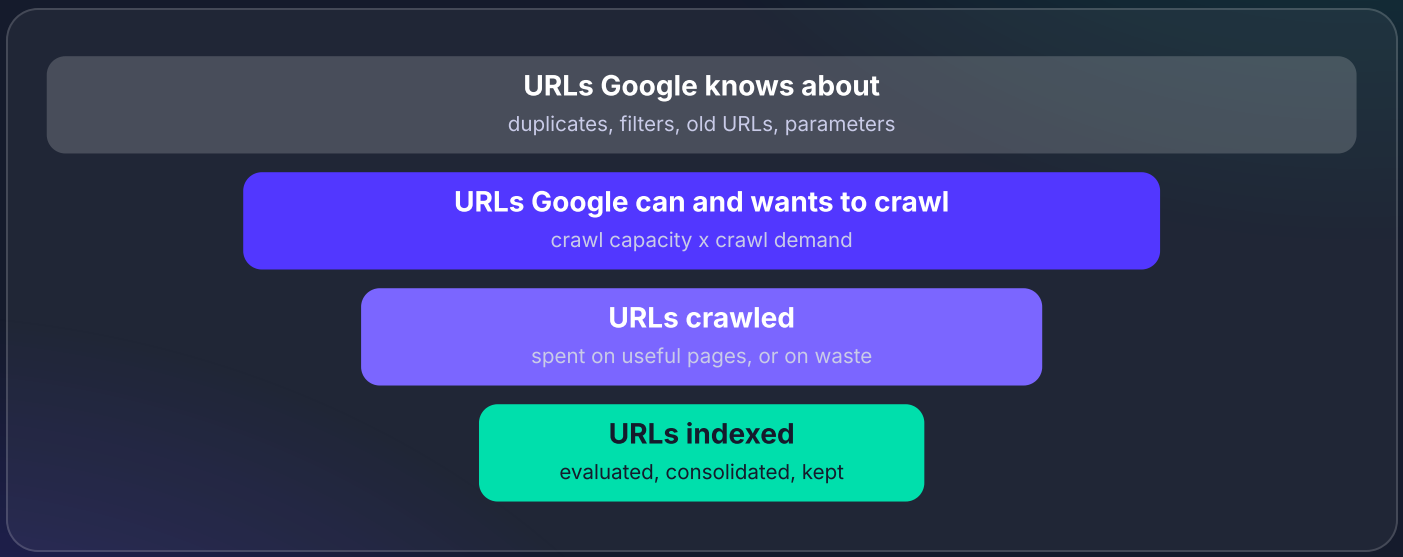
Crawl Budget For Search and AI Crawlers

How Google decides how much of your site to crawl, what wastes that crawling, and how to point Googlebot and AI crawlers at the pages that matter.

Guide · Technical SEO and AI crawlers

FROM KNOWN URLS TO INDEXED PAGES



2	3	1
parts: capacity and demand	demand factors you influence	budget per hostname

Ameneh Saeednia
Senior SEO and Generative Search Strategist

Start here

Does crawl budget matter for your site?

Google is direct about this: if your site doesn't have many pages that change quickly, or new pages are crawled **the same day** they're published, **you don't need to worry about crawl budget**. Keep your sitemap current and check the Page Indexing report. [1]

Google's guide is mainly for	Rough size	Change rate
Large sites	1 million+ unique pages	Content changes about **weekly**
Medium or larger sites	10,000+ unique pages	Content changes **daily**
Sites with indexing gaps	Any size	A large share of URLs in **Discovered, currently not indexed**

Signs crawling is your problem

New pages take **days or weeks** to be crawled.
Many URLs sit in **Discovered, currently not indexed**.
Crawl Stats show most requests going to **parameters, filters or redirects**.
URL Inspection reports **Hostload exceeded**. [1]

Signs it isn't

New pages are crawled within a day.
Pages are **Crawled, currently not indexed**. That's a **quality or duplication** issue, not a crawl issue.
You have a few thousand pages that rarely change.
Your problem is rankings, not discovery.

The numbers are rough

Google says its thresholds are **a rough estimate, not exact thresholds**. [1] A 20,000-URL site with endless filter URLs can have a crawl problem. A 2-million-page site with clean URLs may not.

