Stateful Web API Servers

Concepts

Stateless Protocols

- Stateless protocols
  - Each packet of information sent by the client to the server is meaningful in isolation
- Scale up
  - Multiple servers and/or server processes
  - Stateless allows clients to communicate with any server or server process
  - Load balancing
    - Clients can even switch around to different servers or server processes

HTTP/HTTPS Is Stateless

- HTTP/HTTPS is designed to be stateless
- Allows scale up

Web Servers Are Stateless by Default

Web servers:

- Use the HTTP/HTTPS protocol
- Are stateless by default
- Can scale up

Stateful Web Servers

- Consider the simple case where a user visits a website without logging in:
  - User first visits the website.
  - Web server creates a unique SID (session ID).
  - Web server stores session data using the SID as a key.
  - Web server creates a web header with the SID as a cookie.
  - User sets the SID cookie.
  - Subsequent requests from the user have the SID cookie.
  - Web server can retrieve and update the session data using the SID cookie from the user.
  - Web server is now stateful.

Stateful Web Servers (cont.)

- Note that the stateful web server we just created
  - Can still scale up
  - Allows users to contact any web server
  - Facilitates load balancing
- Cookies live for days, weeks, or months
  - Rebooting does not clear cookies nor stop tracking

Website Tracking

- Websites track users
- They use SID and other cookies to create a stateful process to track user activity
- Users do not have to be logged in to be tracked
- Cookies can be valid for days, weeks, or months
  - Rebooting your computer does not clear cookies
- Turn off cookies
  - Most websites will not function with cookies turned off
- Government restrictions
  - Notice how some websites warn you about cookies

Stateful Web Server Login

- User logs into a website.
- Web server authenticates the login and creates a unique SID (session ID).
- Web server stores session data using the SID as a key.
- Web server creates a web header with the SID as a cookie.
- User’s browser sets the SID cookie.
- Web server is now stateful.

Stateful Web Server Transactions

- Subsequent requests from the user have the SID cookie.
- Web server can retrieve and update the session data using the SID cookie from the user.

Stateful Web Server Logout

- User sends logout message with SID.
- Web server destroys the session, the SID, and session data.
- Web server notifies user that logout was successful.
- Subsequent attempts to use the SID will receive an error message from the web server.

Staying Logged In for Days, Weeks, etc.

- Cookies can live for days, weeks, or months.
- Cookies are not cleared on reboot or power off.
- SID is stored in a cookie.
- Website logins based on a SID cookie can stay logged in for day, weeks, or longer, even between reboots.

Stateful Web API Server Login

- User program makes a web API call to login and passes credentials.
- Web server authenticates the login and creates a unique SID.
- Web server stores session data using the SID as a key.
- Web server typically passes the SID back using JSON.
- Web API server is now stateful.

Stateful Web API Server Transactions

- User program will typically need to include the SID in JSON on subsequent API calls.
- Web server can retrieve and update the session data using the SID from the user program.

Stateful Web API Server Logout

- User program makes logout API call using SID typically in JSON.
- Web server destroys the session, the SID, and session data.
- Web returns the API call with successful logout.
- Subsequent attempts to use the SID in API calls will results in an error.

The End

Stateful Web API Servers

Business Cases

Rate Limiting

- API calls are rate limited:
  - Per day
  - Per hour
  - Per minute
  - Varies by subscription level
- Stateful session data allows us to keep track
- It’s important because a single program on a laptop or desktop can generate tens of thousands of API calls per second

Free Accounts

- Free accounts are marketing ploys to allow users to try out an API in hopes they will like it and pay for a subscription.
- Students often use free accounts because they do not have big budgets like companies have.
- Free accounts typically come with low rate limits.

Rate Limiting Tips

- Throttle API calls using sleeps between each API call.
- Check each API call return for rate-limiting data.
- Pause before you get rate limited.
- If you do get rate limited, pause for an hour and increase your sleep cycles.

Building in Rate Limiting

- Anytime we create a publicly facing API, we need to make sure we build in rate limiting.
- Otherwise, it is very easy for a single program to overload a server.

