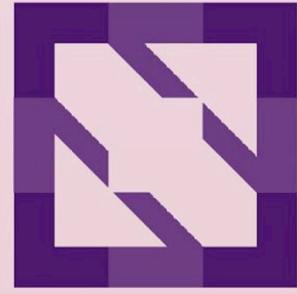


KubeCon



CloudNativeCon

North America 2023



KubeCon



CloudNativeCon

North America 2023

Kube of Thoughts

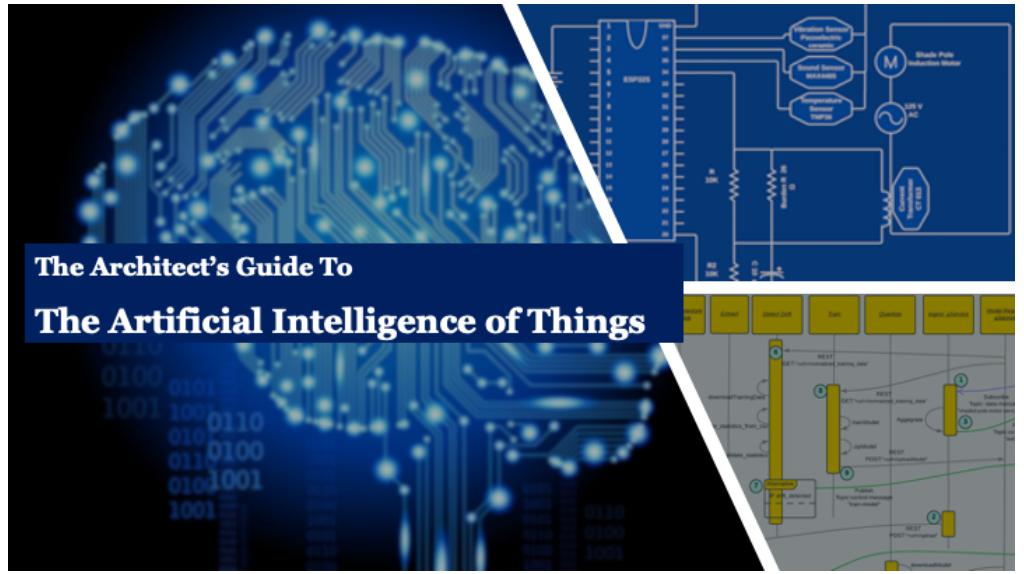
Scaling Generative AI Models With Kubernetes and Inference Decision Trees

Asheesh Goja

About Me



Principal Solutions Architect @ AWS



[Kubecon 2021 Talk](#)

Agenda

- Problem
- Tree of Thought Strategy (ToT)
- Complexity and efficiency challenges
- Software patterns and architecture to improve efficiency
- Using K8s to make ToT practical and efficient
- Overall system architecture
- Live Demo

A Math Game



KubeCon

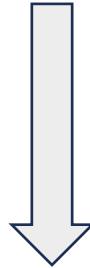


CloudNativeCon

North America 2023

1 1 11 11

Arrange using (+ - * /)



24

$$1 + 1 + 11 + 11 = 24$$

Behold the power of LLMs

Prompt : Arrange the following numbers using the arithmetic operators to obtain 24, use each number only once

1 1 11 11

ChatGPT PLUS

Make a content strategy
for a newsletter featuring free local wee...

Tell me a fun fact
about the Roman Empire

Send a message ➤

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT September 25 Version](#)

GPT4

Bard Experiment

Hello again

Tell me what's on your mind, or pick a suggestion. I have limitations and won't always get it right, but your feedback will help me improve.

Understand Create Explore

advancements in tech... taglines for my store give me a shell co

debug Python code product name ideas best beaches in...

how something works sales pitch meal planning

Enter a prompt here

Bard may display inaccurate or offensive information that doesn't represent Google's views. [Bard Privacy Notice](#)

Bard

A Untitled ✓ AG

Claude by ANTHROPIQUE

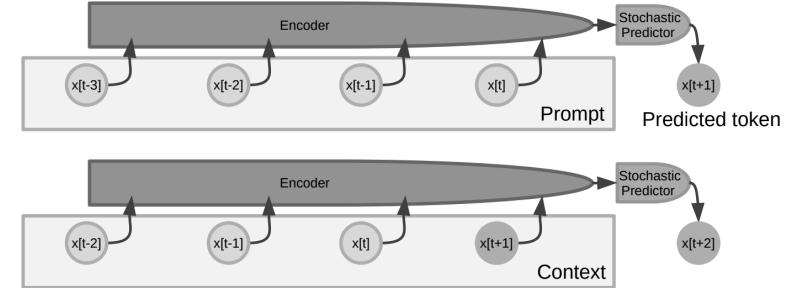
Message Claude...