How it works

Capacity plus demand

Google can't crawl the whole web, so it limits how much time it spends on each site. That allocation is your **crawl budget**, and it's set **per hostname**: www.example.com and blog.example.com have **separate budgets**. [1]

Crawl capacity limit
How much Google can crawl without overloading your server. Rises with fast, healthy responses. Falls with slowness, 5xx and 429. [1]

+

Crawl demand
How much Google wants to crawl. Driven by perceived inventory, popularity and staleness. [1]

=

Crawl budget
The set of URLs Google **can and wants** to crawl on one hostname. [1]

Crawl capacity limit (hostload)

Every site starts with the **same conservative limit**. Google raises it over time if there is demand and the site stays healthy. [1]
Stable or improving **response times and TTFB** raise it. **5xx errors, 429s and slowdowns** lower it. [1]
The capacity limit is **shared across all Google crawlers**. High demand from one, for example AdsBot or Shopping, can leave less for others. [1]

Crawl demand

Perceived inventory: all the URLs Google knows. Google calls this the factor **you can positively control the most**. [1]
Popularity: more popular URLs are crawled more often to stay fresh. [1]
Staleness: Google recrawls often enough to catch changes. [1]
Site-wide events such as **site moves** temporarily raise demand. [1]

Crawled is not indexed

After crawling, each page is **evaluated, consolidated and assessed** before it gets into the index. [1] Better crawling gets more pages **looked at**. Only quality and uniqueness get them **kept**.

Part 1

What wastes crawling

If Google spends its time on URLs it shouldn't, it may **not explore the rest of your site**, or may not raise your budget. [1] These are the usual culprits.

Waste	What it looks like	Fix
Faceted navigation	`?color=red&size=s&sort=price` in endless combinations [2]	robots.txt disallow, or URL fragments (Part 3)
Duplicate URLs	http/https, www/non-www, trailing slashes, tracking parameters	Redirect to one version, canonical, clean internal links
Soft 404s	"No results" or empty pages returning 200	Return a real 404. Soft 404s **keep getting crawled** [1]
Redirect chains	A → B → C → D after several migrations	Redirect straight to the final URL. Chains **hurt crawling** [1]
Infinite spaces	Calendars, internal search results, endless pagination	Block in robots.txt, cap pagination
Removed pages still linked	Internal links to 404s and old URLs	Update links. Use 404 or 410 for removed pages [1]
Slow, heavy pages	High TTFB, large render cost	Faster server, caching, lighter pages (Part 4)
Thin archives	Tag and paginated archives with little unique content	Keep as simple crawl paths; noindex or consolidate thin tags

Where to look first

Open **Crawl Stats** in Search Console and sort requests by URL pattern in your logs. On most sites, **two or three URL patterns** account for most of the waste.

Part 2

Choosing the right control

Each tool does a different job. Google's crawl budget guide is specific about which one **saves crawling** and which doesn't. [1]

Tool	Stops crawling?	Stops indexing?	Use it for
robots.txt disallow	**Yes**	Usually, but a blocked URL can still be indexed without content if linked	Filters, sorts, internal search, infinite spaces you never want crawled
noindex	**No**. Google still requests the page, then drops it [1]	Yes	Pages users need that shouldn't be in search
rel=canonical	Partly, over time [2]	Consolidates to the canonical	Near-duplicates that must stay live
404 / 410	**Mostly**. A strong signal not to crawl again [1]	Yes, removed from index [4]	Pages permanently removed
301 redirect	Old URL is crawled less over time	Target becomes canonical	Moved or merged pages
rel=nofollow on links	Weak and less effective long term [2]	No	Not a crawl budget tool

Two common mistakes

Using **noindex** to save crawl budget. It doesn't: Google still has to fetch the page to see the tag. [1]
Using **robots.txt** to temporarily push crawling elsewhere. Google **won't shift** freed budget to other pages unless you're already at your capacity limit. [1]

Blocked vs removed

Blocked URLs stay in the crawl queue **much longer** and are recrawled when the block is lifted. [1]
404/410 URLs are a strong signal not to come back. [1]
So: **remove** what's gone, **block** what should never be crawled.