Typical Sequence for an API Design

- Users login
- API calls
- Users logout

Map API

- Validate an address
- Give an address, get the point in latitude, longitude
- Given a point in latitude, longitude, get a list of the closest addresses to the point
- Give a box and get a list of all addresses in the box
- Given a point, get a list of closest business based on keyword

Weather API

- Get current weather data for a city
- Get historical weather data for a city
- Get current weather data for a list of cities
- Get historical weather data for a list of cities

Stock Quote API

- Get current stock data for a stock symbol
- Get historical stock data for a stock symbol
- Get current stock data for a list of stock symbols
- Get historical stock data for a list of stock symbols
- Create a watch list of stocks
- Add a stock to a watch list
- Get a list of watch lists
- Get a list of stocks for a watch list
- Get current stock quotes for stocks in a watch list

Cloud API

- Create a VM from an image
- Get a list of VMs and running state
- Create a firewall rule
- Get a list of firewall rules
- Allows us to write logic to create and monitor cloud-based clusters of VMs

Email API

- Send an email
- Get new emails matching on parameters like sender, subject, etc.
- Allows us to write logic on top of API such as:
  - Loop through a list of customers and send them an account-specific email
  - Auto reply to an email with dynamic content

SMS (Text Message) API

- Send a text message
- Read a text message
- Allows us to write logic on top of API such as:
  - Loop through a list of customers and send them an account-specific text messages
  - Auto reply to a text message with dynamic content

The End

Scaling Up Web API Servers

Concepts

Simple Web Server Architecture

- Single web server
- Database

Static Content

- Files are changed at the server level, not at the user request level.
- Examples
  - Text: html
  - Formatting: css
  - Images: png, jpg, jpeg, gif
  - Audio: mp3
  - Video: mp4
  - Client-side scripts: javascript
  - Compressed files: zip, gz, 7z
  - Excel files: xlsx

CDN: Content Delivery Network

- Hundreds of edge servers all over the world
- Serve static files to users from the closest edge server
- Examples
  - User in Sydney, Australia would get a file from local edge server
  - User in Munich, Germany would get the same file from their local edge server
  - Likewise, users in Hong Kong, Buenos Aires, and Berkeley would get the same file from their local edge servers

Move Static Content to CDN

- First step: move static content to CDN
- Offloads a lot of work from our web server
- Users are getting a lot of content from local edge servers, so it makes it appear our web server is running a lot faster than it is
- Easy fix, yet very impactful

Reverse Proxy

- Placed between a web server and the public internet
- Recall that static content is best handled by a small number of threads which can each serve thousands of user connections
- Also protects against attacks like denial of service
- Serves static content (usually to the CDN)
- Passes dynamic content requests to the web server

Hot Railing

- Connection between reverse proxy and web server often called a rail (or railing) in slang
- High speed connection often called a hot rail (or hot railing) in slang

Scaling Up Reverse Proxies

- Multiple reverse proxies
- Can be spread out in multiple data centers all over the world, as needed
- Load balancers often used to increase/decrease the number of reverse proxies and balance traffic to/from them

Application Servers

- Web servers can perform the application logic.
- Alternatively, application logic can be moved to application servers.
- Web servers are connected to application servers (can also be hot railed).

Application Servers (cont.)

- Application servers can scale up.
- Also, separating web server logic from application server logic helps with scale up.
- Typically, there are more application servers than web servers.
- Application servers do not have to be in the same data center as the web servers.
  - Not as common as with reverse proxies and web servers

Transactional Database

- Application servers need to communicate with a transactional database
  - Or web servers in the absence of application servers
- Transactional databases are not able to scale up very much
- Weak link
- Final frontier in web server and web API server architecture

Scaling Up the Database Layer

- Cross-pollination of ideas between big data database scale up and web server database layer scale up
- Immutable model
  - Bulk inserts, bulk deletes, no updates
  - Eventual consistency (BASE)
- Allows us to scale up static or semi-static parts of the database layer that do not change that often

Scaling Up Web API Servers

Same architecture as scaling up web servers

Summary of Scale Up

- Static content is pushed using CDN
- Public internet is connected to reverse proxies
- Reverse proxies are hot railed to web servers
- Web servers are hot railed to application servers
- Application servers are hot railed to a transactional database
- Semi-static pieces of the transactional database layers can be pushed out to application servers using the big data immutable model

The End

Scaling Up Web API Servers

Business Cases

Video Streaming Service, Part I

- Current frequently watched videos would be pushed to CDN edge servers all over the world.
- Less frequently watched videos could be stored in the reverse proxies.
- Infrequently watched videos could be moved to reverse proxies on an as-needed basis.