☞ to send, shift + ☞ to add a new line, ⌘K to create a new chat

Claude2

Why is that ?

- Auto Regressive
 - Self attention based
 - LLMS such as GPT4, Claude or PaLM 2 are AR LLMS
 - Decoder-only Transformer Variants
- Sequential Processing
 - Work in a sequential manner, predicting one token at a time based on previous token
 - Difficult for them to solve problems that require multi-step reasoning.
 - They don't explore different continuations within a single thought process
- No Feedback Loop
 - They don't have a mechanism to self-check their work or iterate upon it
 - Do not explicitly perform logical correctness checks as they generate text
 - This amplifies small errors, leading to bigger mistakes as the solution progresses.
- No Backtracking
 - Not able to backtrack and correct their mistakes, reassess
 - Do not consider alternative solution paths due to lack of exploration
 - Unlike human problem-solving AR-LLMs lack of heuristic-guided solution search



$$(11 + (11 \div (1 + 1))) = 24$$

Here's the step-by-step breakdown:

1. $1 + 1 = 2$
2. $11 \div 2 = 5.5$
3. $11 + 5.5 = 16.5$

So this arrangement actually doesn't yield 24, and it seems like I made a mistake in my previous message. I apologize for the confusion.

Here's a correct arrangement:

$$(11 \div (1 - \frac{1}{11})) = 24$$

Here's the step-by-step breakdown:

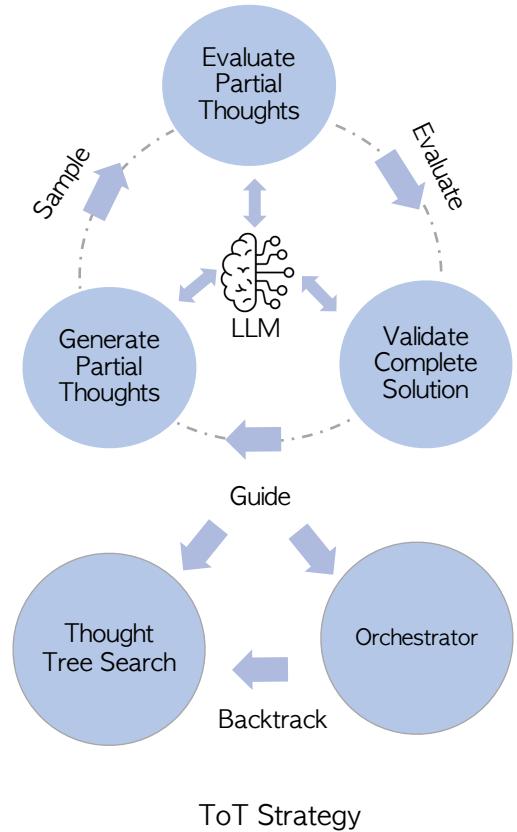
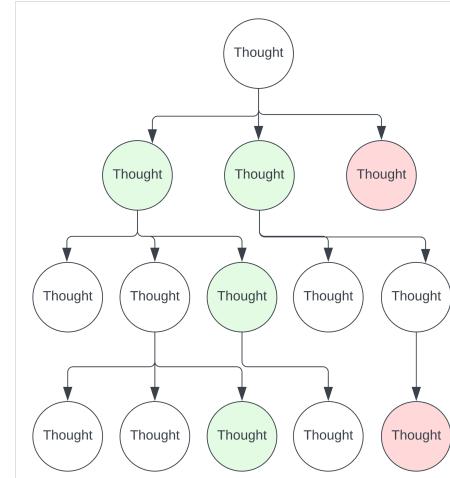
1. $\frac{1}{11} = 0.0909$
2. $1 - 0.0909 = 0.9091$
3. $(11 \div 0.9091) = 24$

