

warning "Backend update (October 2026)" GitHub Models inference was retired on July 30, 2026. Any references below are retained only for historical context while this lab is migrated. Use Lab 013 ? Free LLM backends to configure Foundry Local or Ollama before continuing.

Lab 038: AI Cost Optimization

Level: L300 Path: ? Pro Code Time: ~45 min ? Cost: Azure Free Tier ? Most strategies are free to implement

What You'll Learn

Understand AI API cost drivers (tokens, model choice, request frequency) Apply 5 key strategies to reduce costs: caching, batching, model routing, prompt compression, and streaming Implement semantic caching to avoid duplicate LLM calls Build a model router that uses cheap models for simple queries and expensive models for complex ones Monitor and alert on cost with Azure Monitor

Introduction

A production AI agent can easily cost \$50?500/month even with modest usage if not optimized. The good news: most cost reduction strategies also improve latency and user experience.

The key cost levers:

Factor | Impact | Optimization | Model choice | 10?50× cost difference | Route to cheaper model when possible | Token count | Linear cost | Compress prompts, trim history | Cache hits | Eliminate cost entirely | Semantic cache for repeated queries | Request volume | Linear cost | Batch, debounce, deduplicate | Output length | Linear cost | Use max_tokens, structured output |

Part 1: Measure Your Baseline

Before optimizing, measure what you're spending:

[diagram or code sample]

Part 2: Strategy 1 ? Semantic Cache

A semantic cache stores LLM responses and reuses them for semantically similar queries, even when the wording differs:

[diagram or code sample]

Part 3: Strategy 2 ? Model Router

Use cheap models for simple queries, expensive models for complex ones:

[diagram or code sample]

Part 4: Strategy 3 ? Prompt Compression

Trim conversation history to keep only the most recent + summary:

[diagram or code sample]

Part 5: Cost Monitoring with Azure Monitor

In production, track costs with Azure Monitor:

[diagram or code sample]

In the Azure Portal, create an alert:

[diagram or code sample]

? Knowledge Check

question "1. What is semantic caching, and how does it differ from exact-match caching?"

Exact-match caching returns cached results only when the input is byte-for-byte identical.

Semantic caching uses embeddings to find cached responses for semantically similar inputs ?

"Show me tents" and "What tents do you have?" get the same cached response even though the wording differs. Semantic caching is far more effective for conversational AI.

question "2. When should you NOT use a cheap model in a model router?" Use a premium model

for: multi-step reasoning (chain-of-thought), comparisons between complex options, personalized recommendations that require weighing many factors, tasks where errors are costly (medical, financial, legal). Use cheap models for: lookups, listing, formatting, simple Q&A where accuracy tolerance is higher.

question "3. What are the risks of aggressively compressing prompt history?" The agent may lose context it needs: a user mentioned an allergy 5 turns ago and the compressed summary missed it. A poorly summarized history can cause contradictory responses in long conversations. Always keep the most recent N turns verbatim; only summarize older turns. Test with scenarios that require long-range context.

Summary

Strategy | Potential Savings | Complexity | Semantic caching | 40?80% of API calls | Medium
| Model routing | 10?40× cost reduction | Low | Prompt compression | 20?60% token reduction
| Low | Batching requests | 10?30% (latency trade-off) | Medium | Output limiting
(max_tokens) | 5?20% | Very Low |

Next Steps

Observe what you optimized: ? Lab 033 ? Agent Observability Evaluate quality after
optimization: ? Lab 035 ? Agent Evaluation Full production deployment: ? Lab 028 ? Deploy
MCP to Azure

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

[Download generated PDF](#) [Download generated ZIP](#)

Supporting file: CHEATSHEET.md

Supporting files: Lab 038: AI Cost Optimization

Start with the documented local or free path. Record assumptions and validation evidence.

Keep hosted services optional unless the lab marks them required.

Supporting file: README.md

Lab 038: AI Cost Optimization

Use these files with the lab guide and generated kit.