Part 3

Faceted navigation and parameters

Filters are useful for shoppers and travellers, but parameter-based filters create **near-infinite URL spaces**. Google says this causes **overcrawling** and **slower discovery** of new, useful pages. [2]

If filter URLs don't need to be in search

Disallow them in robots.txt. Allow the individual item pages plus one listing page without filters. [2]
Or use **URL fragments** (`#color=red`) for filters. Google generally doesn't crawl fragments, so they have **no impact** on crawling. [2]

If some filter URLs should rank

Pick the **few combinations with real demand** (for example *private tours in Rome for families*) and make them clean, static URLs.
Keep a **consistent parameter order** and return **404** for empty result sets.
Link to them like normal pages; block the rest.

Example robots.txt, adapted from Google [2]

```
user-agent: Googlebot
disallow: /*?*sort=
disallow: /*?*price=
disallow: /*?*group_size=
allow: /*?category=all$
```

Test before you ship

A wrong wildcard can block **real pages**. Test rules against a sample of URLs, ship them, and watch **Crawl Stats** and the Page Indexing report for two to four weeks.

Part 4

Server health, speed and caching

Capacity is earned. **Fast, stable responses** raise your limit. Errors and slowdowns lower it. [1]

Signal Googlebot sees	Effect	What to do
Stable or faster response times and TTFB	Capacity limit **goes up** [1]	Server-side caching, CDN, efficient queries
Slower responses, rising latency	Capacity **goes down** [1]	Monitor TTFB for Googlebot in logs
5xx errors	Crawling slows. Indexed URLs are kept for a while, but **eventually dropped** [4]	Fix at source; alert on 5xx spikes
429 too many requests	Treated as a **server error**; crawling slows [4]	Use only in emergencies
304 Not Modified	Google reuses its cached copy, saving your bandwidth [1]	Support If-Modified-Since / ETag
Hostload exceeded in URL Inspection	Your server is the bottleneck [1]	Add server resources, if it makes business sense [1]

Make pages cheap to fetch and render

If Google can **load and render** faster, it may read **more of your site**. [1]
Trim render-blocking scripts and heavy third-party tags.
Serve key content in the **initial HTML** where possible.

Don't block resources

Blocking CSS or JS in robots.txt to "save budget" can stop Google from **rendering** your pages properly. Block **URLs**, not the resources pages need.

Part 5

Raising demand: be worth crawling

Google gives two ways to get **more** crawl budget: more server resources, and **better content for the product you target**, including popularity, user value, uniqueness and serving capacity. [1]

What Google says drives demand

Popularity: popular URLs are recrawled more often. [1]
Staleness: pages that change get revisited. [1]
Sitemaps: Google reads them regularly. Include everything you want crawled and use `<lastmod>` for updated content. [1]
Quality and uniqueness for the Google product you're targeting. [1]

What Kopp's patent summaries add

A crawler **scheduler** scores each URL by how often its content **changes**, and compares the score to a threshold to decide what to crawl. [6]
Another scheduler uses **sitemap metadata**: last modification date, change frequency, title, authority and priority. [7]
A crawl algorithm assigns pages a **crawl value** from bandwidth and change indicators, using machine learning, and recrawls when the value passes a threshold. [8]

Lever	Practical move
Honest lastmod	Update `<lastmod>` only when content really changes. Fake dates teach the scheduler to ignore you
Internal links	Link new pages from hubs and the homepage on day one (see the Internal Linking guide)
Popularity	Earn links and mentions to key sections (see the Backlinks guide)
Prune waste	Remove or merge low-value URLs so demand isn't spread thin (see the Content Refresh guide)

Patents are not proof

Patent summaries show **possible** mechanisms, not confirmed ranking or crawling behaviour. They line up with Google's own guidance here: **change frequency, sitemaps and importance** drive recrawling.

Part 6

AI crawlers

AI products crawl too, and each vendor uses **separate user agents for search and for training**. Decide per bot, on purpose.

