# Whirlwind:

## Overload protection, fault-tolerance and self-tuning in an Internet services platform

Peter Donald <peter@realityforge.org>
La Trobe University, Melbourne, Australia

# Incident Resource Information System

- Used during emergency situations
- Tracks movement of resources
- Generates payments for personnel

# Dramatic load increase during emergencies

- Resources organized into a hierarchical tree modeling geographic regions
- Clients listen to and update specific sub-trees
- Emergencies cluster geographically
- Resource movements increase during emergencies

# Load increase lead to...

- Response time spikes
- Throughput loss
- Unpredictable behavior
- Crashing

# Requirements

- Load management strategies:
  - Admission control
  - Service degradation
  - Throttling
  - Adaptive scheduling policy
- Fault tolerance
- Self-Tuning
- Future hardware upgrade path

# Design Elements

- Processes
- Messages
- Groups
- Principals
- Message Predicates
- Schedulers
- Adaptive Concurrency Level

# Processes

- Isolated
- Sequential
- Cleanup on termination via onTerminate() method
- Normal or non-normal termination
- Monitor other processes

# Messages

- Asynchronous
- Post operation can fail
- Post failure typically indicates overload
- Sender handles failure

# Groups

- Contains processes with similar;
  - resource,
  - computational and
  - concurrency requirements
- Share a;
  - message predicate,
  - thread pool and
  - scheduler
- Cohort scheduling

# Principals

- Single Unit of Work
- Single Unit of Load
- Associated with multiple processes

# Message Predicates

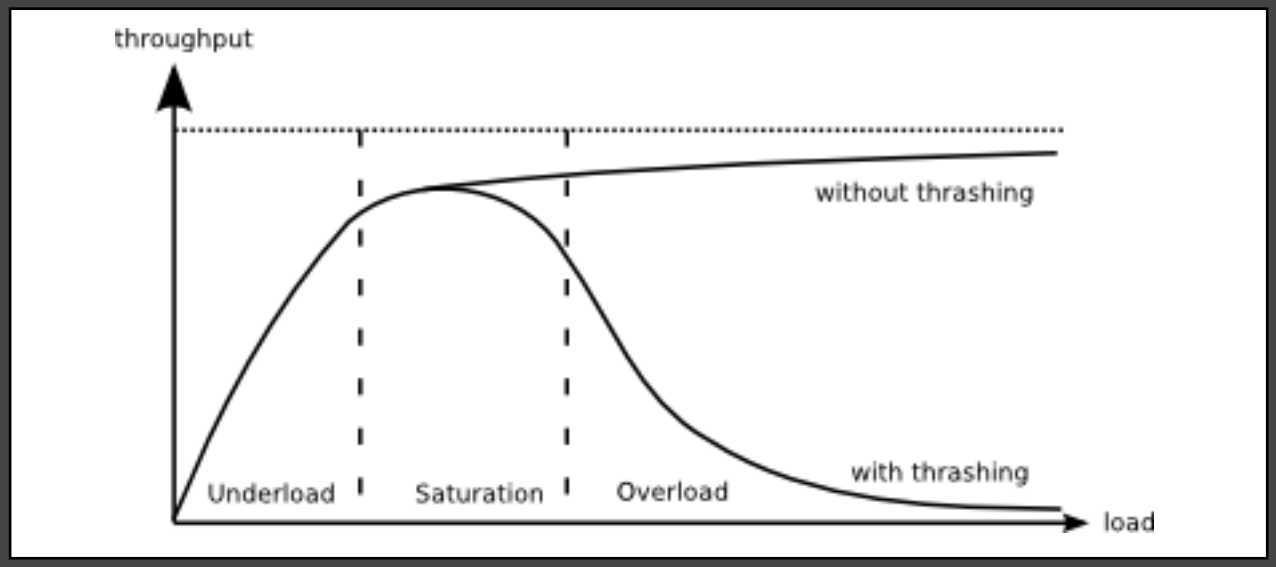
- Queried before message queued
- Rejection typically signals the group is overloaded
- Rejection decision based on;
  - message cost
  - class or priority of principal associated with target process
- Rejection allows sender to determine response
  - Admission control
  - Service degradation