Prompting Strategies

Strategy	Technique	Heuristics	Limitations	Advantages
Chain-of-thoughts (CoT)	Sequential Generation	Use intermediate steps leading to output.	Ambiguous decomposition of thoughts	Transparency and tractability in solving complex tasks.
CoT-SC (Self Consistency)	Ensemble Method	Multiple independent chains and selection of most frequent output.	No exploration of alternative paths within chains of thought	Considers multiple thought processes, potentially resolving ambiguity.
Tree of Thoughts (ToT)	Inference Decision Trees	Combinatorial exploration of multiple reasoning paths and partial solutions as a tree structure.	Needs sophisticated and efficient tree search exploration of vast combinatorial problem space.	Local exploration of different steps within a thought process Global exploration using backtracking and look ahead

Tree of Thoughts – Strategy

- Thought Representation – Tree
 - Reasoning represented as tree with nodes of partial solution thoughts.
 - Consider multiple diverse and independent solution paths
- Generate multiple partial thoughts – Sampling
 - From tree nodes explore diverse potential thoughts
 - Actively generate multiple possible thoughts at each step
 - Sample thoughts independently for diversity
- Evaluate each partial thought – Self Reflection
 - Gauge each state, self evaluate, compare partial solutions, choose best.
 - Evaluate validity of each thought, used to guide the tree search
- Validate the correctness of the proposed solution
 - Validates the correctness of a complete solution
- Tree Search Algorithm
 - Plan search trajectory based on the feedback from orchestrator
 - Determine optimal tree search algorithm based on the problem.
 - BFS
 - DFS
 - A* (Best-First Search + Dijkstra's algo)
- Orchestrator
 - Facilitates multi round conversations with LLM
 - Use backtracking and look ahead to make global choices
 - Ensures rich diversity potential thoughts explored