Bot	Used for	If you block it
Googlebot	Google Search, AI Overviews and AI Mode use the Search index	You leave Google Search
Google-Extended (token only)	Whether Google-crawled content can train future **Gemini** models and ground some Gemini features [3]	No effect on Search inclusion or ranking. It has **no separate user agent**, so it adds no crawling [3]
OAI-SearchBot	Surfacing sites in **ChatGPT search** [5]	Not shown in ChatGPT search answers, except as navigational links [5]
GPTBot	Content that may be used to **train** OpenAI models [5]	Signals content shouldn't be used for training [5]
ChatGPT-User	Visits triggered by a **user request** [5]	robots.txt rules **may not apply** to user-initiated visits [5]

Practical stance for most brands

Allow search bots (Googlebot, OAI-SearchBot and other vendors' search agents). They decide whether you can be **cited**.
Make a **business decision** on training bots. Blocking them doesn't remove you from AI search.
If both OpenAI bots are allowed, OpenAI says it may use **one crawl for both** to avoid duplication. [5]

Watch the load

Check logs for **AI user agents** and their share of requests.
Verify bots against **published IP ranges** before trusting the user agent. [5]
Google plans to support the **AIPREF** training-preference signal in early 2027. [3] Expect these controls to keep changing.

Part 7

Diagnosing crawl problems

1

Crawl Stats report

Search Console, Settings. Total requests, response times, **response codes, file types, purpose** (discovery vs refresh) and host status over 90 days.

2

Page Indexing report

Volume of **Discovered, currently not indexed**, soft 404s and redirect errors. Google recommends checking it regularly. [1]

3

URL Inspection

Last crawl date for key URLs, and capacity problems such as **Hostload exceeded**. [1]

4

Server logs

The only full view: every bot hit by URL pattern, status and response time, including **AI crawlers**.

Question	Healthy answer	Where
How fast are new pages crawled?	Within a day or two	URL Inspection, logs
Share of Googlebot hits on parameter / filter URLs	Small	Logs
Share of hits returning 3xx, 4xx, 5xx	Low and stable	Crawl Stats
Average response time	Stable or falling	Crawl Stats
Discovered, currently not indexed	Falling after fixes	Page Indexing
Priority sections crawled in the last 30 days	All of them	Logs grouped by folder

Work in this order

1. Fix errors and speed. **2.** Remove waste from the inventory. **3.** Strengthen internal links and sitemaps to priority pages. **4.** Re-measure after four to six weeks.

Checklist and sources

Crawl budget checklist

Inventory

[ ] Know whether crawl budget **matters** for your site
[ ] Facets and sorts **blocked** or fragment-based
[ ] One version of every URL; **no duplicate** hosts
[ ] **Soft 404s** fixed to real 404s
[ ] **No redirect chains**; internal links to final URLs
[ ] Removed pages return **404 or 410**

Health and demand

[ ] Stable **TTFB**; 5xx and 429 monitored
[ ] **304** caching supported
[ ] Sitemaps complete, with **honest lastmod**
[ ] New pages linked from **hubs** on day one
[ ] **AI bot** policy decided per user agent
[ ] Crawl Stats and logs reviewed **monthly**

Sources

01 **Optimize your crawl budget. Google Crawling Infrastructure**
developers.google.com/crawling/docs/crawl-budget

02 **Managing crawling of faceted navigation URLs. Google**
developers.google.com/crawling/docs/faceted-navigation

03 **Google's common crawlers (Google-Extended). Google**
developers.google.com/crawling/docs/crawlers-fetchers/google-common-crawlers

04 **How HTTP status codes affect Google's crawlers. Google**
developers.google.com/crawling/docs/troubleshooting/http-status-codes

05 **Overview of OpenAI crawlers. OpenAI**
developers.openai.com/api/docs/bots

06 **Scheduler for search engine crawler. Kopp patent summary**
kopp-online-marketing.com/patents-papers/scheduler-for-search-engine-crawler

07 **Web crawler scheduler that utilizes sitemaps. Kopp patent summary**
kopp-online-marketing.com/patents-papers/web-crawler-scheduler-that-utilizes-sitemaps-from-websites

08 **Crawl algorithm. Kopp patent summary**
kopp-online-marketing.com/patents-papers/crawl-algorithm

Guide by **Ameneh Saeednia**, seogirl.ca, drawing on Google's crawling documentation, OpenAI's crawler documentation and **Olaf Kopp**'s public patent summaries, prepared with AI assistance. Bot policies change often; check vendor pages before acting.