

# **Nginx Internals**

Joshua Zhu

09/19/2009

# Agenda

- **Source code layout**
- Key concepts and infrastructure
- The event-driven architecture
- HTTP request handling
- Mail proxying process
- Nginx module development
- Misc. topics

# Source Code Layout

- Files

- `$ find . -name "*\.[hc]" -print | wc -l`  
234
- `$ ls src`  
core event http mail misc os

- Lines of code

- `$ find . -name "*\.[hc]" -print | xargs wc -l | tail -n1`  
110953 total

# Code Organization

- `core/`
  - The backbone and infrastructure
- `event/`
  - The event-driven engine and modules
- `http/`
  - The HTTP server and modules
- `mail/`
  - The Mail proxy server and modules
- `misc/`
  - C++ compatibility test and the Google perftools module
- `os/`
  - OS dependent implementation files

# Nginx Architecture

- Non-blocking
- Event driven
- Single threaded<sup>[\*]</sup>
- One master process and several worker processes
- Resource efficient
- Highly modular

# The Big Picture



# Agenda

- Source code layout
- **Key concepts and infrastructure**
- The event-driven architecture
- HTTP request handling
- Mail proxying process
- Nginx module development
- Misc. topics

# Memory Pool

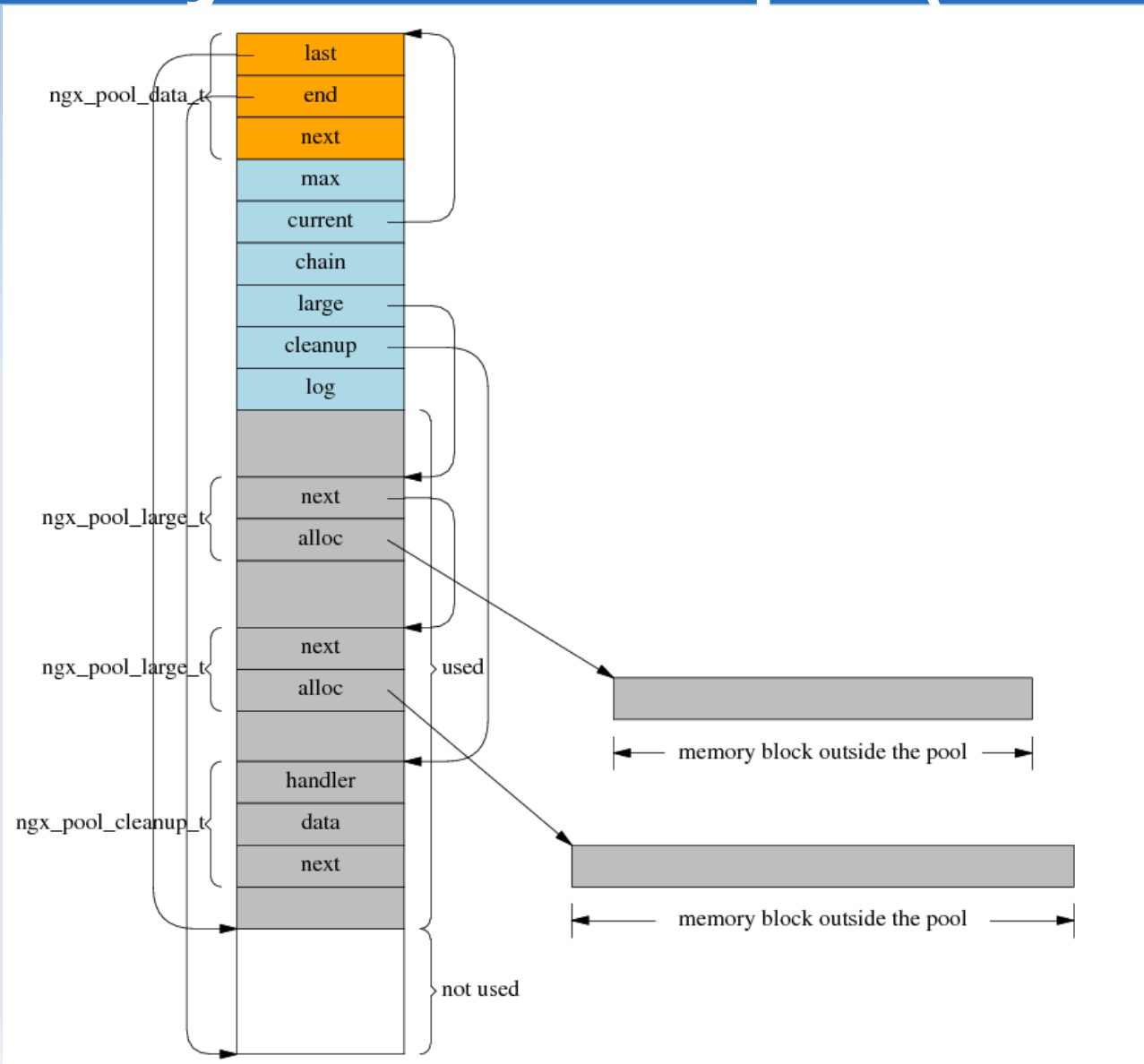
- Avoid memory fragmentation
- Avoid memory leak
- Allocation and deallocation can be very fast
- Lifetime and pool size
  - Cycle
  - Connection
  - Request



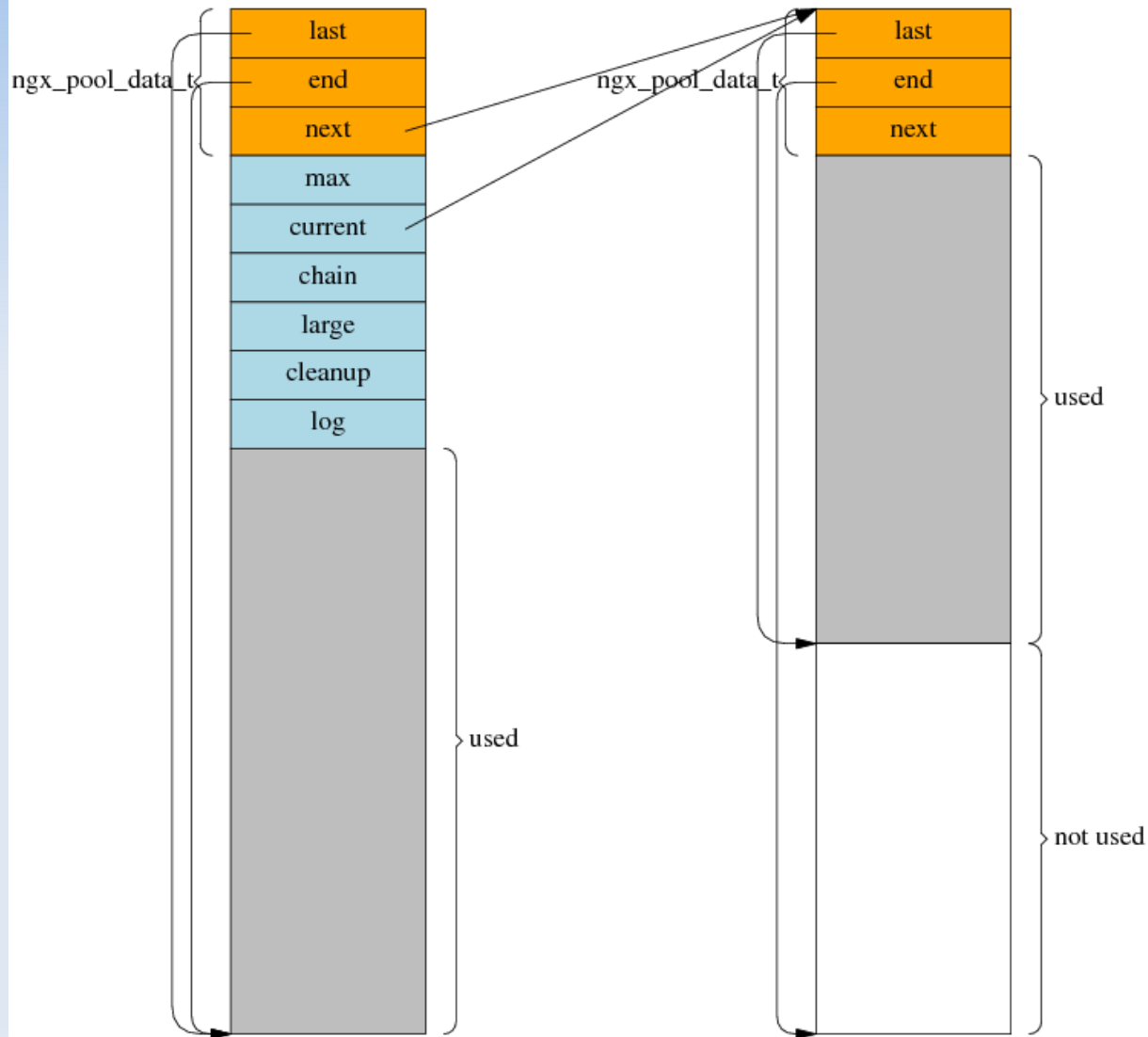
# Memory Pool (cont'd)