Solving Game of 24 using ToT

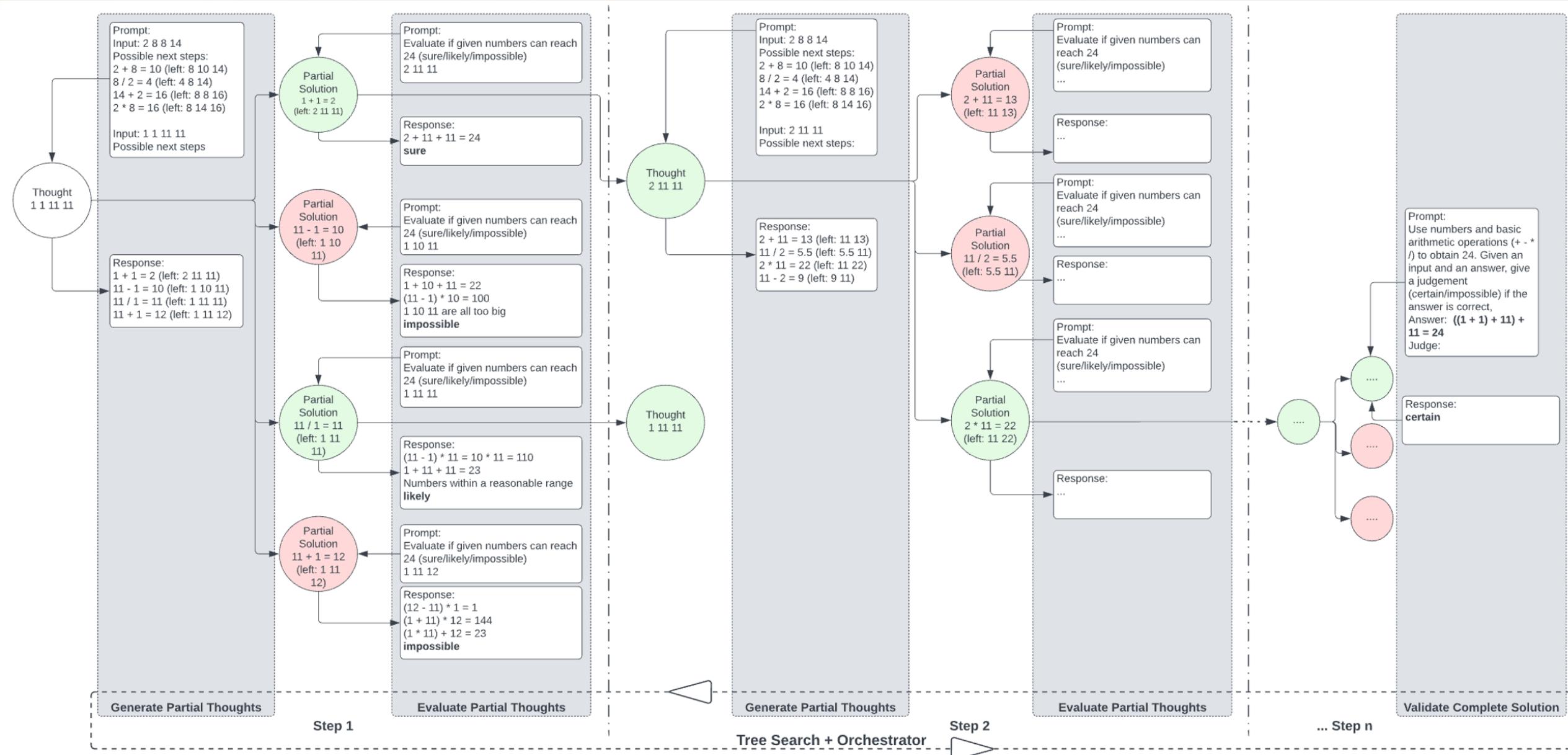


KubeCon



CloudNativeCon

North America 2023



Is this practical ?



KubeCon



CloudNativeCon

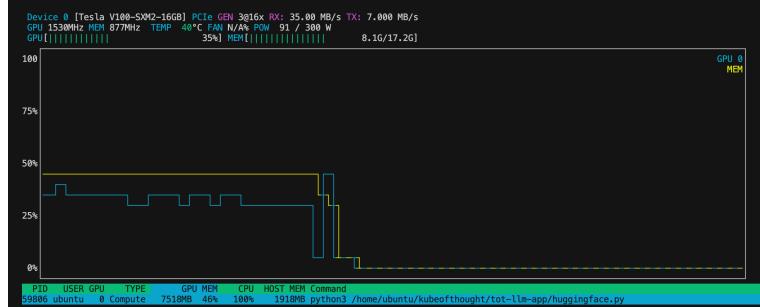
North America 2023

- Complexity
 - Time
 - Space
 - CPU memory
 - GPU Memory - CUDA out of memory
- Throughput
 - LLM API Rate Limits
 - LLM Token Limits
- Latency
- Cost to operate

Algorithm	Time Complexity	Space Complexity
DFS	$O(B^{D+1})$	$O(BD)$
BFS	$O(B^{D+1})$	$O(B^D)$
Best-First Search	$O(B^{D+1})$	$O(B^D)$
A*	$O(B^{D+1})$	$O(B^D)$

D is depth and B is average branching factor

RuntimeError: CUDA out of memory. Tried to allocate 200.00 MiB (GPU 0; 15.78 GiB total capacity; 14.56 GiB already allocated; 38.44 MiB free; 14.80 GiB reserved in total by PyTorch) If reserved memory is >> allocated memory try setting max_split_size_mb to avoid fragmentation. See documentation for Memory Management and PYTORCH_CUDA_ALLOC_CONF



