Website Data Collection Optimization Report

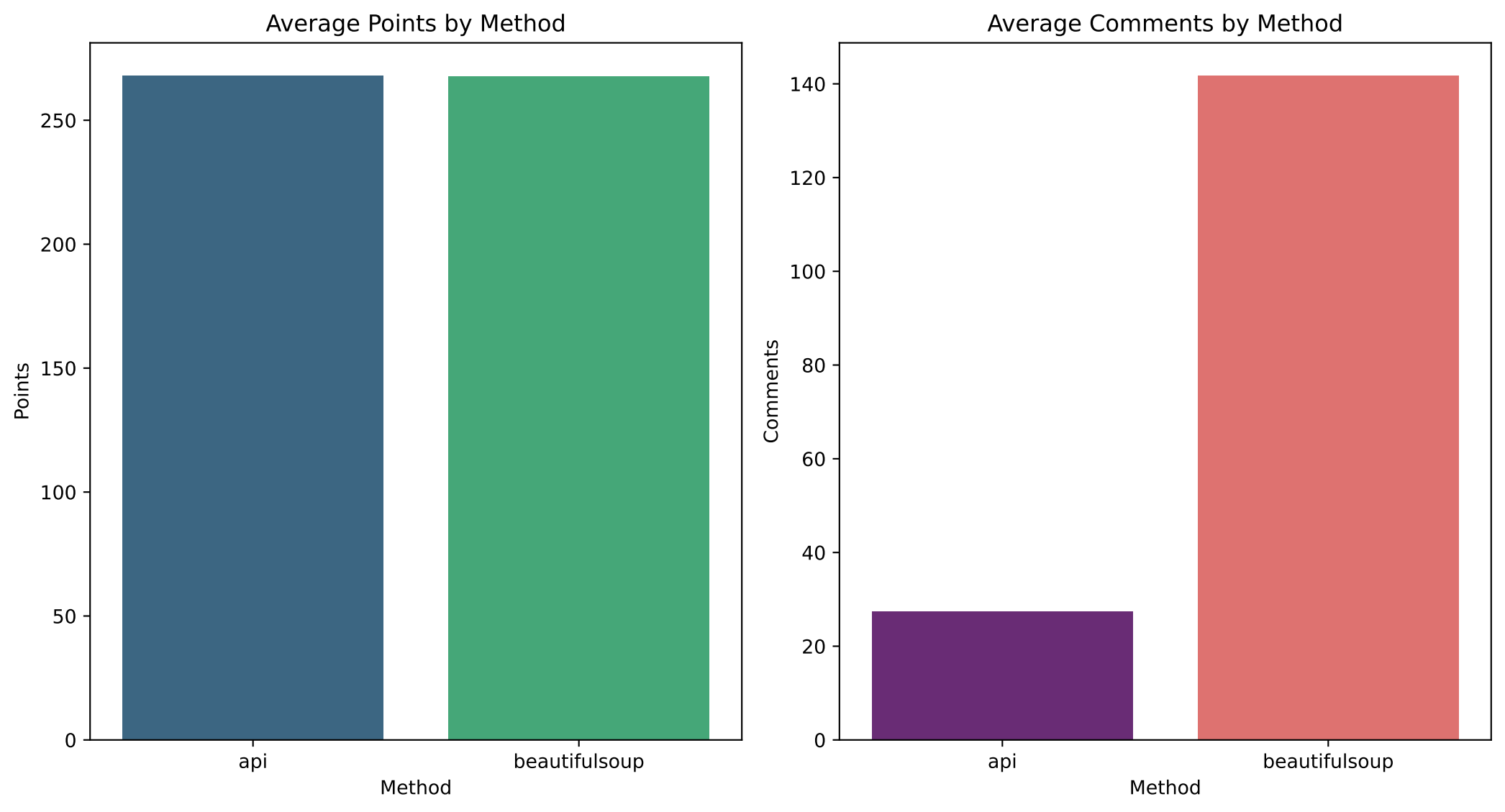
Target Site: https://news.ycombinator.com
Data Collected: Titles, points, comment counts, authors, first comment

Key Highlights
• Total requests issued: 90
• Aggregate bandwidth consumed: 6721.0 KB
• Fastest method: Api
• Selenium captured full rendered context (comment text) at the cost of higher latency.

Objective:
Compare scraping efficiency, observe network behaviour, and recommend the most resilient workflow.

Performance & Network Summary

method	total_time_s	total_requests	total_bytes	avg_latency_ms
beautifulsoup	51.77	30	6677.4 KB	1114.6
api	5.96	60	43.6 KB	226.9


Average Points by Method
Points
0
50
100
150
200
250
api
beautifulsoup
Method
Average Comments by Method
Comments
0
20
40
60
80
100
120
140
api
beautifulsoup
Method

Headline Keyword Signals

```
Positive Indicators (more likely to rank high):
talk                  (0.92)
android open          (0.75)
android               (0.75)
wanted                (0.70)
boring wanted         (0.70)
boring                (0.70)
open                  (0.55)
flight                (0.46)
flight problem        (0.46)
pro                   (0.46)

Negative Indicators (less likely to rank high):
saw                   (-0.28)
default               (-0.24)
project               (-0.24)
project shadowglass   (-0.24)
decline               (-0.24)
decline deviance      (-0.24)
shadowglass           (-0.24)
https                 (-0.24)
https default         (-0.24)
deviance              (-0.24)
```

Recommended Strategy & Hardening Checklist

Optimal Workflow

- Use the API collector for frequent polling (fastest: Api).
- Augment with the BeautifulSoup scraper to capture rendered comment context.
- Schedule Selenium runs hourly to validate UI changes and keep parsing selectors fresh.

Hardening Steps

- Enforce rate limiting via the configurable `throttle_s` arguments.
- Restrict outbound ports with `ufw` during tests to ensure graceful degradation.
- Capture traffic with `tcpdump` and archive `.pcap` files in `network/` for audits.
- Route high-volume runs through a proxy or VPN and refresh credentials securely.