- ngx\_pool\_t
  - Small blocks
  - Large blocks
  - Free chain list
  - Cleanup handler list
- API
  - ngx\_palloc
    - memory aligned
  - ngx\_pnalloc
  - ngx\_pccalloc

# Memory Pool Example (1 Chunk)

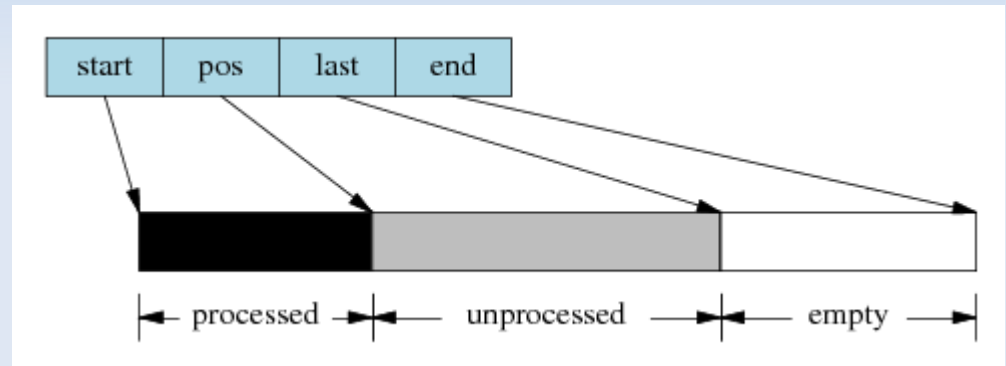


# Memory Pool Example (2 Chunks)



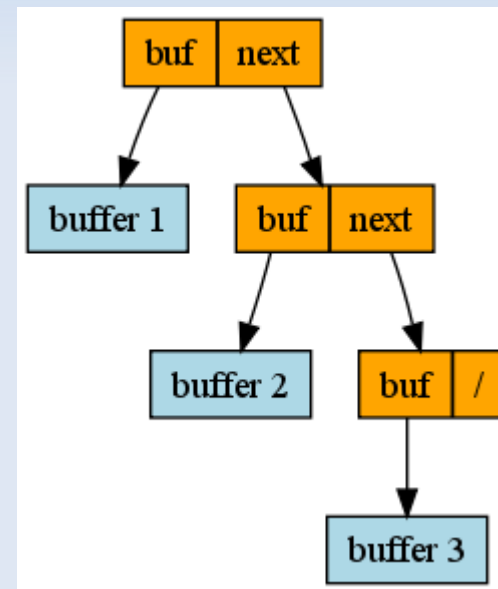
# Buffer Management

- Buffer
  - Pointers
    - memory
      - start/pos/last/end
    - file
      - file\_pos/file\_last/file
  - Flags
    - last\_buf
    - last\_in\_chain
    - flush
    - in\_file
    - memory
    - ...



# Buffer Management (cont'd)

- Buffer chain
  - Singly-linked list of buffers
- Output chain
  - Context
    - in/free/busy chains
  - Output filter
- Chain writer
  - Writer context



# String Utilities

- ngx\_str\_t
  - data
  - len
  - sizeof() - 1
- Memory related
- String formatting
- String comparison
- String search
- Base64 encoding/decoding
- URI escaping/unescaping
- UTF-8 decoding
- String-to-number conversion

# Data Structures

- Abstract data types
  - Array
  - List
  - Queue
  - Hash table
  - Red black tree
  - Radix tree
- Characteristic
  - Set object values after added
    - keep interfaces clean
  - Chunked memory (part)
    - efficient

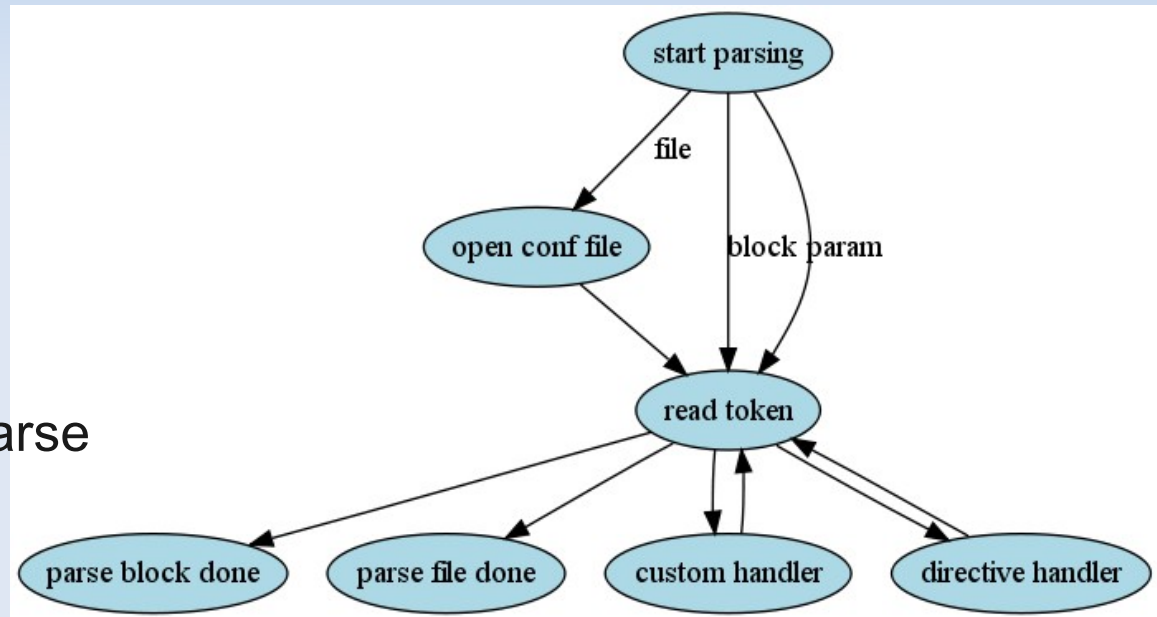
# Logging

- Error log
  - Level
  - Debug
- Access log
  - Multiple logs
  - Log format
    - variables
  - Per location
- Rotation



# Configuration File

- Directive
  - name
  - type
  - set
  - conf
  - offset
  - post
- Parsing
  - ngx\_conf\_parse
- Values
  - init
  - merge



# Configuration File (cont'd)

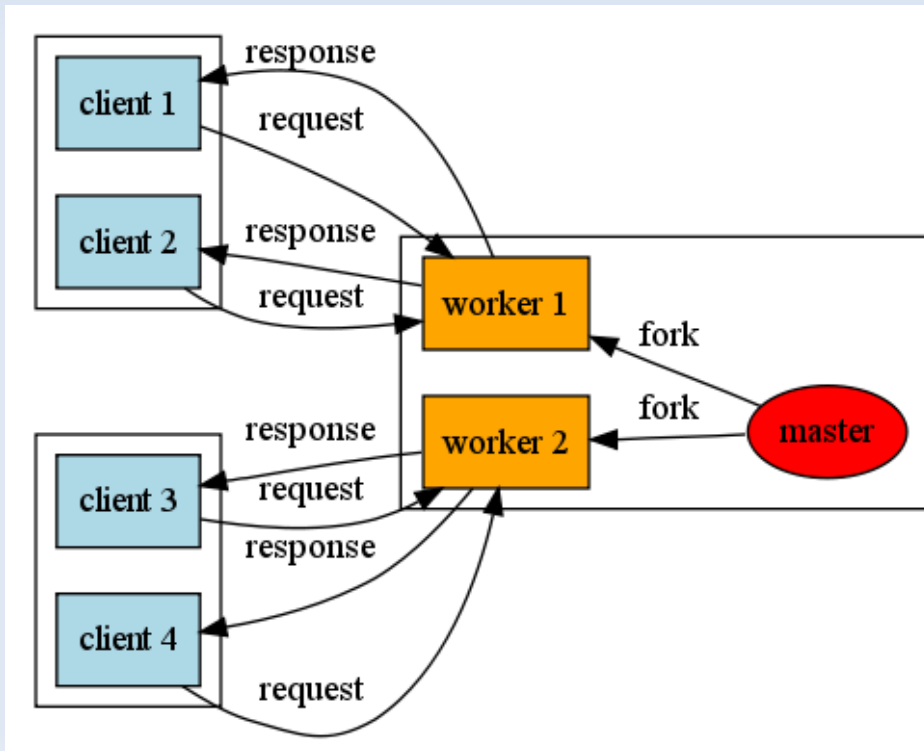
- Block
  - events
  - http
  - server
  - upstream
  - location
  - if
- Variables
  - Buildins
  - Other types
    - http\_
    - sent\_http\_
    - upstream\_http\_
    - cookie\_
    - arg\_

