bet365

bet365

Lessons learnt re-writing a PubSub system

Chandru Mullaparthi - Principal Software Architect at bet365

About

- **Founded in 2000**
- **Located in Stoke-on-Trent**
- **The largest online sports betting company**
- **Over 19 million customers**
- **One of the largest private companies in the UK**
- **Employs more than 2,000 people**
- **2013-2014: Over £26 billion was staked**
  - **Last year is likely to be around 25% up**
    - **Business growing very rapidly!**
- **Very technology focused company**

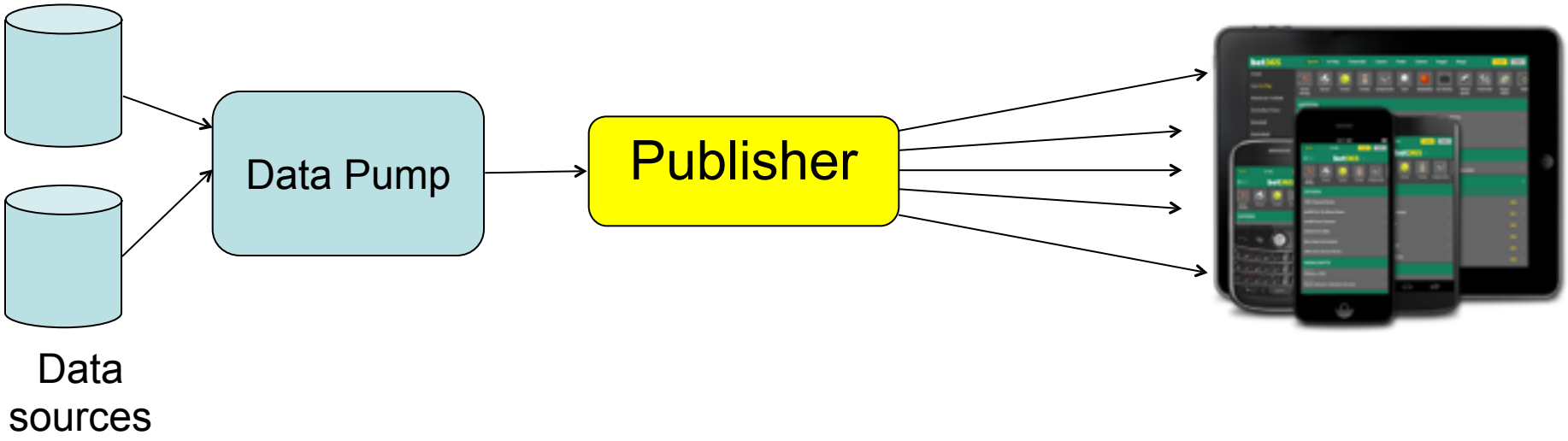
History of Erlang use

bet365

- Started experimenting with Erlang/OTP circa 2011
- 3 major systems written in Erlang
- 4th one in progress
- Code base of varied quality