Video Streaming Service, Part II

- Semi-static parts of the database layer that change once a day can be pushed to the reverse proxy layer or CDN edge servers.
  - Movie details such as title, stars, plots, images, etc.
- Semi-static parts of the database layer that could tolerate an update a few times per day can be pushed to the reverse proxy layer or CDN edge servers.
  - Movie ratings and reviews

Video Streaming Service, Part III

- At this point the web servers and app servers only have to handle such things as logins, logouts, account information and updates, movie searches, etc.

Online Store, Part I

- Current frequently viewed items can be pushed using CDN to edge servers all over the world.
  - Product name, description, images, reviews, etc.
- Less frequently viewed items can be pushed to reverse proxies.
- Infrequently viewed items can be pushed to reverse proxies on an as-needed basis.

Online Store, Part II

- Semi-static parts of the database layer that could tolerate an update a few times per day can be pushed to the reverse proxy layer or CDN edge servers.
  - Pricing, product ratings, and reviews

Online Store, Part III

- At this point, the web servers and app servers only have to handle such things as logins, logouts, account information and updates, product searches, etc.

Airline Reservations, Part I

- Current frequently viewed items can be pushed using CDN to edge servers all over the world.
  - Images, cities served, policies, procedures, airport information, etc.
- Less frequently viewed items can be pushed to reverse proxies.
- Infrequently viewed items can be pushed to reverse proxies on an as-needed basis.

Airline Reservations, Part II

- Semi-static parts of the database layer that could tolerate an update a few times per day can be pushed to the reverse proxy layer or CDN edge servers.
  - Flight schedule, fares (pricing), taxes, airport fees, FAA fees, etc.

Airline Reservations, Part III

- At this point, the web servers and app servers only have to handle such things as logins, logouts, account information and updates, retrieving flight booking levels, actual reservations, etc.

The End

Screen Scraping Web Pages

Concepts

No API Provided

- Suppose a website has information we need.
- We would like to gather information from the website to load into our databases for analytics.
- We look to see if the website has an API and confirm no API is available.
- We look to see if downloads are available and confirm no downloads are available.

Reasons for No API, Part I

- APIs may be hard to create if the website was not designed to accommodate an API from the ground up
  - Retrofitting an API is expensive.
  - Retrofitting an API may cause stability issues with the website.
- If they provide an API or download, you might not visit the website as much
- Website purchased data and their licensing does not allow an API

Reasons for No API, Part II

- Website is to stimulate interest in their commercial software products that contain the data
- Website is to stimulate interest in their commercial download subscriptions
- Market is big companies—they do not want to deal with the small companies or individuals

Reasons for No API, Part III

- Open question: Does adding an API increase or decrease load on the web server? (Let’s revisit later.)

Screen Scraping

- A computer program visits a website in the same manner as a human at the keyboard would.
- The website thinks that the computer program is a human.

HTML Parsing

- Computer program request a URL from the website
- Computer program reads and parses the HTML returned by website
- Issues
  - Program may not cover all possible variations of the HTML
  - HTML could change at any time without prior notice

Scripts Have to Be Run

- Suppose a website has scripts (such as JavaScript) in its HTML.
- We cannot simply parse the script; we must run the script to render the HTML that we can parse.
- We must run a web browser emulator to render the entire webpage in HTML and then parse the HTML.
- These are the same issues we previously mentioned with parsing HTML.

Data in Images

- Suppose a website presents text data in an image
- Might be doing this to block screen scrapers
- Block sensitive data like email addresses, phone numbers, etc.
- Image processing with OCR (optical character recognition)
- OCR can be very error-prone

Websites Dislike Screen Scrapers

- Computer programs can obviously surf a website much faster than a human.
- Screen scraping puts a huge performance hit on website.
- Impacts other human users.
- Websites dislike screen scrapers and try to block them.

Blocking Screen Scrapers

- Captcha: distinguish human from computer
- Change HMTL frequently
- Put data in images that only humans can read
- If login is required, block users who are issuing numerous requests, more than a typical human does
- If login is not required:
  - Use the IP address to record how many transactions from that IP address
  - Issue: IP addresses are often shared

Legal Issues

- Websites typically have terms of service that forbid screen scraping
- Subject yourself to possible lawsuits
- Subject yourself to possible criminal prosecution

Revisit Open Question

Does adding an API increase or decrease load on the web server?