# Agenda

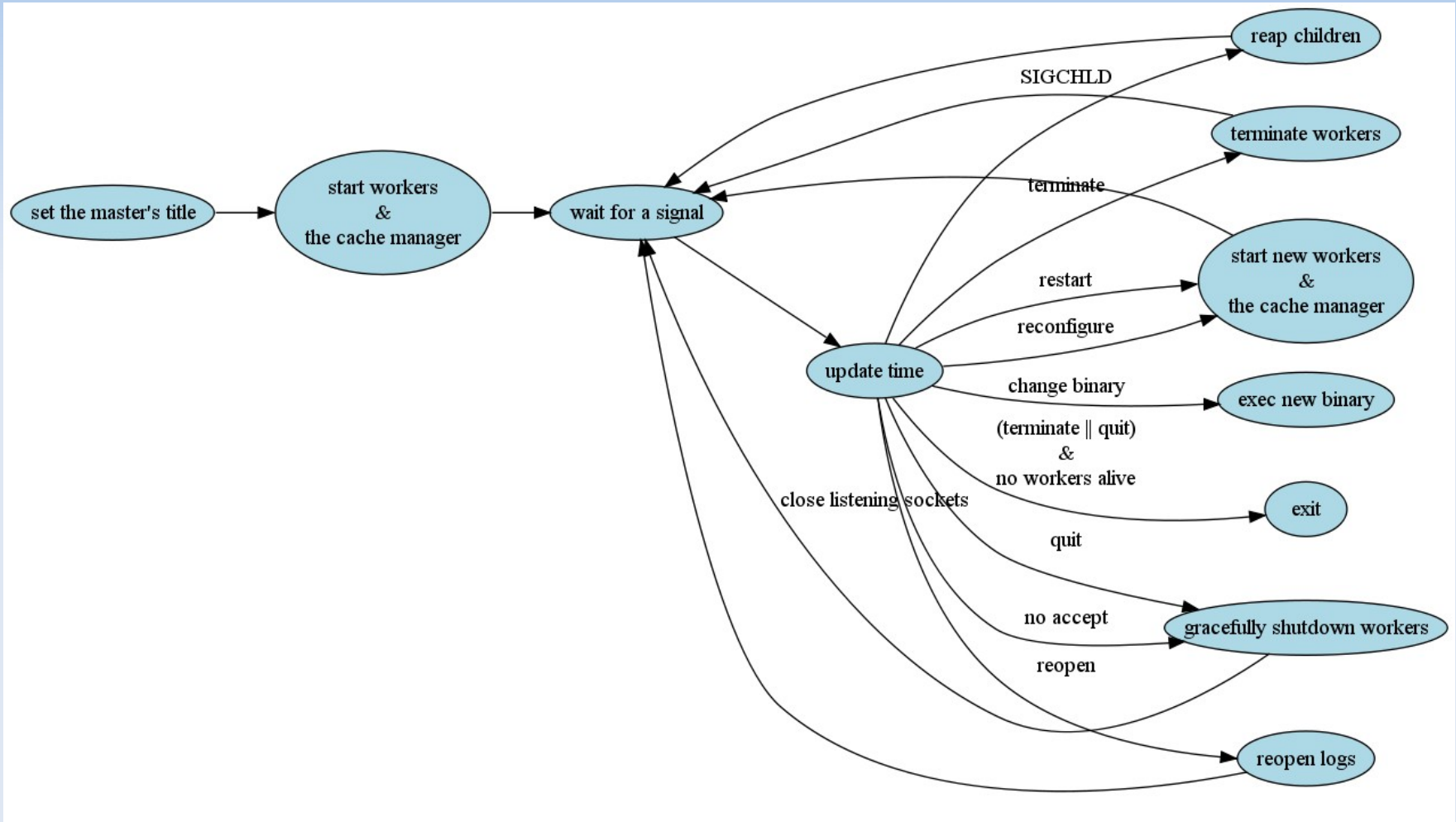
- Source code layout
- Key concepts and infrastructure
- **The event-driven architecture**
- HTTP request handling
- Mail proxying process
- Nginx module development
- Misc. topics

# Master and Workers

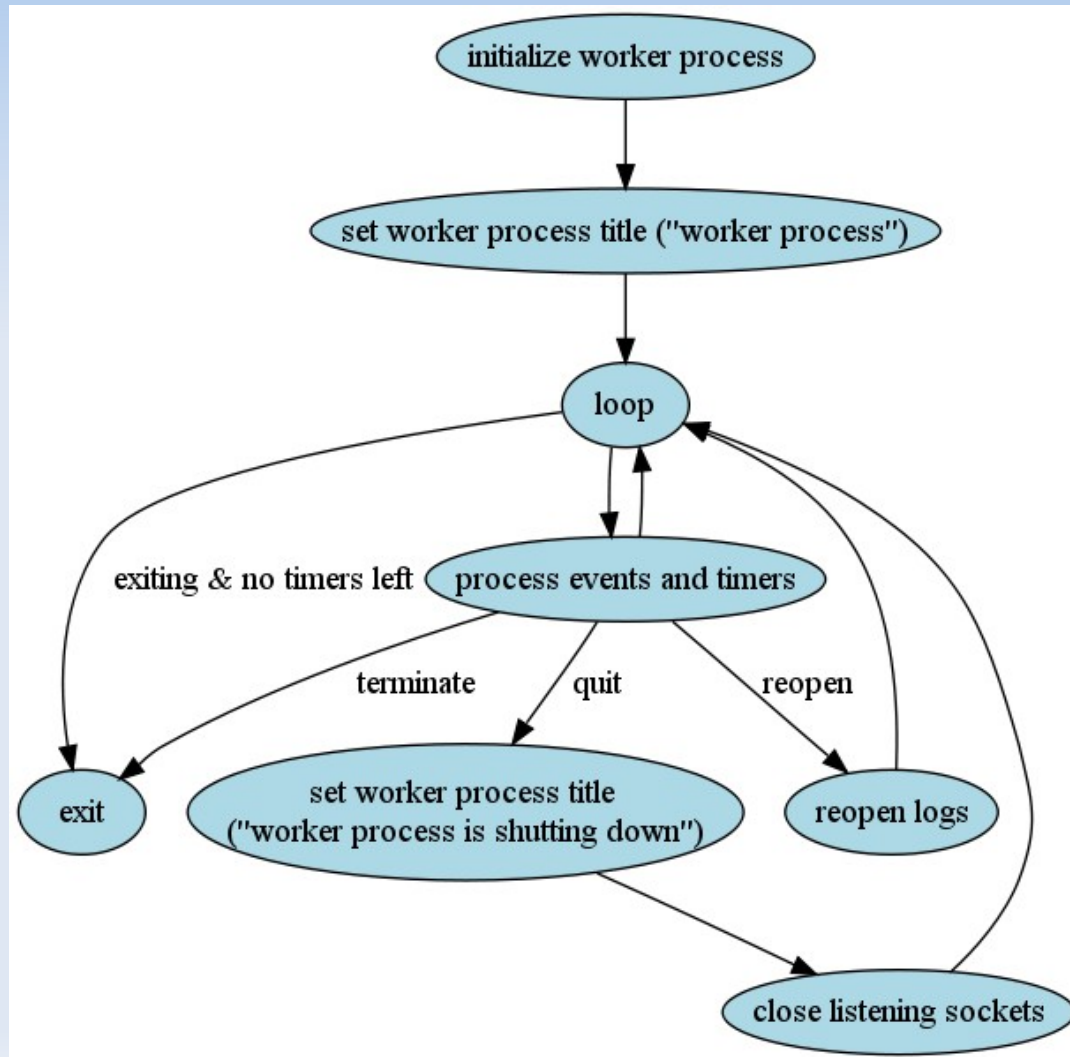
- Master
  - Monitor workers, respawn when a worker dies
  - Handle signals and notify workers
    - exit
    - reconfiguration
    - update
    - log rotation
    - ...
- Worker
  - Process client requests
    - handle connections
  - Get cmd from master