# Schedulers

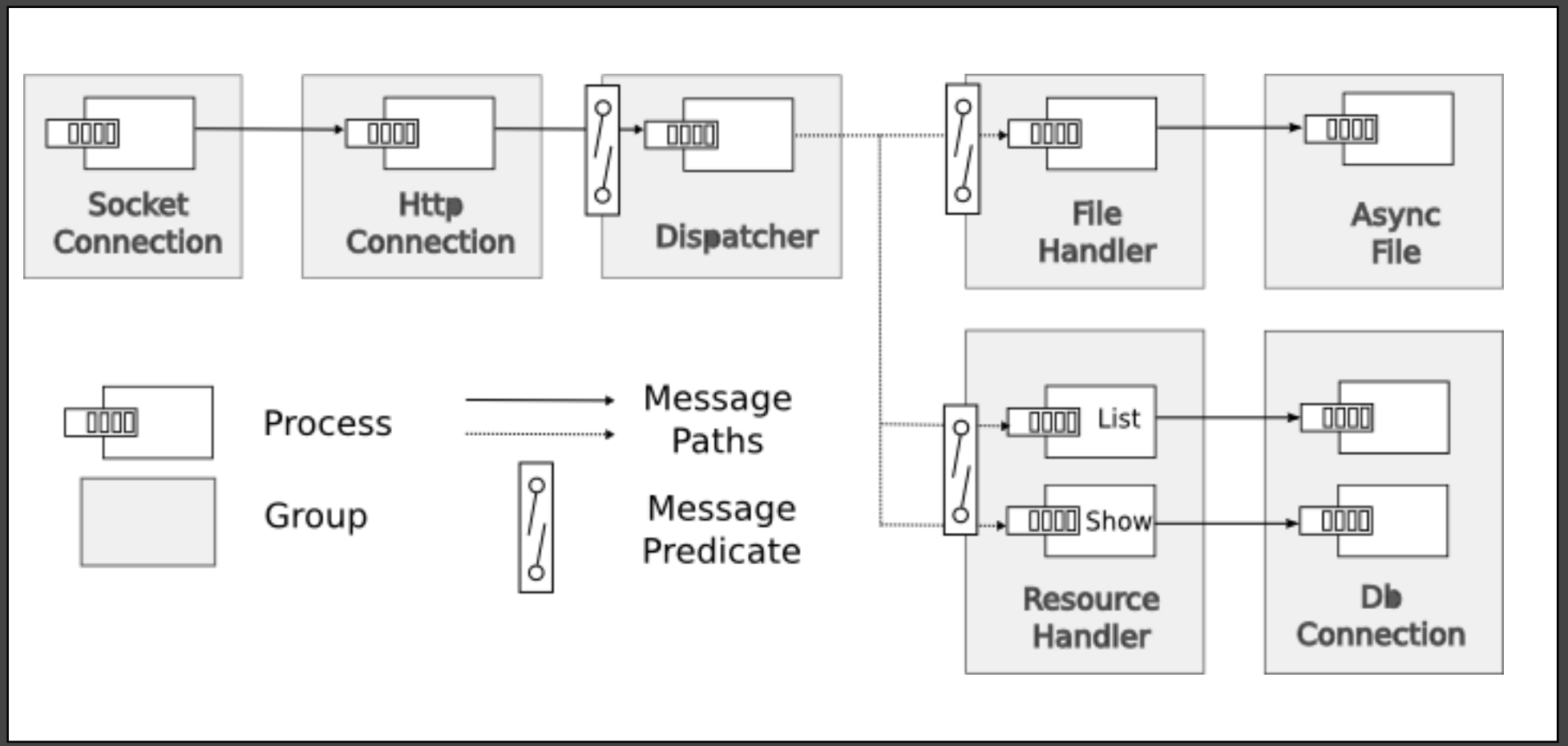
- Determine order that worker threads handle processes
- FIFO by default
- "Request Aware Scheduler" driven by the associated Principal
- "Resource Aware Scheduler" driven by the estimated change in available resources after scheduling a process