Publisher

bet365

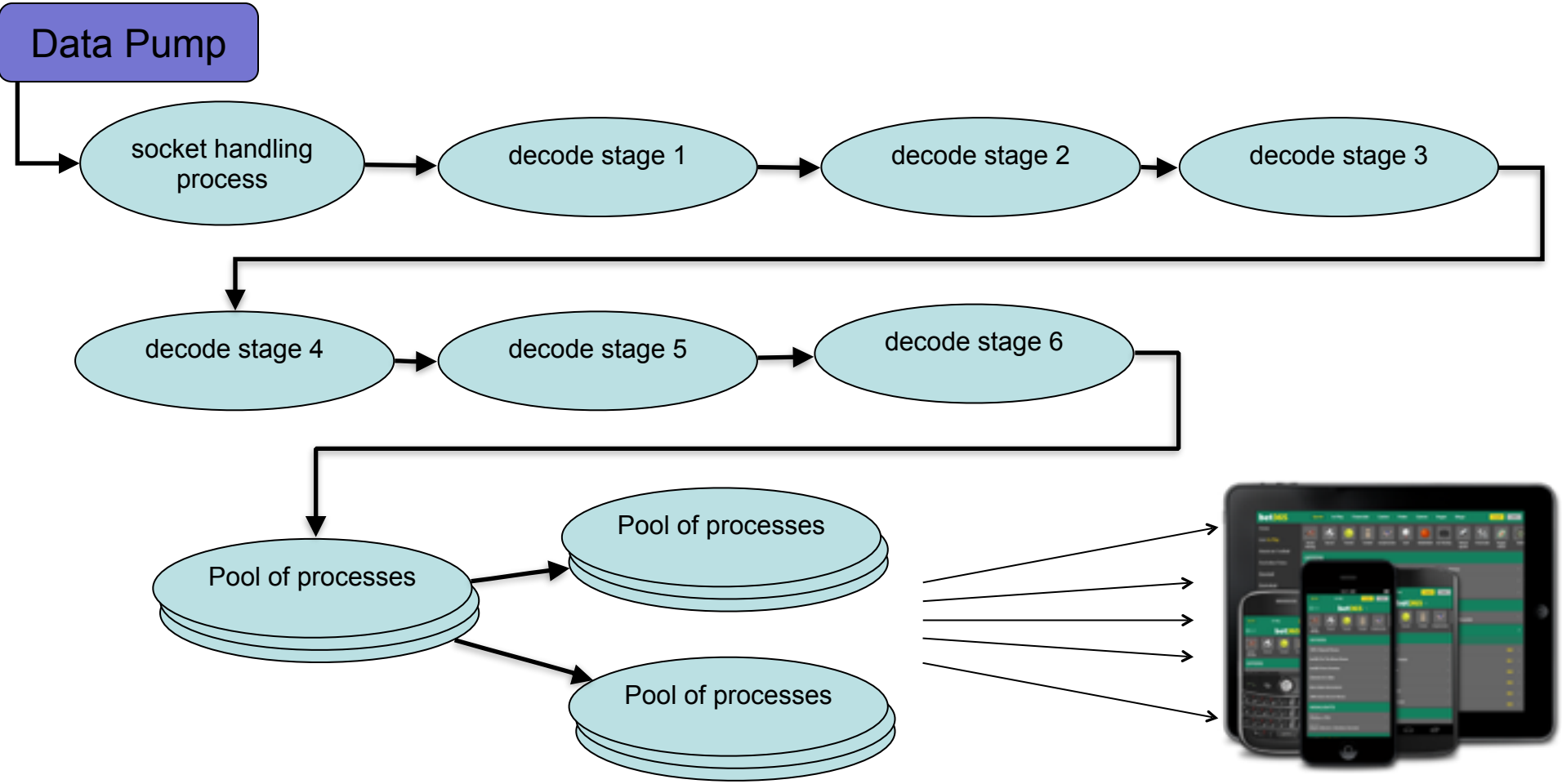


Why rewrite?

bet365

Complicated process structure

Message queue hotspots



New architecture

bet365

- Exploits tree structure of data shipped to clients
- Total number of nodes in the tree ~2.2million
- Approx 130K top level nodes just below root