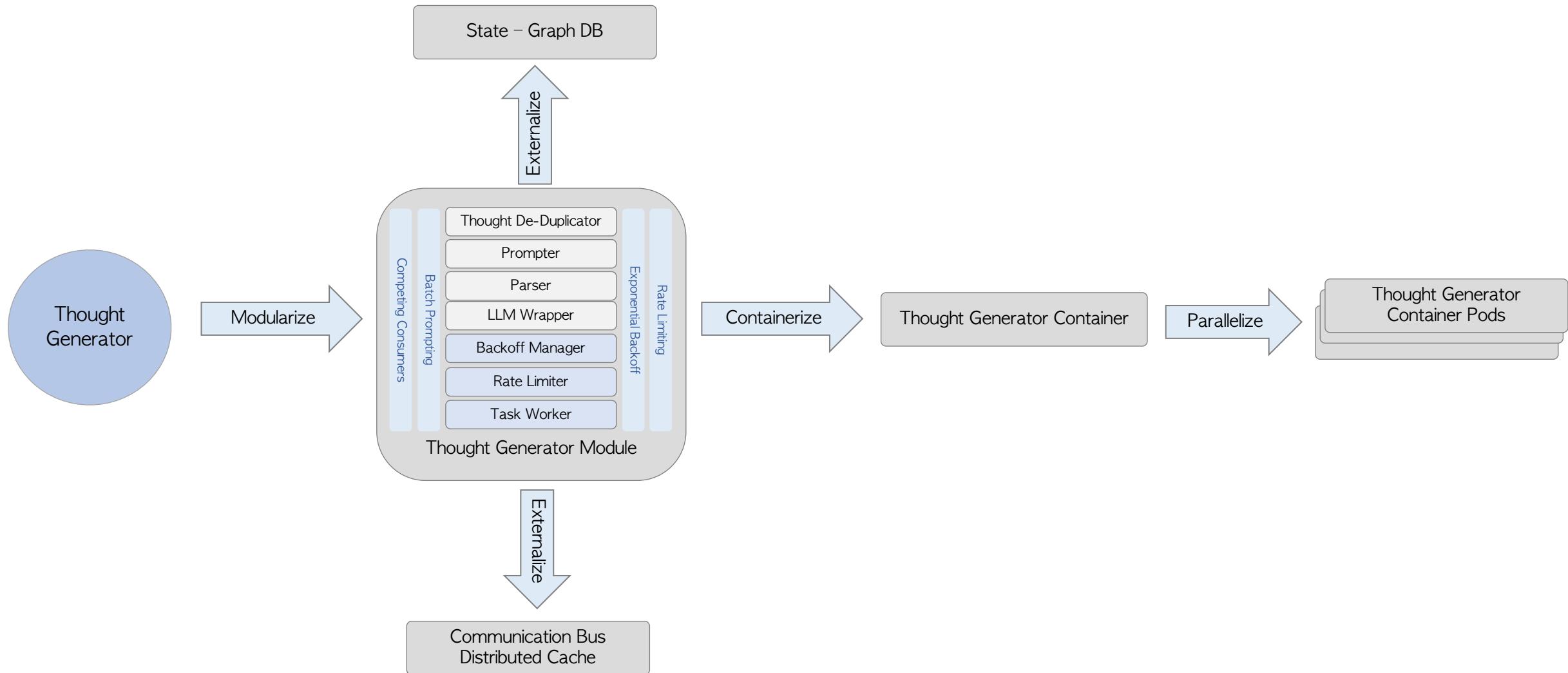
```
error_code=rate_limit_exceeded error_message='Rate limit reached for gpt-4 in organization org-o on tokens per min. Limit: 10000 / min.  
Please try again in 20s.'
```

One TOT (B=3, D=4)
80-100 LLM API calls

Prompt engineering + Software engineering

Problem	Patterns	Approach
Time Complexity	<ul style="list-style-type: none">• Functional Decomposition• Task parallelism• Divide and conquer• Early Termination	<ul style="list-style-type: none">• Parallelize the Tree Search algorithm• Decompose into independent and parallelizable workloads• Distribute workloads over a cluster of compute nodes
Space Complexity	<ul style="list-style-type: none">• Memoization• Spatial Locality• Lazy Initialization• Distributed Cache• State Externalization	<ul style="list-style-type: none">• Reduce Memory Pressure• Smaller token size• Use smaller local LLMs• Use Graph database• De-Duplicate Partial Solutions• Make ToT modules stateless
Throughput Limits	<ul style="list-style-type: none">• Exponential Backoff• Retry with Jitter• Rate Limiting	<ul style="list-style-type: none">• Wait before retrying failed requests• Track API requests made within a certain time• Cache to reduce the number of necessary calls
Latency	<ul style="list-style-type: none">• Event-driven parallelism• Competing Consumers• Scatter-Gather• Load Balancing• Orchestrator-Worker	<ul style="list-style-type: none">• Process execution units concurrently over distributed infrastructure• Use Asynchronous Processing• Use local LLMs
Cost to operate	<ul style="list-style-type: none">• Operator pattern• Auto-Scaling• Horizontal Scaling• Batch Prompting	<ul style="list-style-type: none">• Use Cloud-native design principles• Package the application and its dependencies as containers• Orchestrate to automate deployment, scaling, and operations• Automatically adjust active resources based on the current load• Pack multiple LLM API requests as one request

