The Gearman Cookbook
OSCON 2010

Eric Day
http://oddments.org/
Senior Software Engineer @ Rackspace
Thanks for being here!
Ask questions!

Grab a mic for long questions.
Use the source...
What is Gearman?
It is not German.

(well, not entirely at least)
A protocol with multiple implementations.
A message queue.
A job coordinator.
“A massively distributed, massively fault tolerant fork mechanism.”

- Joe Stump, SimpleGeo
A building block for distributed architectures.
Features

- Open Source
- Simple & Fast
- Multi-language
- Flexible application design
- Embeddable
- No single point of failure
How does Gearman work?
Your Client Application Code

Gearman Client API
(C, PHP, Perl, MySQL UDF, ...)

Your Application

Gearman Job Server
gearmand

Provided by Gearman

Gearman Worker API
(C, PHP, Perl, ...)

Your Worker Application Code
While large-scale architectures work well, you can start off simple.

Source: 01
Foreground
(synchronous)
or
Background
(asynchronous)
Questions?
Let's get cooking!
Required Ingredients:
Job Server

- **Perl Server (Gearman::Server in CPAN)**
  - The original implementation
  - Actively maintained by folks at Six Apart

- **C Server (https://launchpad.net/gearmand)**
  - Rewrite for performance and threading
  - Added new features like persistent queues
  - Different port (IANA assigned 4730)
  - Now moving to C++
Client API

- Available for most common languages
- Command line tool
- User defined functions in SQL databases
  - MySQL
  - PostgreSQL
  - Drizzle
Worker API

- Available for most common languages
  - Usually in the same packages as the client API
- Command line tool
Optional Ingredients

- Databases
- Shared or distributed file systems
- Other network protocols
  - HTTP
  - E-Mail
- Domain specific libraries
  - Image manipulation
  - Full-text indexing
Recipes

- Scatter/Gather
- Map/Reduce
- Asynchronous Queues
- Pipeline Processing
Scatter/Gather

- Perform a number of tasks concurrently
- Great way to speed up web applications
- Tasks don't need to be related
- Allocate dedicated resources for different tasks
- Push logic down to where data exists
Scatter/Gather

Client

- DB Query
- Full-text Search
- Location Search
- Image Resize
Scatter/Gather

- Start simple with a single task
- Multiple tasks
- Concurrent tasks

Source: 03
Scatter/Gather

- Concurrent tasks with different workers
- All tasks run in the time for longest running
- Must have enough workers available

Source: 04
Note on Resize Worker

![Diagram showing the components and connections between Apache, PHP, Gearman Job Server, Storage, NFS, MogileFS, and Resize Worker.](image-url)
Web Applications

- Reduce page load time with concurrency
- Don't tie up web server resources
- Improve time to first byte
  - Start non-blocking requests
  - Send first part of response
  - Block when you need one of the results
Questions?
Map/Reduce

- Similar to scatter/gather, but split up one task
- Push logic to where data exists (map)
- Report aggregates or other summary (reduce)
- Can be multi-tier
Map/Reduce

Client
Task T

Task $T_0$ Task $T_1$ Task $T_2$ Task $T_3$
Map/Reduce

Client

Task $T_0$

Task $T_1$

Task $T_2$

Task $T_3$

Task $T_{00}$

Task $T_{01}$

Task $T_{02}$
Log Service

• Push all log entries to log_collect queue
  • `tail -f access_log | gearman -n -f log_collect`
  • Natural spreading between workers when busy
  • Can shutdown workers to help balance
• Worker for each operation per log server
  • Push operations to where data resides
Log Service

Source: 05
Questions?
Asynchronous Queues

• They help you scale
• Not everything needs immediate processing
  • Sending e-mail, tweets, …
  • Log entries and other notifications
  • Data insertion and indexing
• Allows for batch operations
Delayed E-Mail

- Replace:

  # Send email right now
  mail($to_address, $subject, $body, $headers);

- With:

  # Put email in queue to send
  $client = new GearmanClient();
  $client->addServer('127.0.0.1', 4730);
  $client->doBackground('send_email',
                        serialize($email_options));

Source: 06
Database Updates

- Also useful as a database trigger
- Start background jobs on database changes
- Requires MySQL UDF package

```sql
CREATE TRIGGER tweet_blog
BEFORE INSERT ON blog_entries
FOR EACH ROW
    SET @ret=gman_do_background('send_tweet',
                               CONCAT(NEW.title, " - ", NEW.url));
```
Questions?
Pipeline Processing

- Some tasks need a series of transformations
- Chain workers to send data for the next step

![Diagram showing pipeline processing with tasks and operations]

Client
Task T

Worker
Operation 1

Worker
Operation 2

Worker
Operation 3

Output
Search Engine

- Insert URLs, track duplicates
- Fetch contents of URLs
- Store URLs with title and body
- Search stored URLs
Questions?
Persistent Queues

- By default, jobs are only stored in memory
- Various contributions from community
  - MySQL/Drizzle
  - PostgreSQL
  - SQLite
  - Tokyo Cabinet
  - memcached (not always “persistent”)
Persistent Queues

• Use at your own risk, test in your environment!
• Configure back-end to meet your performance and durability needs
Timeouts

• By default, operations block forever
• Clients may want a timeout on foreground jobs
• Workers may need to periodically run other code besides job callback

Source: 09
gearmand --help

- **--job-retries** - Prevent poisonous jobs
- **--worker-wakeup** - Don't wake up all workers for every job
- **--threads** - Run multiple I/O threads (C only)
- **--protocol** - Load pluggable protocols (C only)
New Distributed Applications

- Think of scalable cloud architectures
- Not just LAMP on a virtual machine
- Elastic servers and services (workers)
- New data models
  - Use eventual consistency whenever possible
- Blogs, wikis, and other web apps powered by EC and queues, not a single logical database
Get involved!

- http://gearman.org/
  - Mailing list, documentation, related projects
- #gearman on irc.freenode.net
- Contact me at: http://oddments.org/
- Stickers!