SEDA: An Architecture for Well-Conditioned, Scalable Internet Services

Matt Welsh, David Culler, and Eric Brewer
University of California, Berkeley
Symposium on Operating Systems Principles (SOSP), October 2001

CS533 - Concepts of Operating Systems
Professor Jonathan Walpole

presented by: Eric Wheeler
January 13th, 2009
Motivation

● High-volume Internet Services
● Graceful load degradation at saturation
  – The Slashdot effect
● Dynamic resource scaling
  – Threading
  – Event queueing
SEDAR

- Good performance
- Hybrid design:
  - Dynamic thread scaling
  - Event handling
- Separate framework from protocol code
- Easy for developers
An HTTP Server

[Diagram showing the process of an HTTP server with stages: start, read, accumulate header, header done, disk read, disk done, write response, done, finish]
Threaded Implementation

[Diagram showing network->dispatcher->dispatch to various requests (request 1, request 2, request 3, request 4, request N)]

[Welsh]
Threaded Metrics

- Log Scale
- Throughput falls non-linearly (non-graceful).
- Context overhead dominates performance

Examples: Apache, IIS, Netscape Server
Event Implementation

- Single threaded scheduler
- Simple event handler callback
- IO Must be non-blocking
- eg, Flash web server from Rice University

[Welsh]
Event Metrics

- No OS or thread contexts
- Linear throughput
- Linear latency degradation
  - Lots of state to track
  - Implementations lack modularity
Staged Event-Driven Architecture (SEDA) HTTP Server Implementation

- Separate modular stages
- Separate queues per stage
- Multiple events handled in a batch
Dynamic Scaling

- Change the number of threads in a pool
- Change the number of events per batch
Jain fairness

\[
f(x) = \frac{\left(\sum x_i\right)^2}{N \sum x_i^2}
\]

- \(x_i\) is the number of requests for each of \(N\) clients.
- 1 means equally fair to all clients.
- If \(k\) out of \(N\) receive an equal share of service, then the other \(N-k\) clients receive no fairness.
SEDA HTTP Server Metrics

- Gbit links
- Note the fairness scale
- Apache drops connections at >150 connections
Long tails

Apache and Flash have responses with long several-minute latencies.
SEDA: Graceful Backoff

- Can watch queue length between stages
  - Drop Events?
  - Error to user
  - HTTP Redirect
  - Linear delay by queue depth
Closing

- SEDA:
  - A modular API
  - Simplifies implementation
  - Encourages code reuse
  - Tight latency distribution
  - Graceful Degredation
  - Fairness
References

- [Harke] Content and images taken from the April 8, 2009 presentation of this paper for CS533 at PSU by Tom Harke.
- [Welsh] Images taken from this SEDA paper.