Patterns to Practice



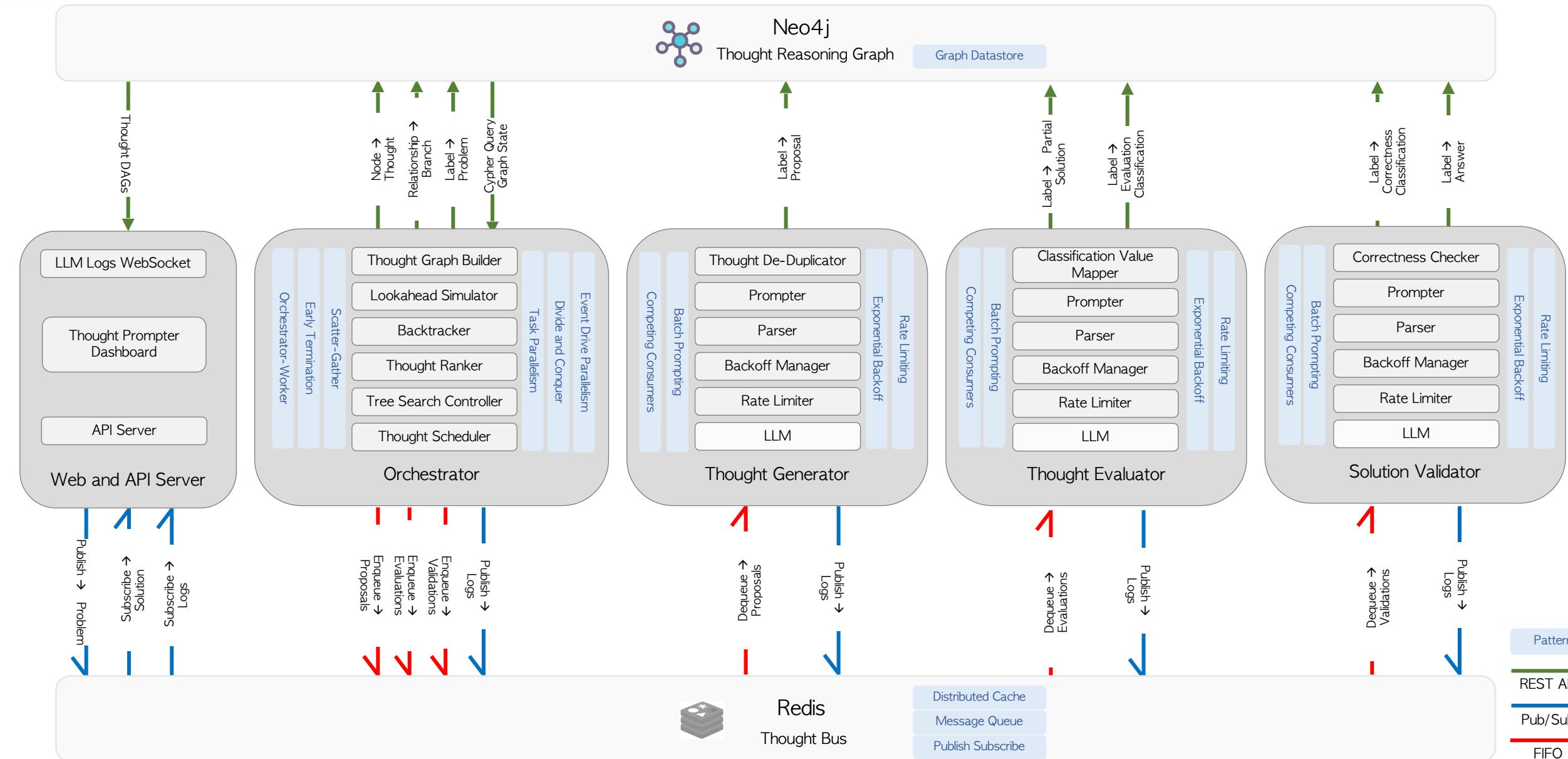
System Architecture and Event Flow



KubeCon

CloudNativeCon

North America 2023



Taming the operational complexity



KubeCon



CloudNativeCon

North America 2023

Kubify ToT Environment

- Express the desired state of the ToT environment as K8S
 - Pods , Deployments , Services
 - ConfigMaps and Secrets
 - Resource Quotas