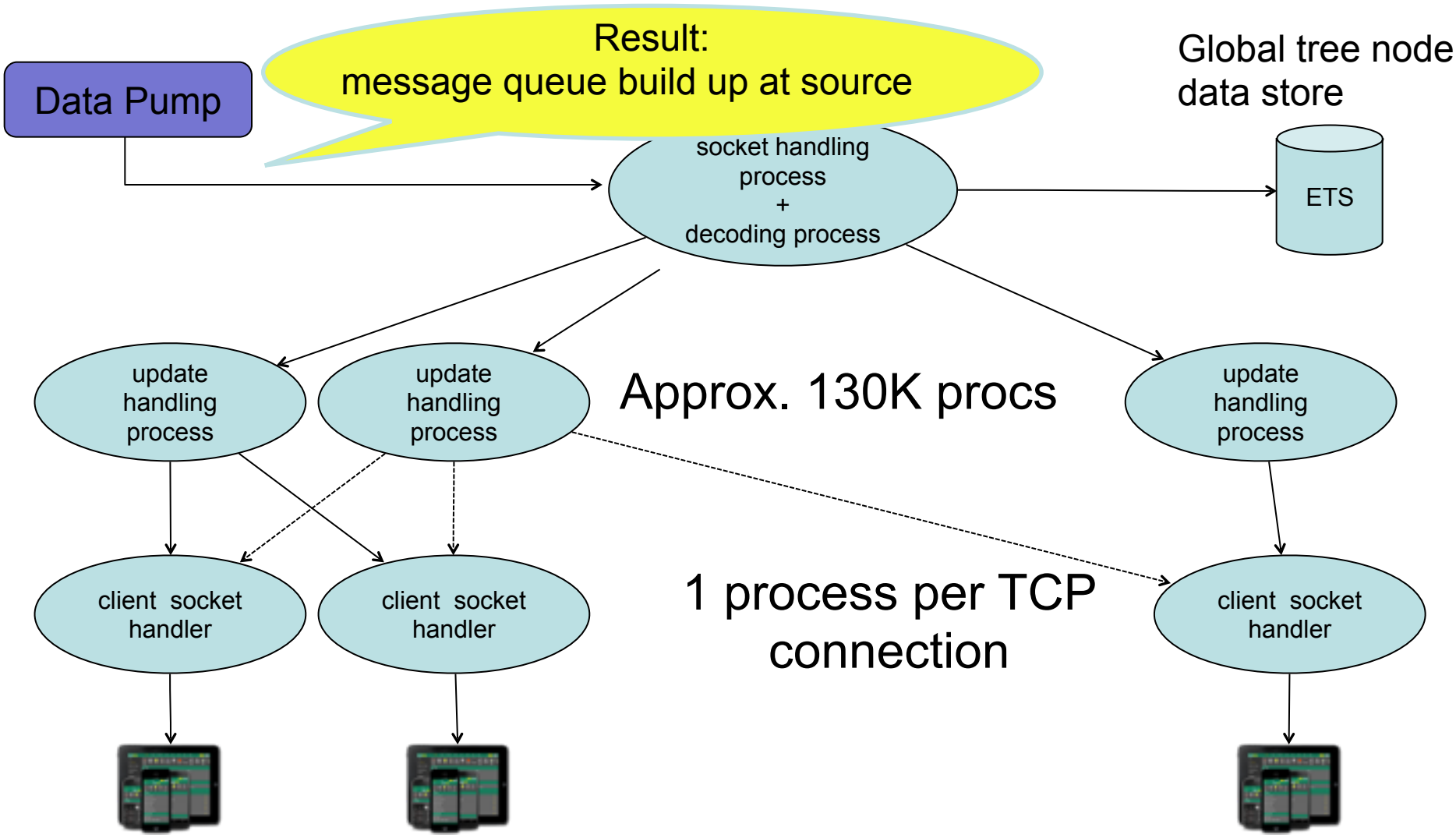
1st attempt

bet365

- One process to take a message off the socket
- Decode
- Update global-state ETS table
- Send message to one of 130K top level procs