# Adaptive Concurrency Level

- Adjust pool size to maximize throughput based on feedback
- Controller based on heuristics or control algorithms such as method of incremental steps



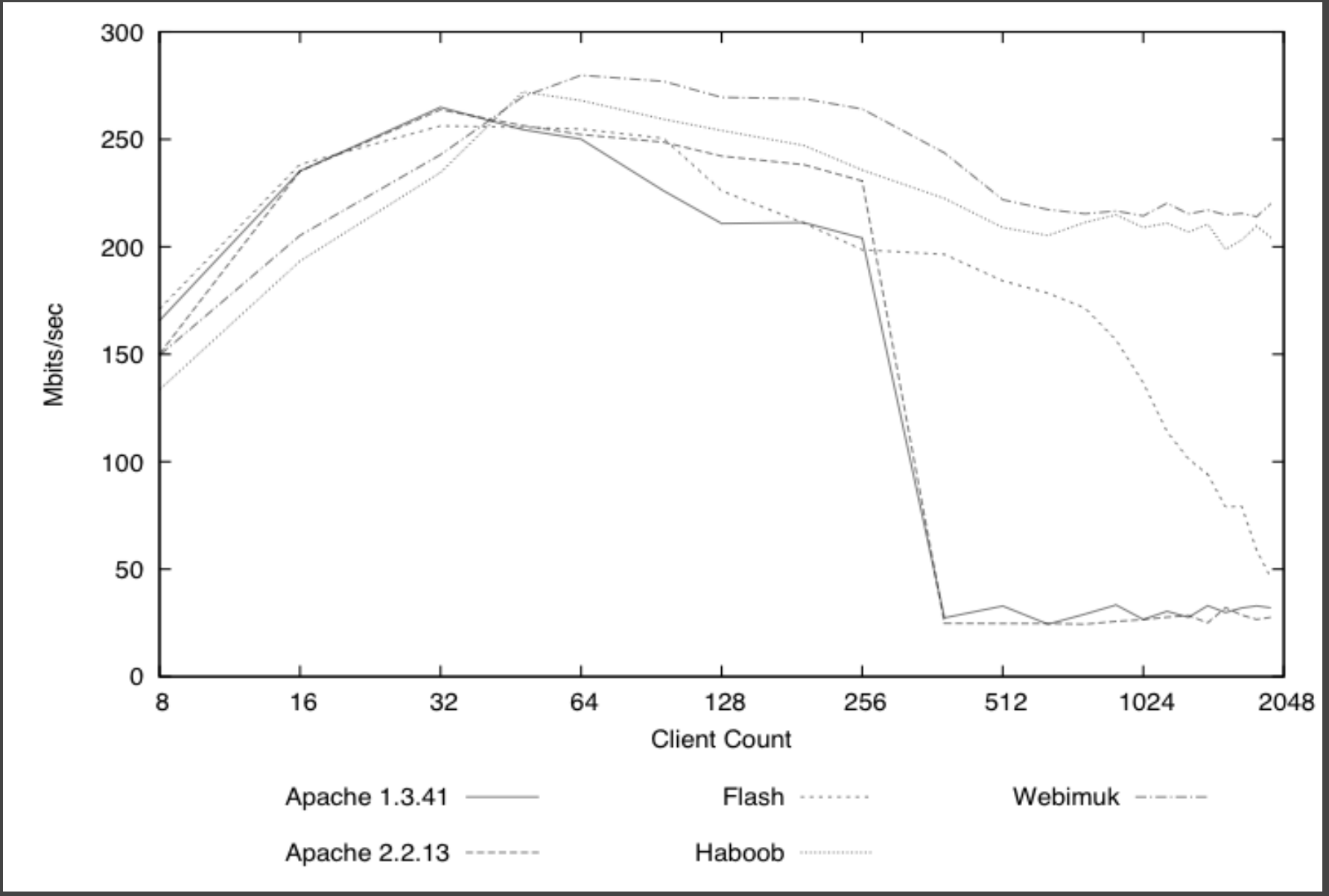
# Example


Socket Connection
Http Connection
Dispatcher
File Handler
Async File
List
Show
Resource Handler
Db Connection
Process
Group
Message Paths
Message Predicate