- Screen scraping is more intensive than an API.
- Having an API might encourage more activity.

The End

Screen Scraping Web Pages

Business Cases

Legal

- Recall that screen scraping:
  - Is against terms of service for most websites
  - Can subject you to lawsuits
  - Can subject you to criminal prosecution
- We will only cover legal business cases.

Old Nonprofit Website

- Does not have an API
- Retrofit would require too much expense and risk stability
- We reach out to the website owner and ask permission to screen scrape
- Since they are a nonprofit and we are using the data for the common good, they grant permission and give us throttle requirements and off-hours window
- We write the screen scraping with throttling and off-hours window

Old For-Profit Website

- Same situation but for profit
- We meet with them and arrange a monthly fee, throttle requirements, and off-hours window

Website We Own

- We own a website
- Probably internal to our company, not external to public
- We need data, but no API
- We can screen scape our own website without permission because we own it

The End

Downloading Files From Web Servers

Concepts

APIs Intensive

- Suppose we want 10,000, 100,000, or even one million data points.
- We would have to make a lot of API calls.
- APIs can be very intensive if someone wants a large amount of data.

Downloads

For data that is commonly requested, it is often more efficient for a website to provide the data in download file(s).

Hybrid

- A specific user wants a customized dataset
- Probably nobody or few others would want the customized dataset
- Solution:
  - User issues an API call to initiate creation of a custom dataset
  - User issues an API call to check if the dataset is created yet
  - Once created, user downloads the dataset
  - Best of both worlds

Downloads Use HTTP(S)

- Downloads typically use HTTP or HTTPS
- GET request if it is a common dataset
- POST request if it is a custom dataset
  - Send JSON with custom parameters
- HTTP response
  - Content type header
  - Message body
    - If text, UTF-8 encoded
    - If binary, base64 encoded and appears as UTF-8 text
- Easy to write a program to download a file

Downloading Text Files

- UTF-8 encoding
- Easy steps
  - Open a file
  - Write the message body to the file
  - Close the file

Downloading Binary Files

- Images, audio, video, Excel, zip, etc.
- Steps
  - Open a file
  - Decode the message body from base64 encoding to binary
  - Write the decoded binary to the file
  - Close the file

Downloading Zip Files

- Other compressed files have the same issues
- Zip files are downloaded the same as binary files
- Issue when we unzip
- Web servers typically translate text files to UTF-8 for us
- Text files archived inside a zip file are not translated to UTF-8
- Need to translate them ourselves
- Major issue
  - OS-specific code pages for Windows and Mac

The End

Downloading Files From Web Servers

Business Cases

Stock Data

- Stock brokerage company wants to generate interest in stock trading
- They provide a free download of daily closing stock prices
- Single file each day
- File is placed in edge servers in CDN
- Satisfies most users with little to no impact on web server
- Using an API would put huge strain on their web servers

Airline Schedule and Fares

- Numerous travel websites compare airline schedules and fares.
- Some airlines view it as more revenue channels.
  - Provide downloads of their schedules and fares as easy way to keep API and screen scraping traffic from their websites
- Other airlines view it as a way to lose business to competitors.
  - No download, no API, block screen scrapers
  - Legal action against websites which show their fares and schedules

Government Agencies

- Government agencies have some legal obligations to make data available to public
- Download files are most cost effective
- Agency websites can be very low-end and poorly designed
  - Buddy deals on government contracts to build websites
  - Often lack any technical expertise
  - No CDN, no scale up
  - Etc.
- Consolidating all datasets to data.gov
  - Hopefully better platform, CDN, etc.

Nonprofits

- Nonprofits typically share their data for free.
- Downloads are much more cost effective than building out and serving an API.
- However, nonprofit budgets can be tight, so even CDN might break the budget.
- CDNs are getting cheaper, including newer pricing models that includes egress.
- Cloud vendor public datasets provide another option.

University Data

- Universities have tons of research data.
- Professors/researchers usually want to share their data to advance humanity.
- Sometimes, government grants require data to be shared.
- The same issues exist as with other nonprofits.

Sneakernet

- Sneakernet is slang for moving files using physical drives such as jump drives or external hard drives instead of a computer network.
- Sneakernet gets less and less common as CDN prices go down.
- Suppose we need TiBs of data: It might make sense to ship an external hard drive back and forth, especially when budgets are tight.

The End