# Master Process Cycle



# Worker Process Cycle



# Inter-process Communication

- Signals
  - Channel
    - socketpair
    - command
- Shared memory
  - Connection counter
  - Stat
  - Atomic & spinlock
  - Mutex

# Event

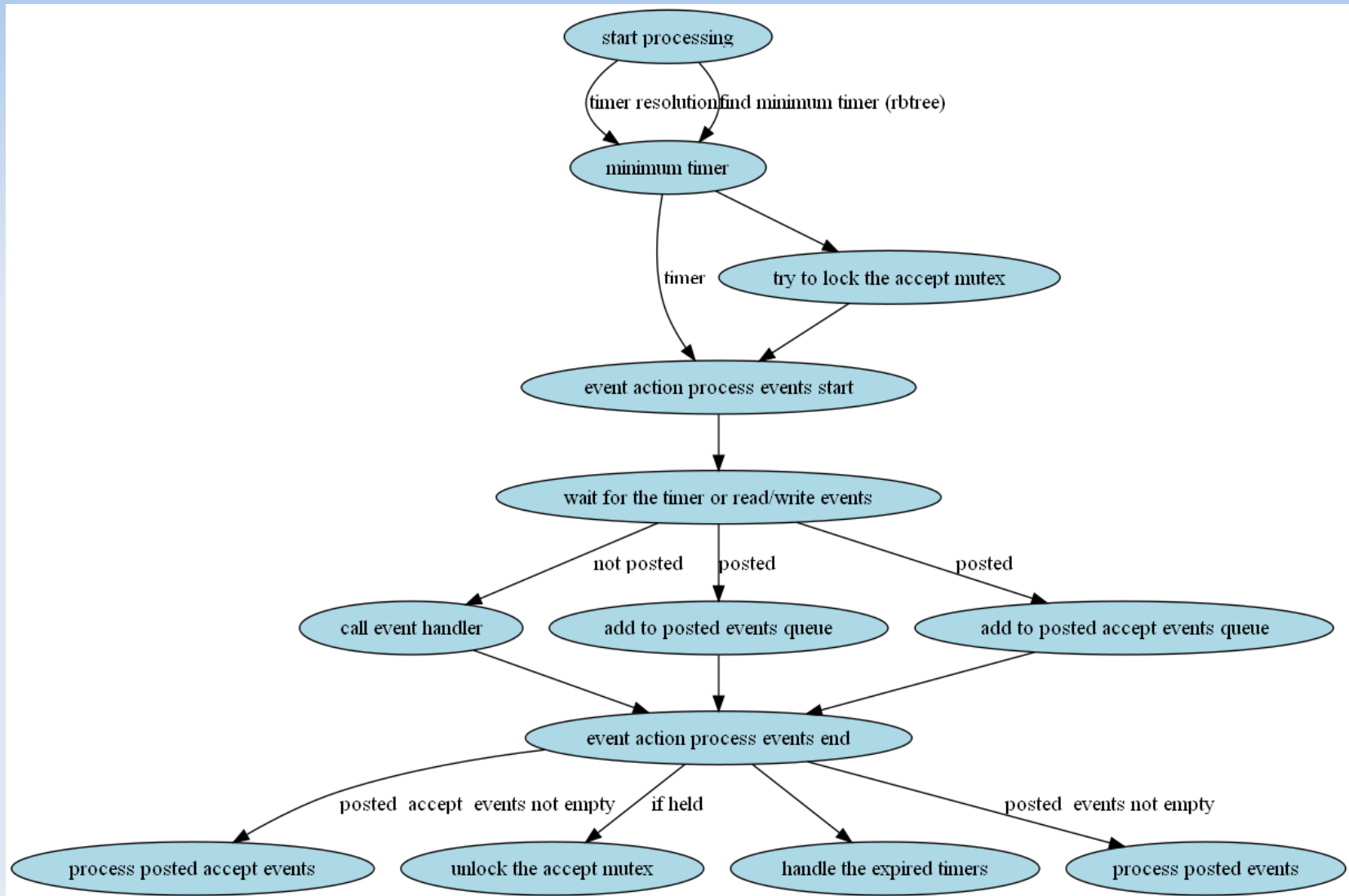
- ngx\_event\_t
  - Read
  - Write
  - Timeout
- Callbacks
- Handlers
  - ngx\_event\_accept
  - ngx\_process\_events\_and\_timers
  - ngx\_handle\_read\_event
  - ngx\_handle\_write\_event
- Posted events
  - Posted accept events queue
  - Posted events queue



# Time Cache

- The overhead of `gettimeofday()`
- Time cache variables
  - `ngx_cached_time`
  - `ngx_current_msec`
  - Time strings
    - `ngx_cached_err_log_time`
    - `ngx_cached_http_time`
    - `ngx_cached_http_log_time`
- Timer resolution
  - Interval timer
    - `setitimer()`

# Events and Timers Processing

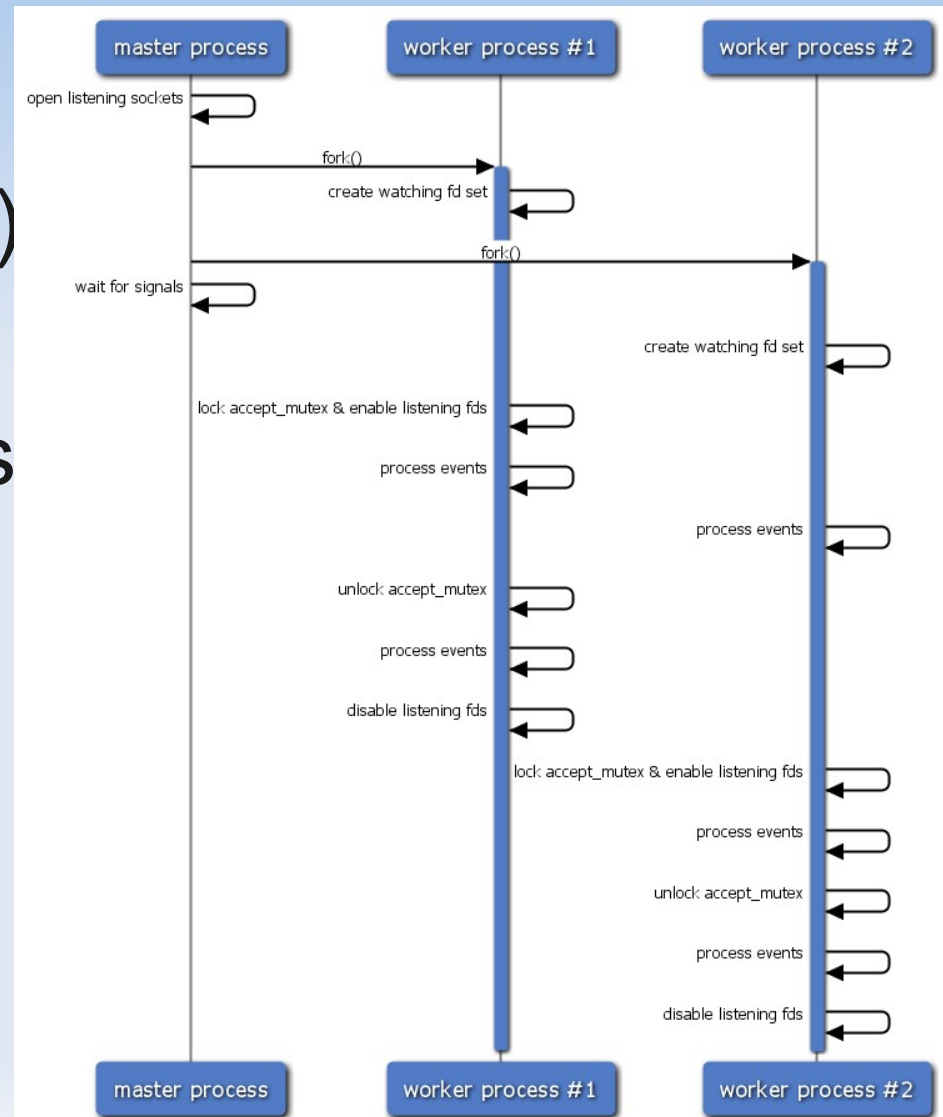


# Timer Management

- Actions
  - Add a timer
  - Delete a timer
  - Get the minimum timer
- Red black tree<sup>[\*]</sup>
  - $O(\log n)$  complexity