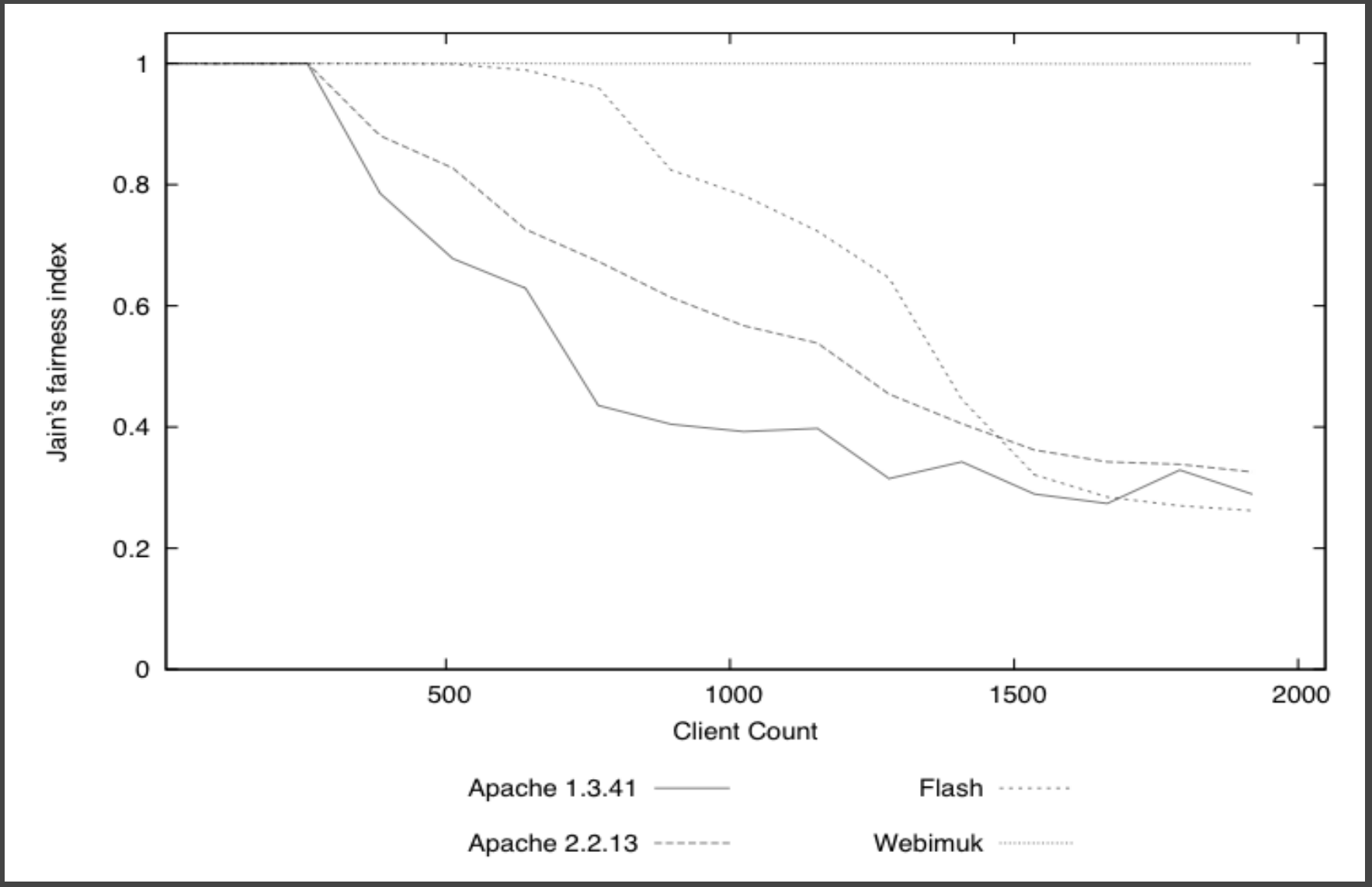
# Evaluation

- Actively used over a period of 5 years
  - Evolution rate slowed down and moved to convenience rather than correctness or performance
- Survived February 2009 Fires.
  - Worst fire recorded in Australia's history
  - Extreme load for long periods of time

