

lab

April 3, 2026

1 Lab 14 — Cost, Caching & Rate Limits

Explore prompt caching, cost estimation, and the Batch API.

Exercises: 1. Prompt caching with a long system prompt 2. Token budget checker 3. Batch API with 20 tasks

```
[ ]: # Setup
import json
import time
from dotenv import load_dotenv
import anthropic

load_dotenv()
client = anthropic.Anthropic()
MODEL = "claude-sonnet-4-20250514"
print("Setup complete.")
```

```
[ ]: # Long system prompt for caching exercises
# This generates a ~2000 token system prompt to test caching

SYSTEM_PROMPT = """You are an expert technical writing assistant specialising
↳in software documentation. Your role is to help developers write clear,
↳accurate, and comprehensive documentation for their projects.

## Core Principles

1. Clarity First: Every sentence should convey exactly one idea. Avoid
↳ambiguity. If a term could be interpreted multiple ways, define it
↳explicitly.

2. Audience Awareness: Always consider who will read the documentation. API
↳references are for experienced developers; tutorials are for beginners.
↳Adjust vocabulary, detail level, and examples accordingly.

3. Structure: Use consistent heading hierarchies. Start with a brief
↳overview, then dive into details. Group related concepts together. Provide a
↳table of contents for documents longer than 3 sections.
```

4. ****Code Examples****: Every concept should include a working code example.□
↳ Examples should be:
- Complete (can be copied and run without modification)
 - Minimal (no unnecessary code that distracts from the concept)
 - Annotated (comments explaining non-obvious lines)
 - Tested (verified to work with the current version)
5. ****Error Documentation****: Document common errors and their solutions.□
↳ Developers spend more time debugging than writing new code, so good error□
↳ documentation is extremely valuable.

Writing Style Guide

- Use active voice: "The function returns a list" not "A list is returned by□
↳ the function"
- Use present tense: "This method creates a new instance" not "This method will□
↳ create a new instance"
- Use second person for tutorials: "You can configure the timeout by..."
- Use imperative mood for instructions: "Set the environment variable" not "You□
↳ should set the environment variable"
- Keep paragraphs short (3-5 sentences maximum)
- Use bullet points for lists of 3 or more items
- Use numbered lists for sequential steps
- Use tables for comparing features or options

Documentation Types

API Reference

- Document every public function, class, and method
- Include: signature, parameters (with types), return value, exceptions,□
↳ examples
- Use consistent formatting across all entries

Tutorials

- Start with what the reader will build/learn
- List prerequisites upfront
- Provide step-by-step instructions
- Include checkpoints where the reader can verify progress
- End with next steps and further reading

Conceptual Guides

- Explain the "why" before the "how"
- Use analogies to connect new concepts to familiar ones
- Include diagrams where they add clarity
- Link to API reference for implementation details

```

### Changelog
- Group changes by category (features, fixes, breaking changes)
- Use semantic versioning
- Include migration guides for breaking changes

## Quality Checklist
Before finalising any documentation, verify:
- All code examples run without errors
- Links point to valid destinations
- Technical terms are used consistently
- No placeholder text remains (TODO, TBD, FIXME)
- Formatting renders correctly in the target platform
- Screenshots are current and annotated if necessary

Remember: good documentation is an investment that pays dividends in reduced_
↳support burden, faster onboarding, and higher developer satisfaction. Treat_
↳documentation with the same rigour as production code."""

print(f"System prompt length: {len(SYSTEM_PROMPT)} characters")

```

1.1 Exercise 1 — Prompt Caching

Send cached and uncached requests. Compare the usage statistics and costs.

```

[ ]: # TODO: Send 3 requests:
# 1. With cache_control on the system prompt (cache write)
# 2. Same request again (cache read)
# 3. Without cache_control (no caching)
#
# For each, print response.usage and calculate the cost.
# Compare cache_creation_input_tokens vs cache_read_input_tokens vs_
↳input_tokens.

raise NotImplementedError("Exercise 1: implement prompt caching comparison")

```

1.2 Exercise 2 — Token Budget Checker

Build a function that estimates cost before calling the API.

```

[ ]: # TODO: Implement estimate_cost(model, input_tokens, max_output_tokens) -> float
# TODO: Implement check_budget(model, messages, system_prompt, max_cost_usd) ->_
↳bool
#
# Pricing (per million tokens):

```

```

# Sonnet 4: input=$3, output=$15
# Haiku 3.5: input=$0.80, output=$4
#
# Use client.messages.count_tokens() to get actual token count.

raise NotImplementedError("Exercise 2: implement budget checker")

```

1.3 Exercise 3 — Batch API

Submit 20 classification tasks via the Batch API.

```

[ ]: # TODO: Create 20 text classification tasks
# TODO: Submit via client.messages.batches.create()
# TODO: Poll for completion
# TODO: Process results and calculate cost savings vs real-time
#
# Sample texts for classification:
SAMPLE_TEXTS = [
    "The new iPhone features a faster processor and improved camera system.",
    "Scientists discovered a new species of deep-sea fish near hydrothermal_
↪vents.",
    "The stock market rallied today on positive earnings reports from tech_
↪companies.",
    "A new study links regular exercise to improved cognitive function in older_
↪adults.",
    "The latest Python release includes pattern matching and improved error_
↪messages.",
    "Climate change is causing more frequent and severe weather events globally.
↪",
    "The restaurant received a Michelin star for its innovative fusion cuisine.
↪",
    "SpaceX successfully launched another batch of Starlink satellites into_
↪orbit.",
    "A major cybersecurity breach exposed millions of customer records.",
    "The new electric vehicle can travel 400 miles on a single charge.",
    "Researchers developed a more efficient solar cell using perovskite_
↪materials.",
    "The housing market shows signs of cooling after two years of rapid growth.
↪",
    "A breakthrough in quantum computing achieved a new record for qubit_
↪stability.",
    "The film received critical acclaim at the Cannes Film Festival.",
    "New regulations require companies to disclose AI usage in customer_
↪interactions.",

```

```
    "The World Health Organization issued guidelines for a new respiratory_
↪virus.",
    "An archaeological dig uncovered a 3000-year-old temple in southern Egypt.",
    "The central bank raised interest rates to combat persistent inflation.",
    "A new programming language aims to combine Rust safety with Python_
↪simplicity.",
    "The Olympics committee announced the addition of three new sports for 2028.
↪",
]

raise NotImplementedError("Exercise 3: implement batch API processing")
```