# Accept Mutex

- Thundering herd
- Serialize accept()
- Lock/unlock
- Listening sockets
- Delay



# I/O

- Multiplexing
  - kqueue/epoll
    - NGX\_USE\_CLEAR\_EVENT (edge triggered)
  - select/poll/dev/poll
    - NGX\_USE\_LEVEL\_EVENT (level triggered)
  - ...
- Advanced I/O
  - sendfile()
  - writev()
  - direct I/O
  - mmap()
  - AIO
  - TCP/IP options
    - TCP\_CORK/TCP\_NODELAY/TCP\_DEFER\_ACCEPT

# Agenda

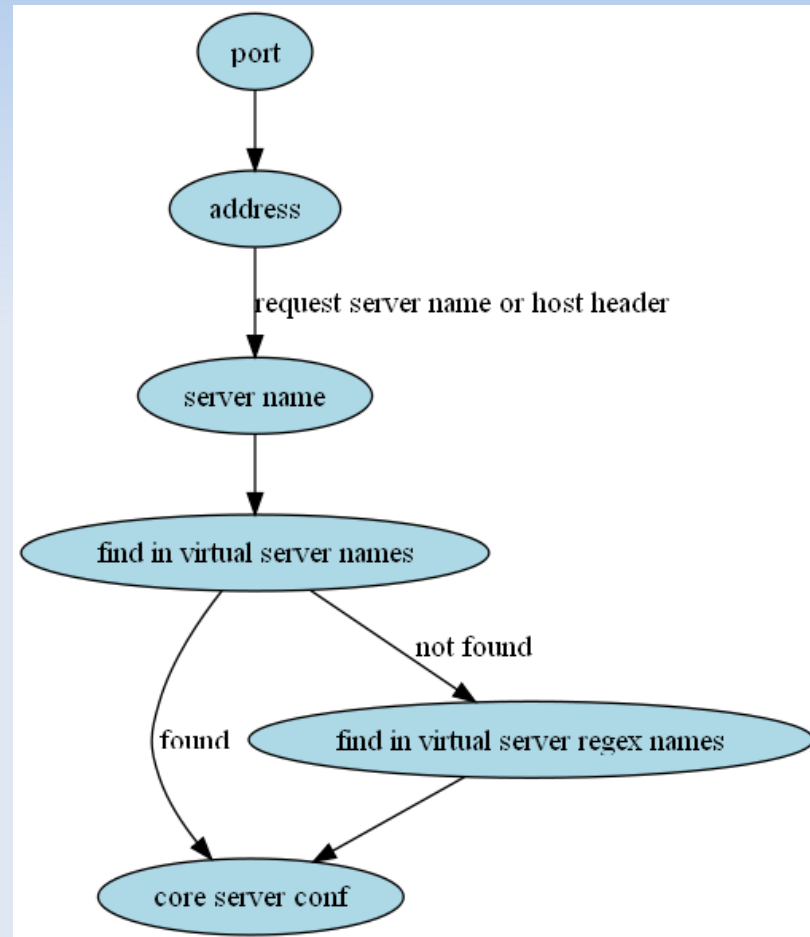
- Source code layout
- Key concepts and infrastructure
- The event-driven architecture
- **HTTP request handling**
- Mail proxying process
- Nginx module development
- Misc. topics

# Important Structures

- Connection
  - ngx\_connection\_t
- HTTP connection
  - ngx\_http\_connection\_t
- HTTP request
  - ngx\_http\_request\_t
    - headers\_in
    - headers\_out
    - ...

# Virtual Servers

- Address
- Port
- Server names
- Core server conf





# Locations

- Location tree
  - Static
  - Regex
    - $= \wedge \sim \sim \sim^*$
- Per-location configuration
  - Value
    - inheritance
    - override
  - Handler
- Named location
  - `try_files/post_action/error_page`

# HTTP Contexts

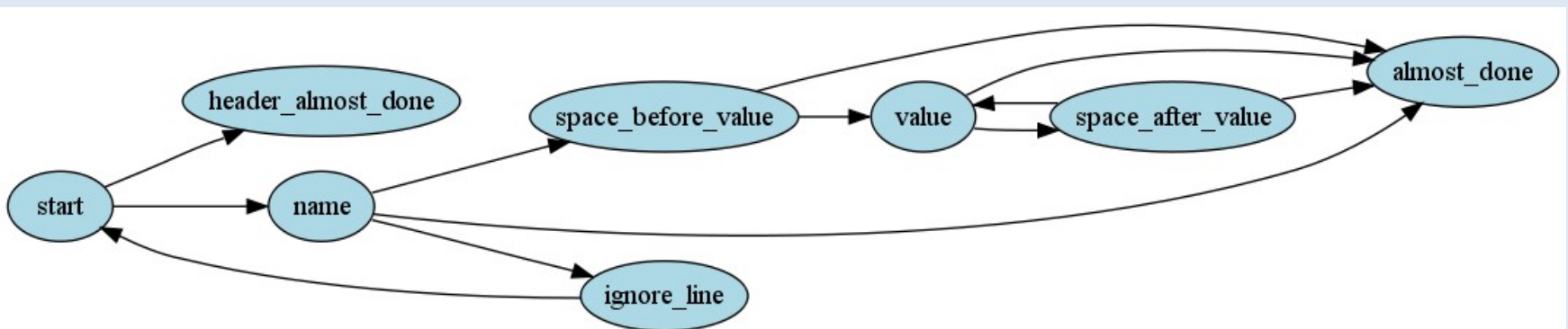
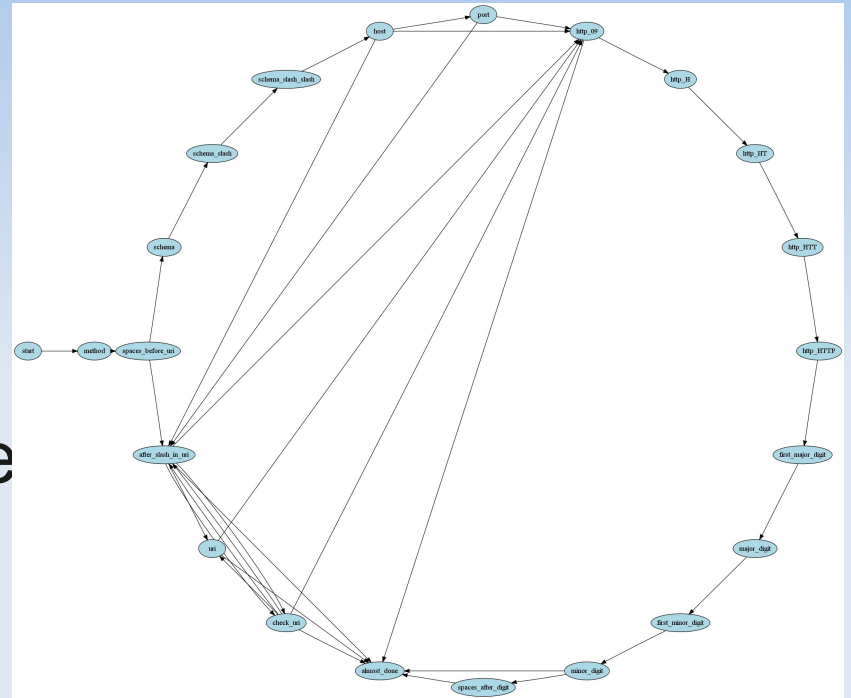
- Types
  - main\_conf
  - srv\_conf
  - loc\_conf
- Request
  - ngx\_http\_get\_module\_main\_conf
  - ngx\_http\_get\_module\_srv\_conf
  - ngx\_http\_get\_module\_loc\_conf
- Parse conf file
  - ngx\_http\_conf\_get\_module\_main\_conf
  - ngx\_http\_conf\_get\_module\_srv\_conf
  - ngx\_http\_conf\_get\_module\_loc\_conf
- Module context
  - ngx\_http\_get\_module\_ctx
  - ngx\_http\_set\_ctx

# HTTP Handling

- Receive data
- Parse the request
- Find the virtual server
- Find the location
- Run phase handlers
- Generate the response
- Filter response headers
- Filter the response body
- Send out the output to the client

# Request Parsing

- Request line
- Headers
- Interesting tricks
  - Finite state machine
  - ngx\_strX\_cmp



# Phases and Handlers

- Phases
  - POST\_READ
  - SERVER\_REWRITE
  - FIND\_CONFIG
  - REWRITE
  - POST\_REWRITE
  - PREACCESS
  - ACCESS
  - POST\_ACCESS
  - TRY\_FILES
  - CONTENT
  - LOG
- Phase handler
  - Checker
  - Handler
  - Next