New architecture

bet365



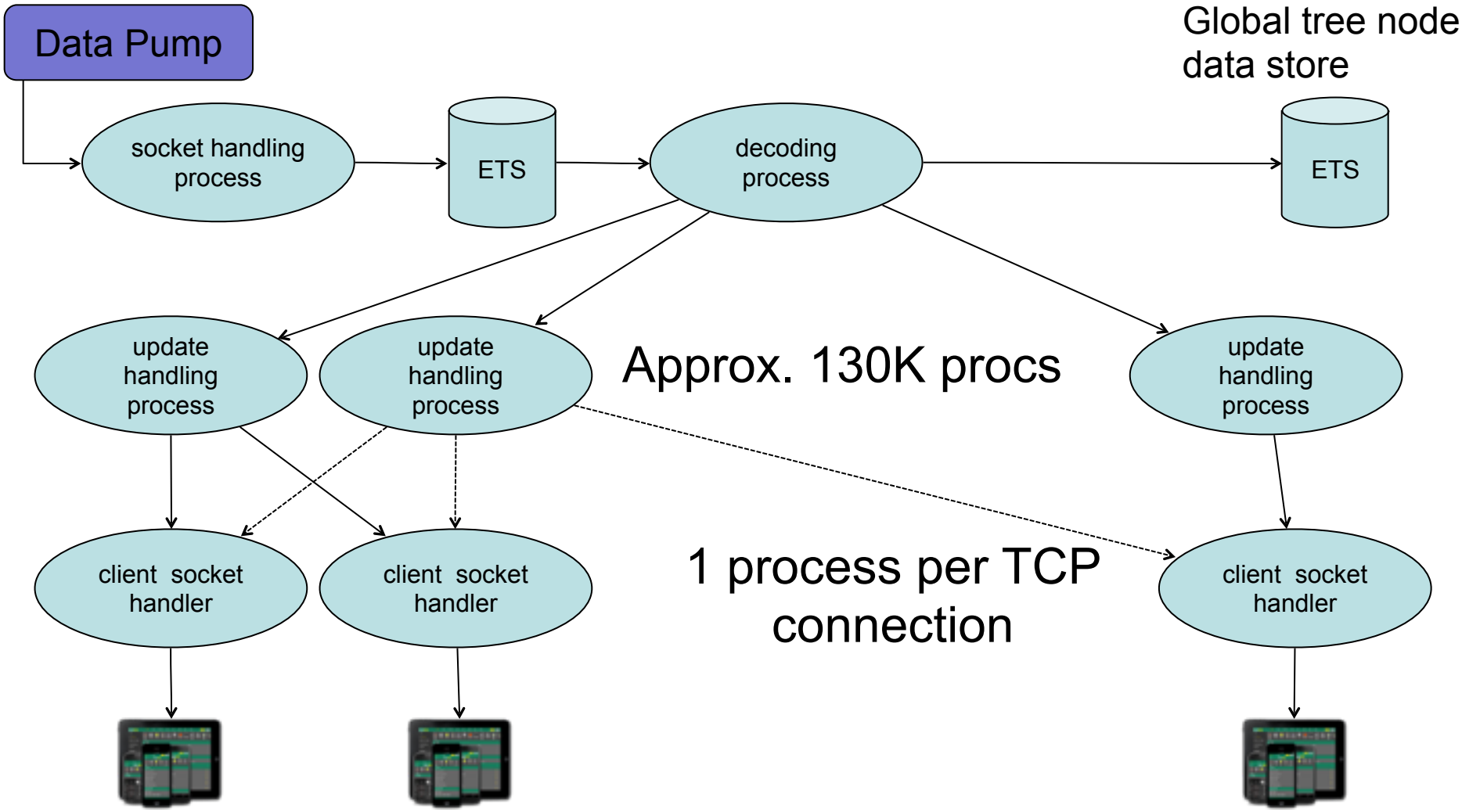
2nd attempt

bet365

- One process to take a message off the socket
- Decoding 1st stage
- Insert into ETS table
- Another process pops message off ETS table
- Update ETS table
- Send message to one of 130K top level procs

New architecture

bet365



Results

bet365

It worked well with 1.8 million nodes in the tree

BUT

throughput fell and latency increased for 2.2m nodes in the tree

Garbage collection

bet365

- GC is per process
- Regular GC of a process is “generational”. Data that survives at least one GC will be put on the old heap
- When there is no space on the old heap, a “fullsweep” is performed

GC continued

bet365

- The time taken to GC eats into the 2000 reductions available for a process
- So a “busy process” (with lots of incoming messages) with lots of garbage produced will quickly spiral out of control

Reduction count penalty

bet365

- Sending message to a process with a “large” message queue
- Long GC
- Proportional to size of message sent to another node
- Proportional to size of binaries created
- …

Nice to have
Better documentation about reduction count consumption and “penalties”

Process priority flag

bet365

- process_flag(priority, high) does NOT mean more reductions are available

- It just means it gets scheduled before others, if it has messages in its queue

Nice to have
Support for processes which have a dedicated execution thread may be useful?

Diagnosis

bet365

```
[root@publisher ~]#
for x in $(seq 10000); \
   do ps -eo ppid,pid,user,stat,pcpu,comm,wchan:32 | \
      grep D | \
      egrep -v "\-|WCHAN"; \
   done;
```

-e : Select all processes
-o : user defined format

ppid : parent process id
pid : process id
stat : process state
pcpu : CPU utilisation
comm : command name
wchan : kernel function name where process is sleeping

```
2   263 root     DN    0.0 khugepaged      call_rwsem_down_write_failed
2   263 root     DN    0.0 khugepaged      call_rwsem_down_write_failed
2   263 root     DN    0.0 khugepaged      call_rwsem_down_write_failed
2   263 root     DN    0.0 khugepaged      call_rwsem_down_write_failed
2   263 root     DN    0.0 khugepaged      call_rwsem_down_write_failed
2   263 root     DN    0.0 khugepaged      call_rwsem_down_write_failed
```

THP

bet365

- Transparent Huge Pages are a feature in 64-bit RHEL
- Enabled by default for all applications
- Abstraction over “huge page” support in Linux
- Most database vendors recommend turning this off

What are huge pages?

bet365

- Typical size of page in memory is 4KB
- 20GB of memory equates to 5,242,880 pages and that many entries in the page table.
- RHEL kernel allows page sizes of 2MB and 1GB on x86_64

How does THP work?

bet365

- Using the huge page feature requires explicit code change to the application (in this case ‘beam’)
- THP aims to make it “just happen”
- It also does some memory defragmentation which seems to be the root cause of the issue

Disabling THP

bet365

```
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
echo no > /sys/kernel/mm/redhat_transparent_hugepage/khugepaged/defrag
```

Permanently disable THP

bet365

```
#!/bin/sh

if [[ $EUID -ne 0 ]]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi

for KERNEL in /boot/vmlinuz-*; do
    grubby --update-kernel="$KERNEL" --args='transparent_hugepage=never'
done
```

source: http://unix.stackexchange.com/questions/99154/disable-transparent-hugepages

