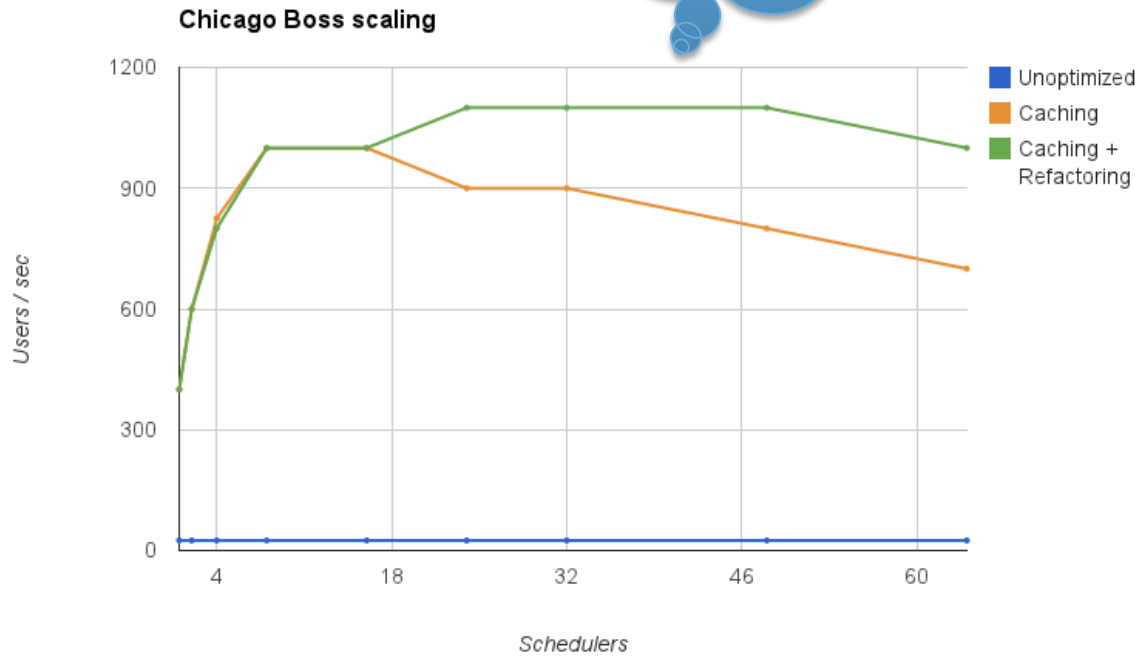
3. Navigate to <http://concurix.com/main/bench>



Promising future

The manycore era is here.

Tons of headroom still!



The science behind Concurix!

Dr. Ying Li, Concurix Chief Scientist: 4:05 PM Crystal Lounge

$$Cluster^{(t)}_i = \{ p : \|p - center^{(t)}_i\| \leq \|p - center^{(t)}_j\| \forall j \leq k \}$$

$$C_{EC}(v) = \lambda \sum_{\{u,v\} \in E} C_{EC}(u)$$

$$\theta = \arccos\left(\frac{a \cdot b}{\sqrt{\sum_{k=1}^n a^2_k} \sqrt{\sum_{k=1}^n b^2_k}}\right)$$