# Phases and Handlers (cont'd)

- Phase engine
  - Handlers
    - server\_rewrite\_index
    - location\_rewrite\_index
    - r->phase\_handler
- Default checkers
  - ngx\_http\_core\_generic\_phase
  - ngx\_http\_core\_find\_config\_phase
  - ngx\_http\_core\_post\_rewrite\_phase
  - ngx\_http\_core\_access\_phase
  - ngx\_http\_core\_post\_access\_phase
  - ngx\_http\_core\_try\_files\_phase
  - ngx\_http\_core\_content\_phase

# Phases and Handlers (cont'd)

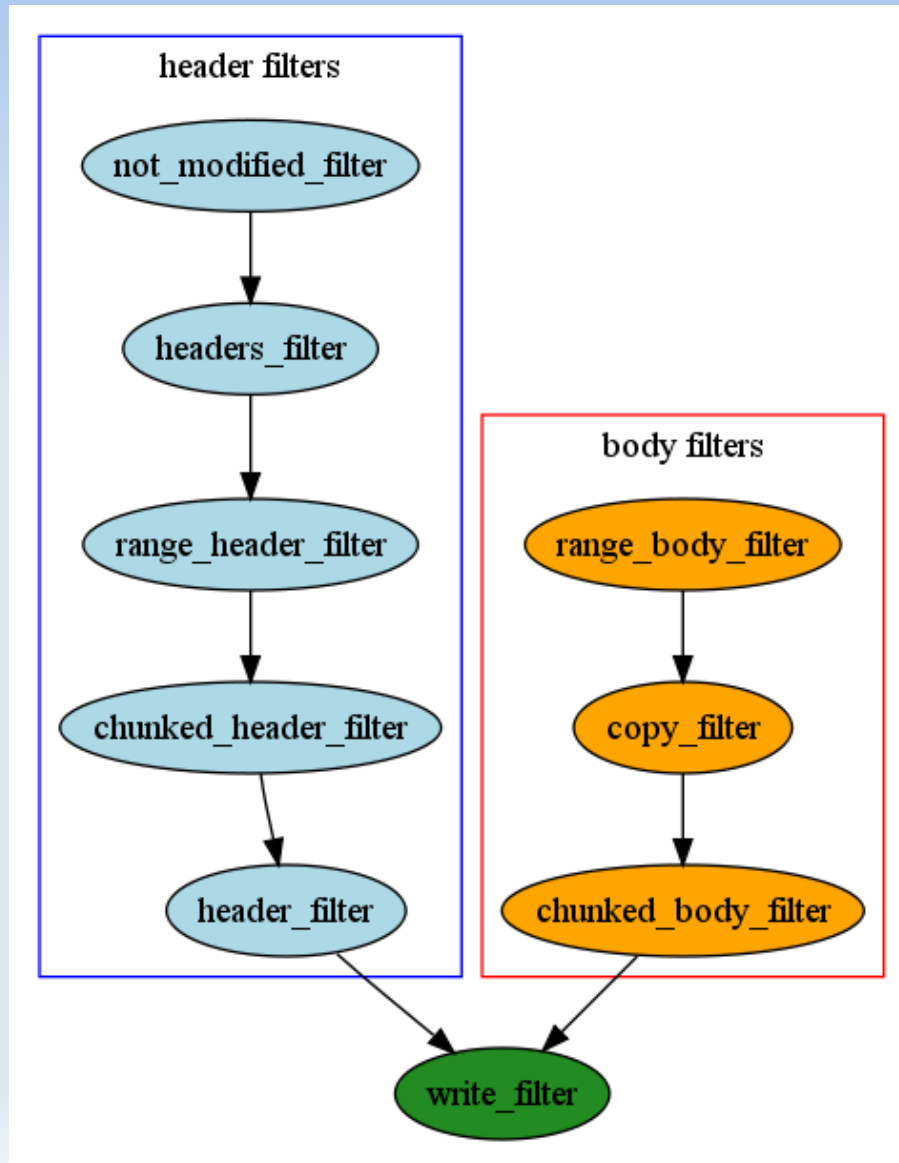
phase	modules
POST_READ	realip
SERVER_REWRITE	rewrite
REWRITE	rewrite
PREACCESS	limit_req, limit_zone, realip
ACCESS	access, auth_basic
CONTENT	autoindex, dav, gzip, index, random_index, static
LOG	log

# Filter Chain

- Singly-linked list like (CoR)
- Filter response only
  - Header filter
  - Body filter
- Send out the response
  - ngx\_http\_send\_header
    - top\_header\_filter
  - ngx\_http\_output\_filter
    - ngx\_http\_top\_body\_filter
  - ngx\_http\_header\_filter
  - ngx\_http\_copy\_filter
  - ngx\_http\_write\_filter
- Process order