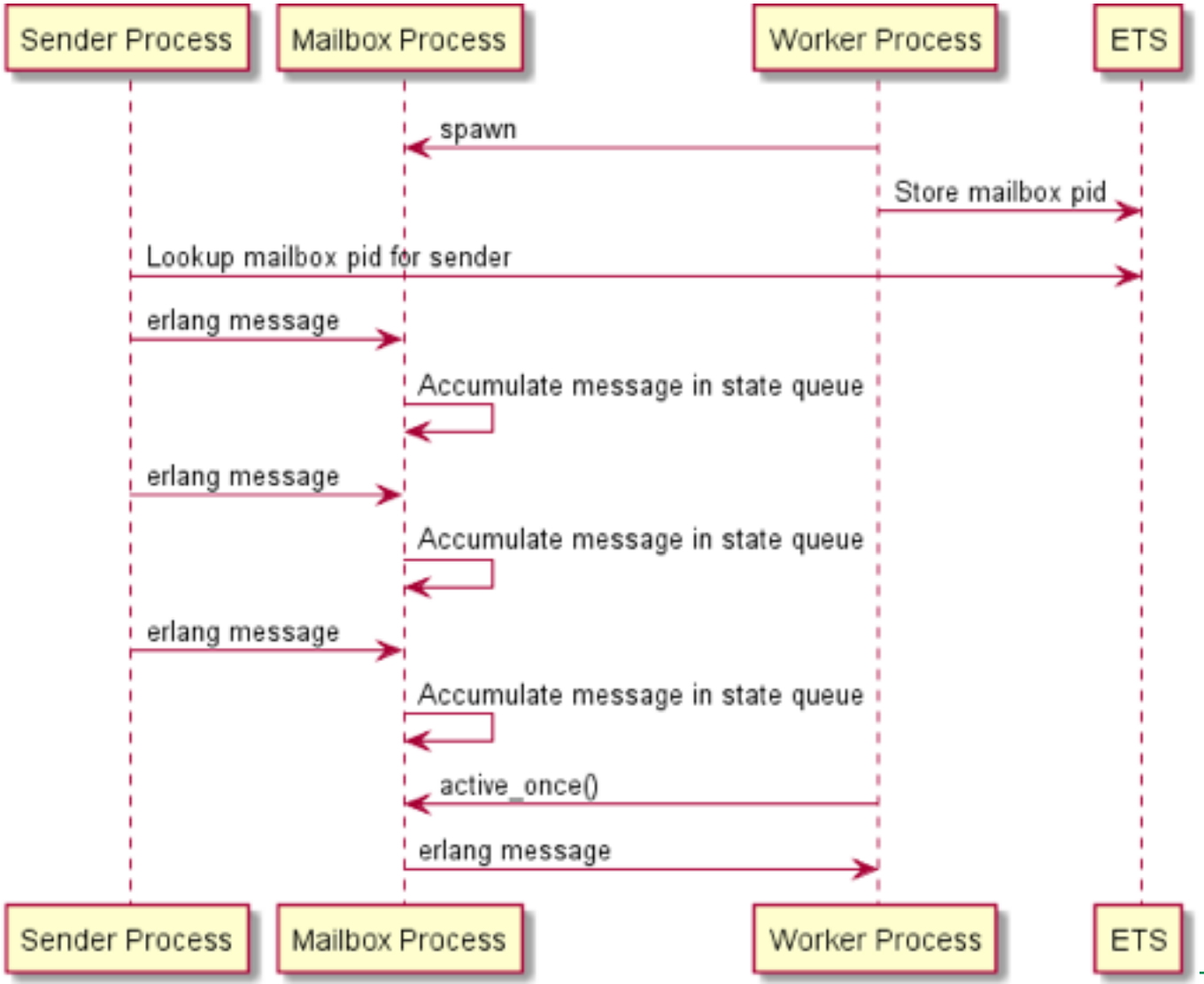
Message passing

bet365

- Double edged sword
  - Easy to write reasonably complex code, quickly
  - Enough rope to hang yourself easily
- Developed a process message queue with {active, once} semantics
- No improvement in performance, but no noticeable degradation either!