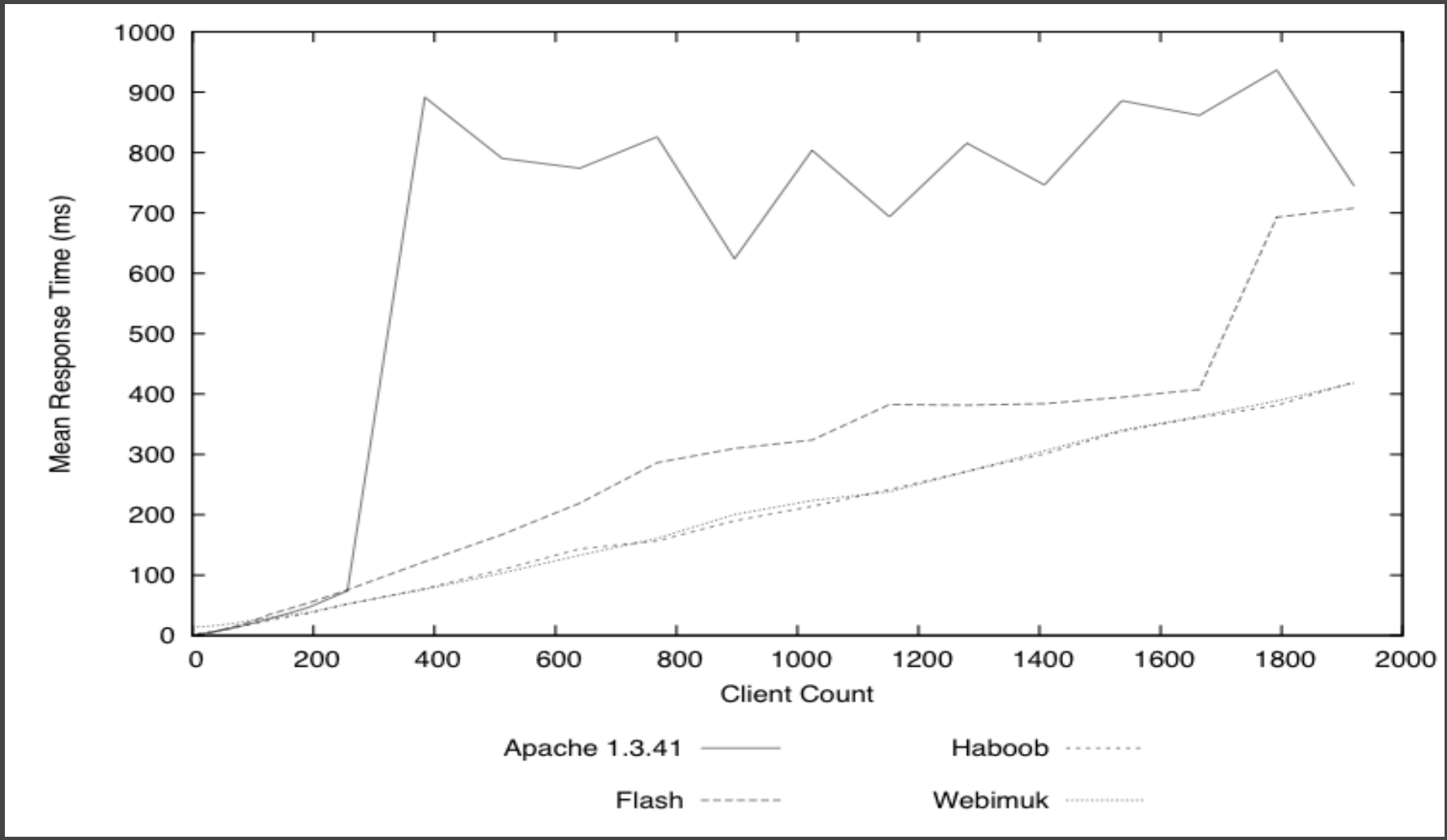
# Performance



# Performance (2)



# Performance (3)



# Questions?