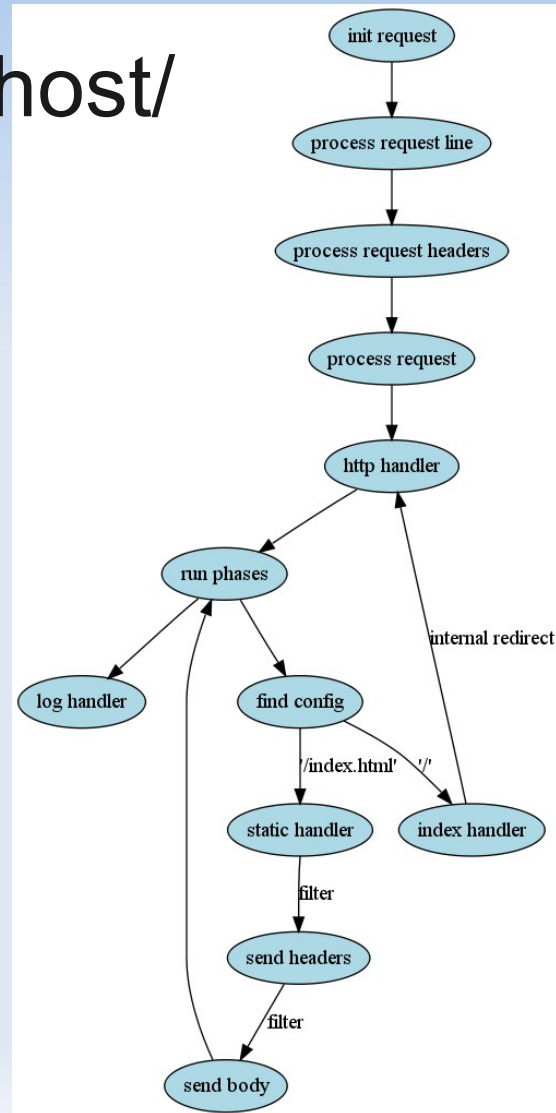


# Filter Chain Example



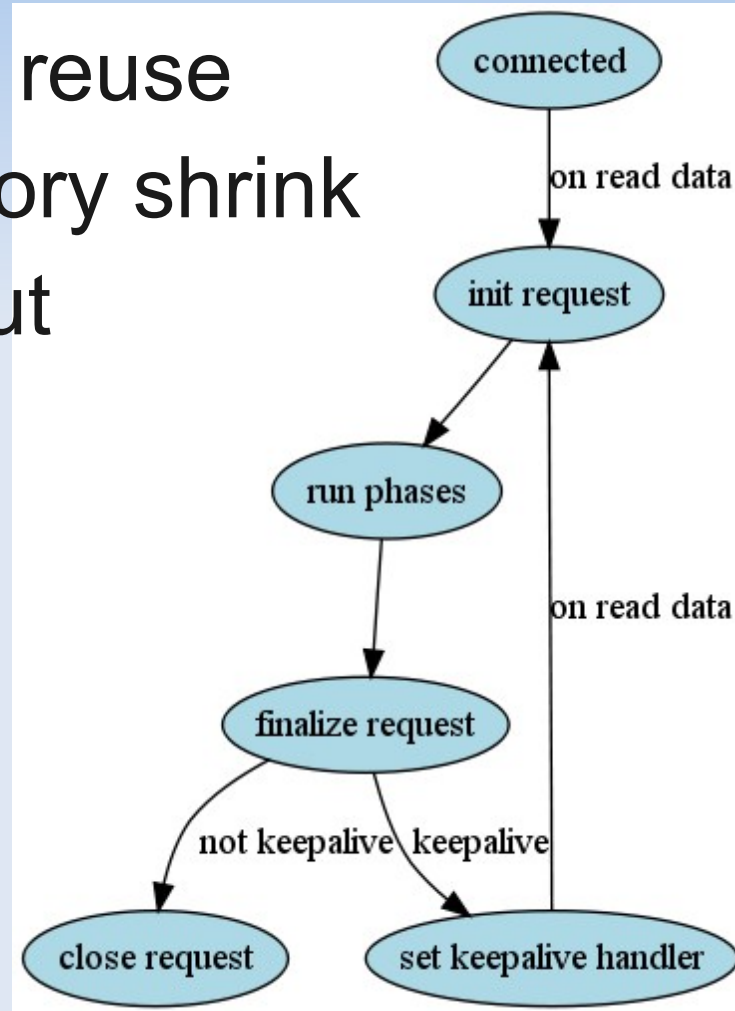
# HTTP Handling Example

- `curl -i http://localhost/`



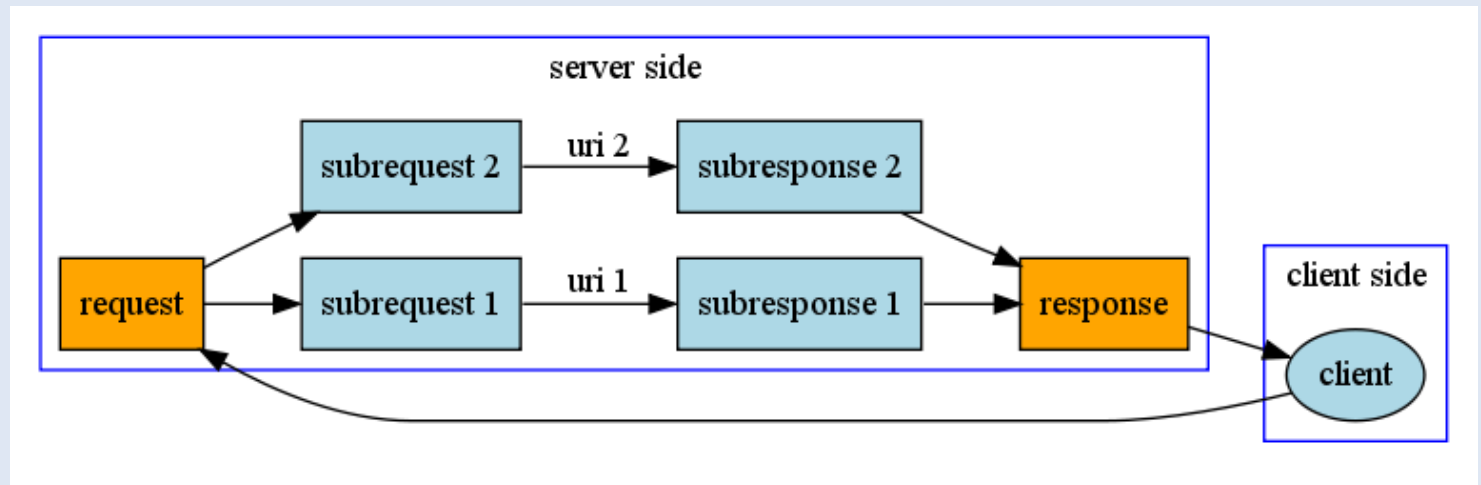
# HTTP Keep-Alive

- Request memory reuse
- Connection memory shrink
- Keep-alive timeout
- Request count



# Subrequest

- Filters
  - Addition filter
  - SSI filter
- Maximum subrequests



# Internal Redirect

- Return a different URL than originally requested
- Examples
  - `try_files`
  - `index/random_index`
  - `post_action`
  - `send_error_page`
  - `upstream_process_headers`

# Upstream

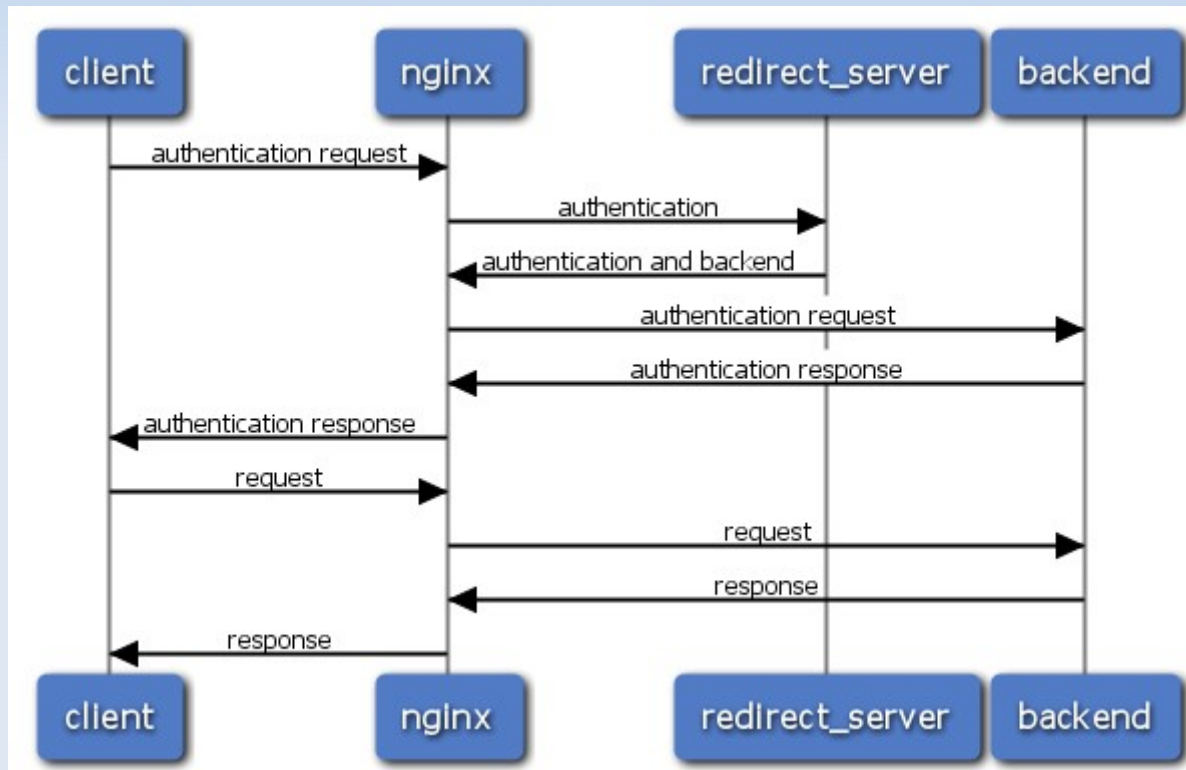
- Hooks
  - input\_filter\_init
  - input\_filter
  - create\_request
  - reinit\_request
  - process\_header
  - abort\_request
  - finalize\_request
  - rewrite\_redirect
- Modules
  - FastCGI
  - Proxy
  - Memcached
- Event pipe
- Load balancer

# Agenda

- Source code layout
- Key concepts and infrastructure
- The event-driven architecture
- HTTP request handling
- **Mail proxying process**
- Nginx module development
- Misc. topics

# Mail Proxy

- Sequence diagram





# Mail Proxy (cont'd)

- Mail session
  - Command parsing
  - Packets relay
- Things you can do
  - Load balancing
  - Authentication rewriting
  - Black lists/white lists

# Agenda

- Source code layout
- Key concepts and infrastructure
- The event-driven architecture
- HTTP request handling
- Mail proxying process
- **Nginx module development**
- Misc. topics

# General Module Interface

- Context
  - index & ctx\_index
- Directives
- Type
  - core/event/http/mail
- Hooks
  - init\_master
    - called at master process initialization
  - init\_module
    - called when the module is loaded
  - init\_process
    - called at worker process initialization
  - exit\_process
    - called at worker process termination
  - exit\_master
    - called at master process termination

# Core Module Interface

- Name
- Hooks
  - create\_conf
  - init\_conf
- Examples
  - Core
  - Events
  - Log
  - HTTP

# Event Module Interface

- Name
- Hooks
  - create\_conf
  - init\_conf
  - event\_actions
    - add
    - del
    - enable
    - disable
    - add\_conn
    - del\_conn
    - process\_changes
    - process\_events
    - init
    - done

# Mail Module Interface

- Protocol
  - type
  - init\_session
  - init\_protocol
  - parse\_command
  - auth\_state
- create\_main\_conf
- init\_main\_conf
- create\_srv\_conf
- merge\_srv\_conf

# HTTP Module Interface

- Hooks
  - preconfiguration
  - postconfiguration
  - create\_main\_conf
  - init\_main\_conf
  - create\_srv\_conf
  - merge\_srv\_conf
  - create\_loc\_conf
  - merge\_loc\_conf