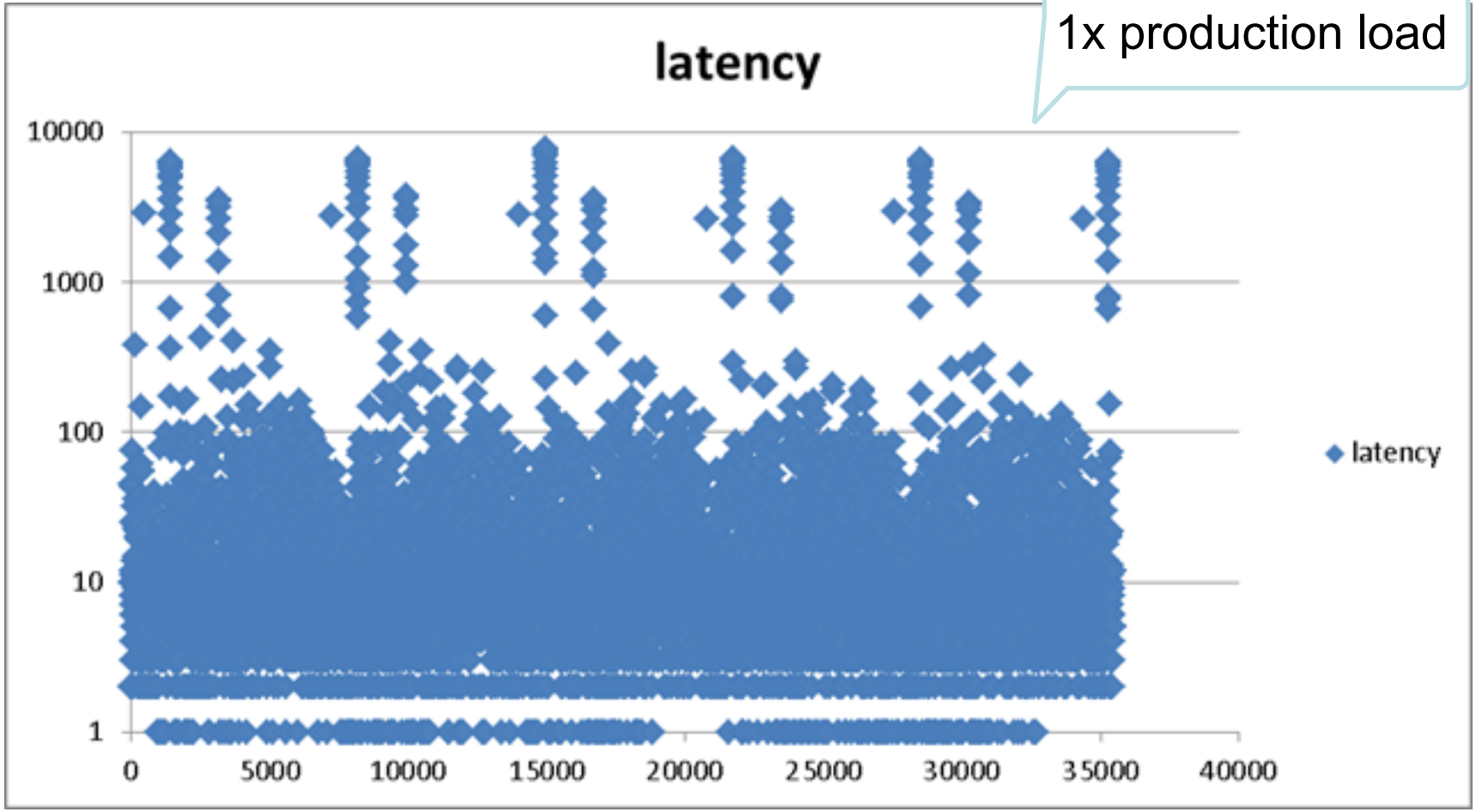
Message queue

bet365



Results

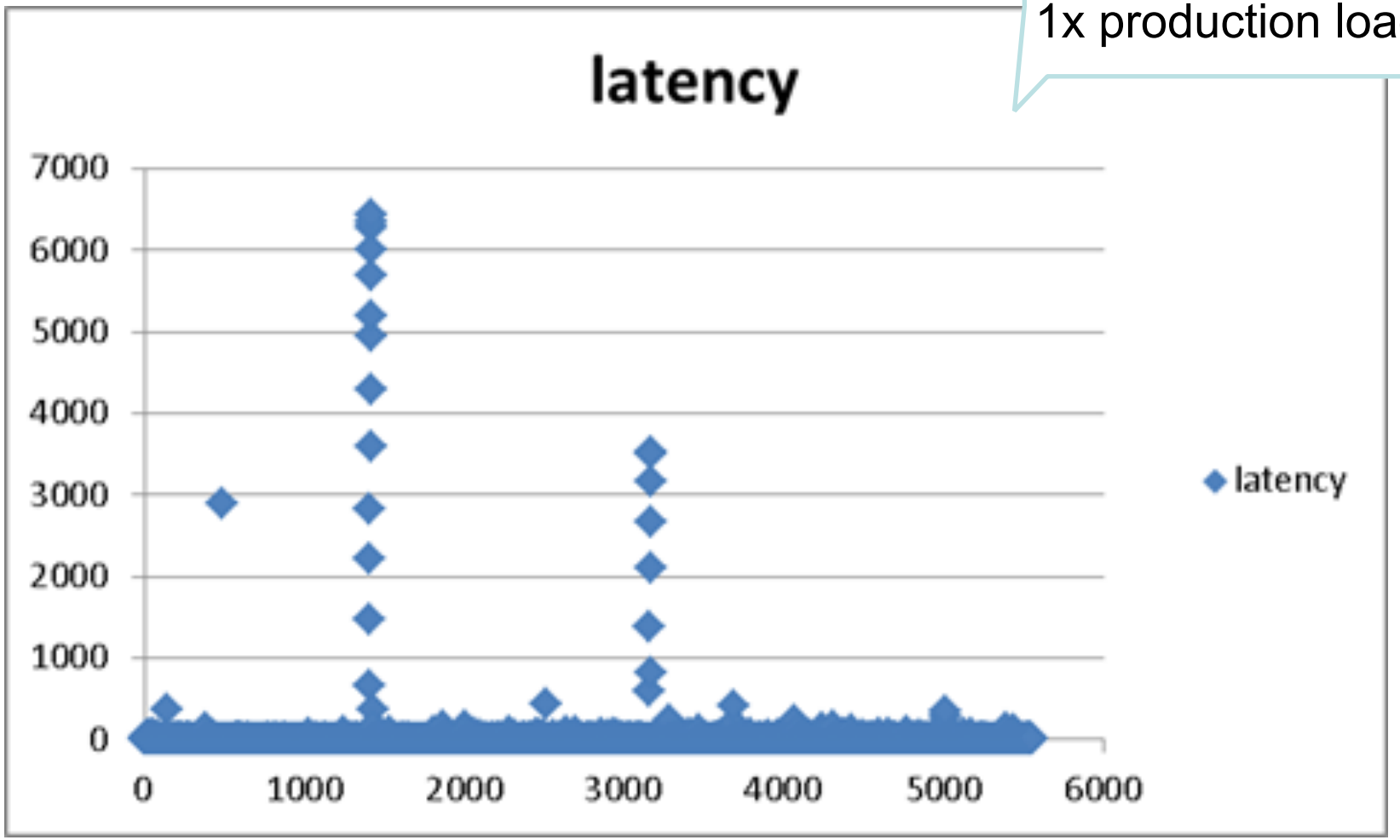
bet365



Results

bet365

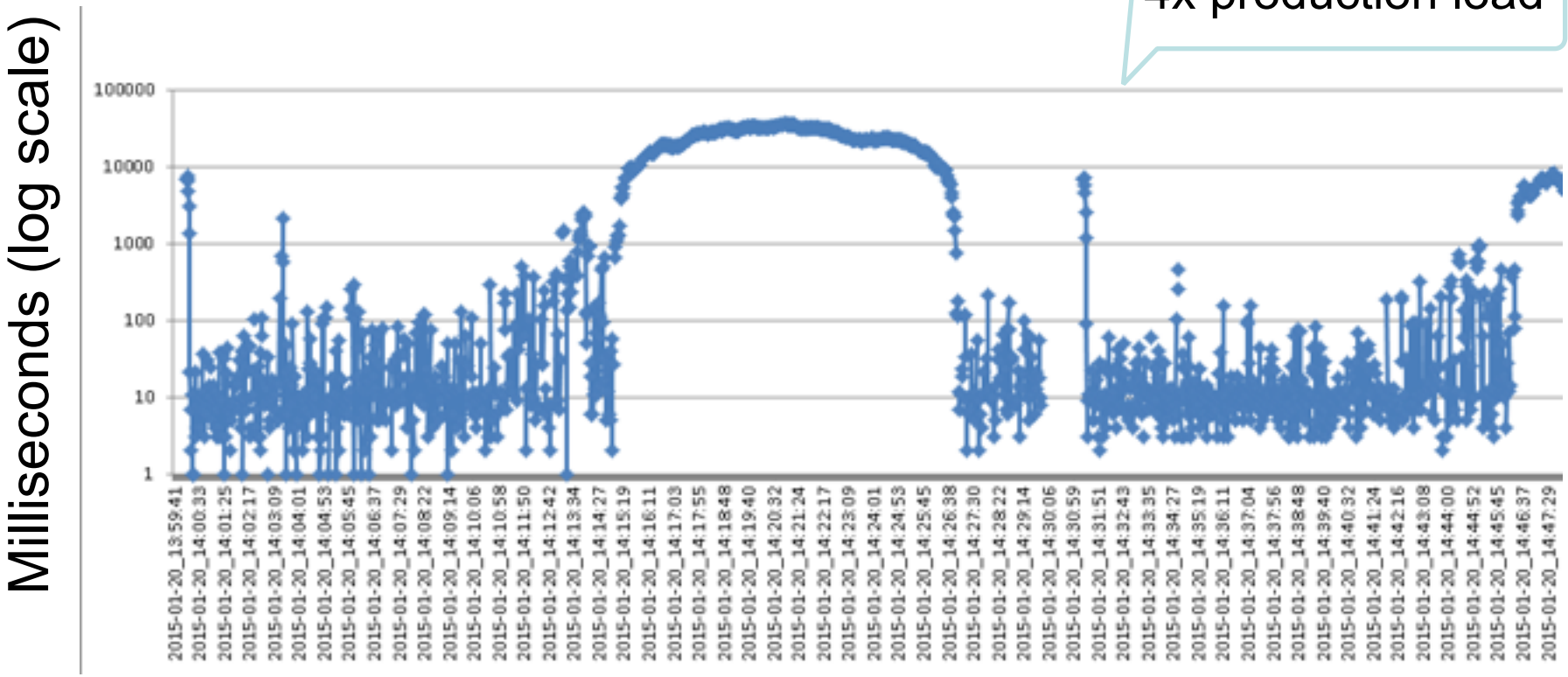
100K clients
1x production load



Results

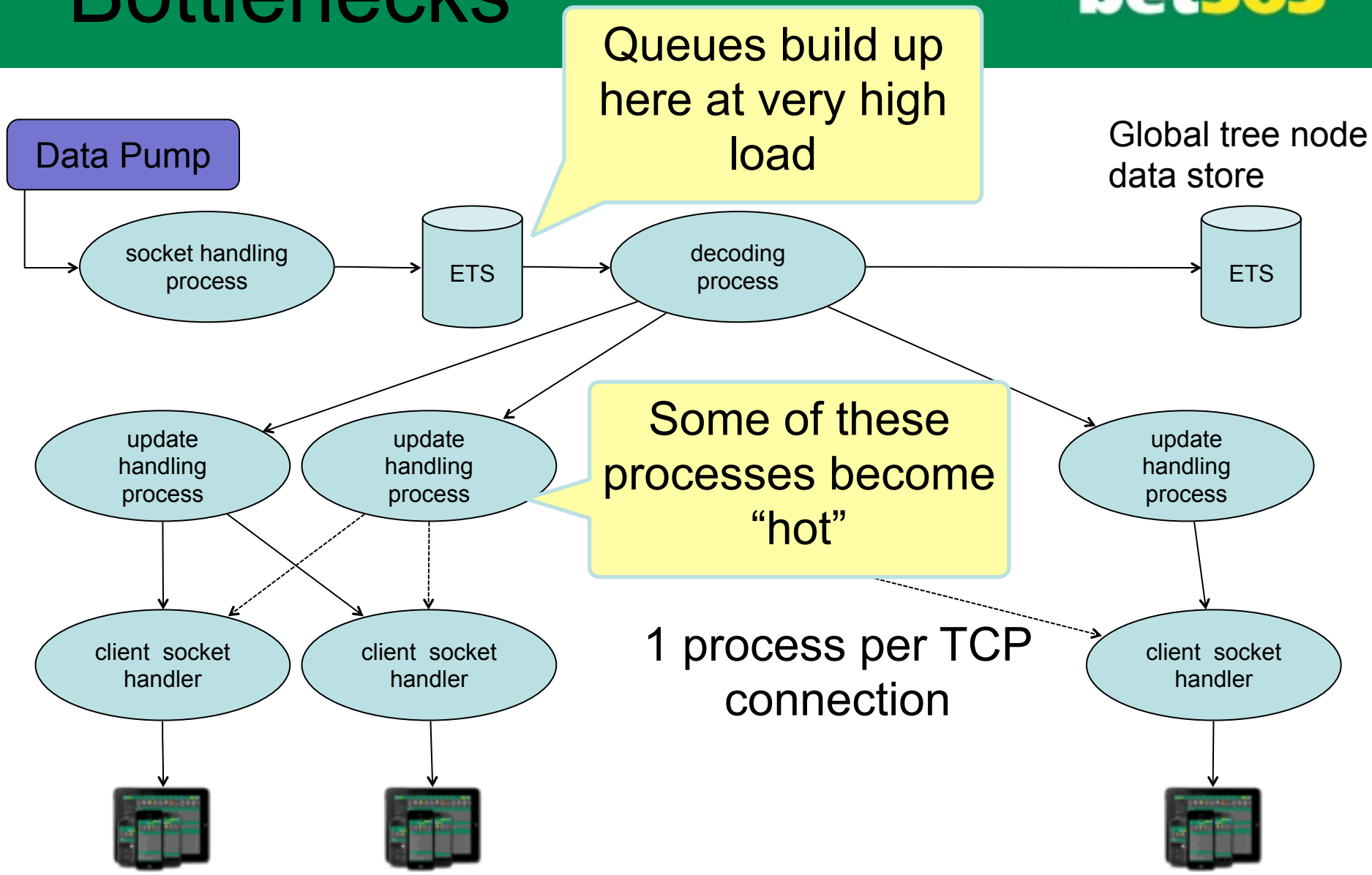
bet365

100K clients
4x production load



Bottlenecks

bet365



Work in progress

Distributed Data Grid

bet365

- An mnesia like database
- Without the storage backend issues
- And without the “partitioned network” problems with mnesia
- Something similar to Infinispan

Bounded message q

bet365

- Long term itch

- Seems fairly straightforward to achieve
  - Additional option max_msg_q_length in process_flag/2 BIF
  - Modify process_info/2 BIF
  - Kill process if message queue length exceeds configured limit

Deep packet inspection

bet365

- Analyse traffic patterns in real-time
- Proof of concept being worked on using Erlang
- Speed will be an issue here so an interesting challenge

COME WORK WITH US!