# A “Hello World” HTTP Module

- Creating a hello world! module
  - Files
    - ngx\_http\_hello\_module.c
    - config
  - Build
    - ./configure --add-module=/path/to/hello/module
  - Configuration
    - location & directive



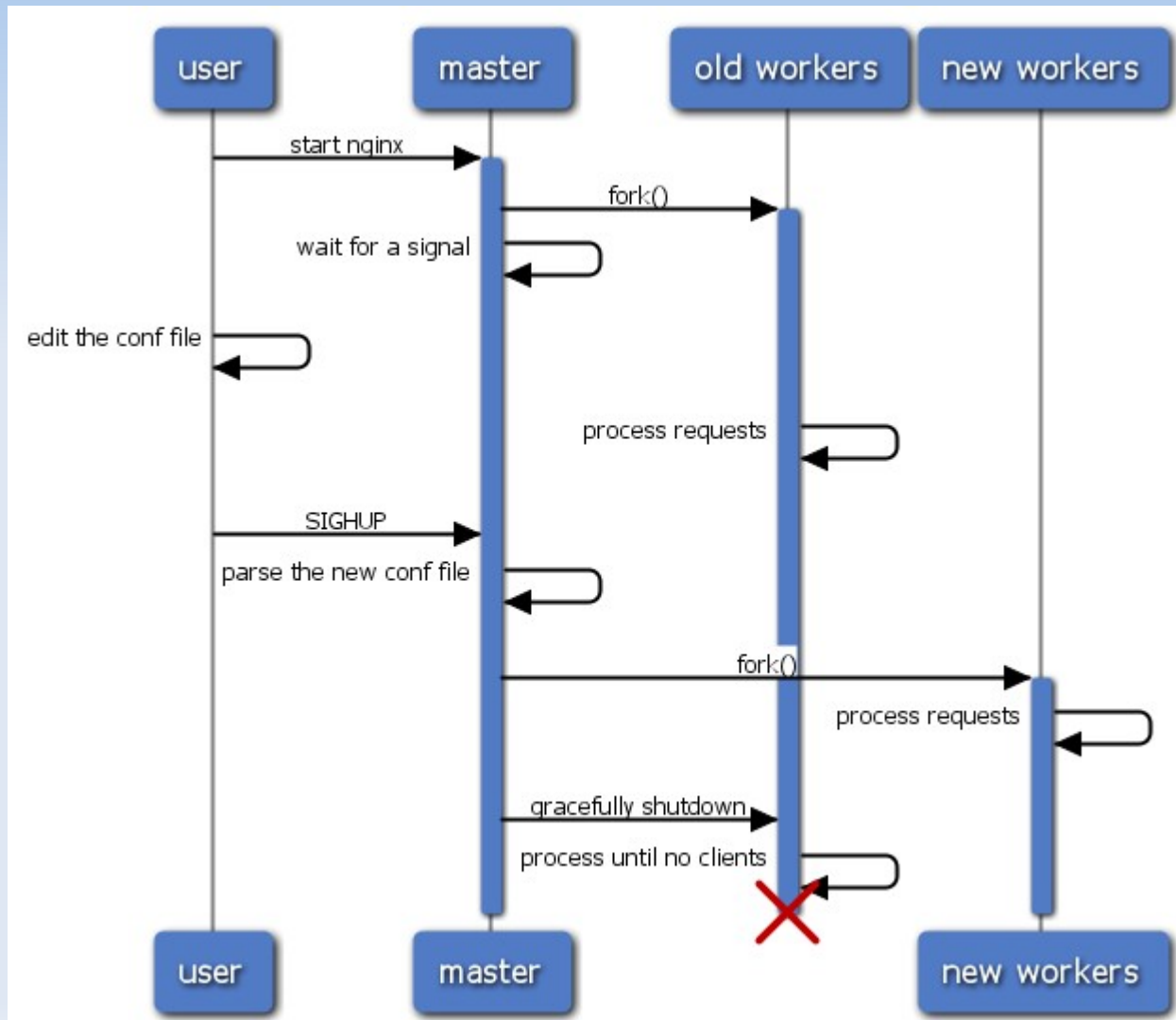
# Agenda

- Source code layout
- Key concepts and infrastructure
- The event-driven architecture
- HTTP request handling
- Mail proxying process
- Nginx module development
- **Misc. topics**

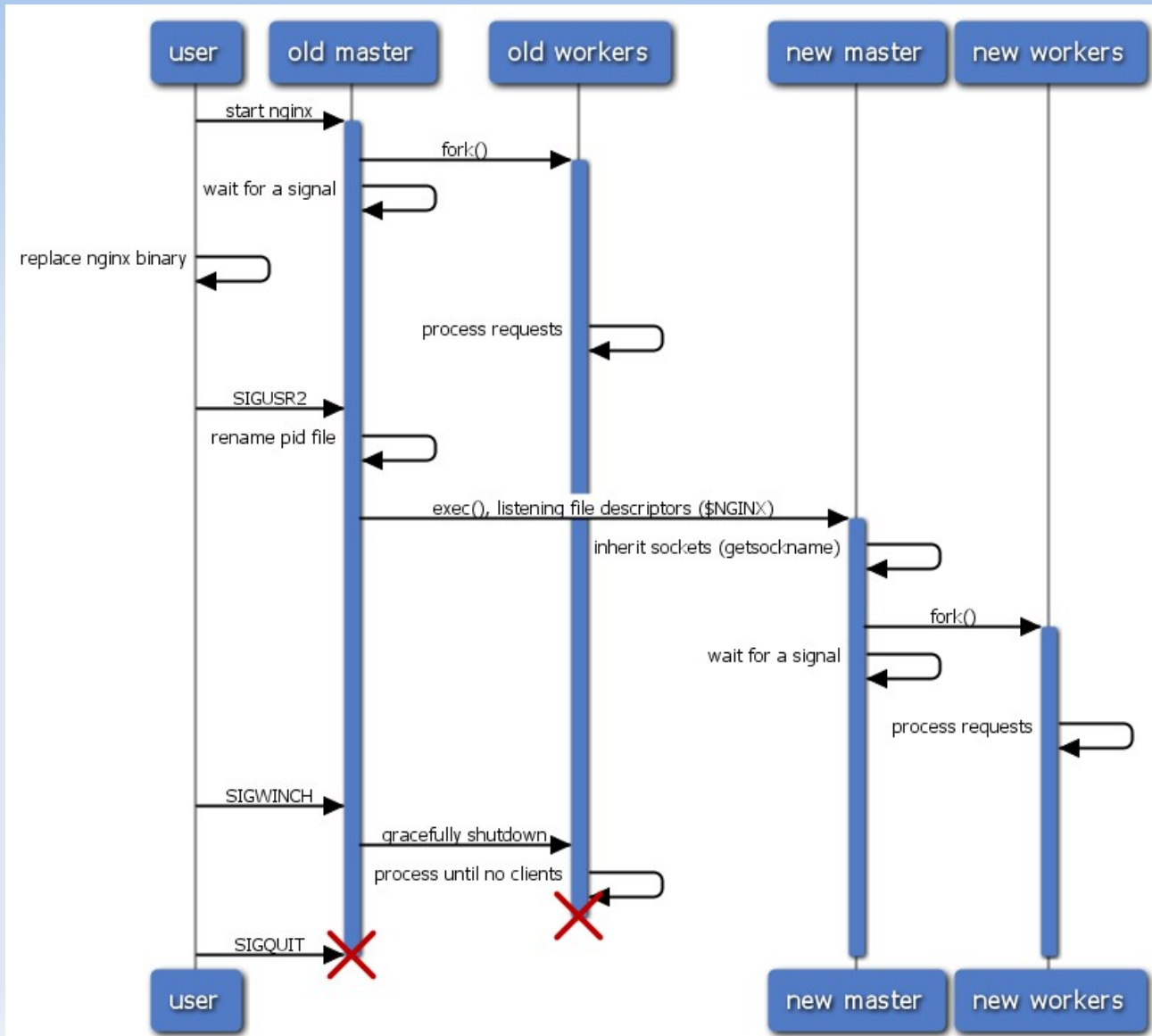
# Auto Scripts

- Handle the differences
  - OS
  - Compiler
  - Data types
  - Libraries
- Module enable/disable
- Modules order

# Reconfiguration



# Hot Code Swapping



# Thank You!

- My site: <http://www.zhuzhaoyuan.com>
- My blog: <http://blog.zhuzhaoyuan.com>