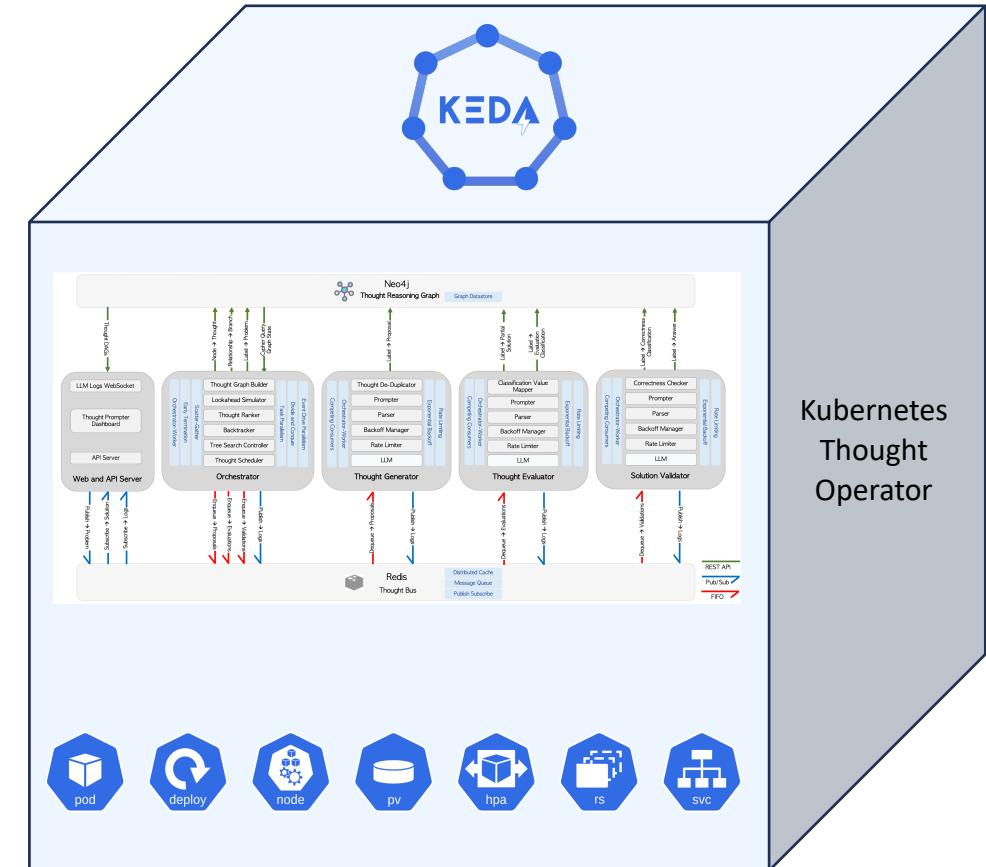
Automate operations - Kubernetes Thought Operator

- Deployment - Automate ToT deployments with predefined configurations and dependencies.
- Configuration Management - Manage ToT configurations, ensuring consistency and best practices.
- Updates and Upgrades - Automate update and upgrade ToT system with no or minimal downtime.
- Monitoring and Logging - Capture ToT relevant metrics and logs
- Self-Healing and Error Recovery - Automatically detecting and fix common issues, and perform error recovery
- Resource Optimization - Automatically adjust resource allocations based on real-time metrics and predefined policies
- Security and Compliance - Maintains compliance by consistently applying predefined policies.
- Service Discovery - Expose Orchestrator, Redis and Neo4j as service endpoints
- Storage Management - Manage the lifecycle of storage resources used by Neo4j and Redis as PVs and PVCs



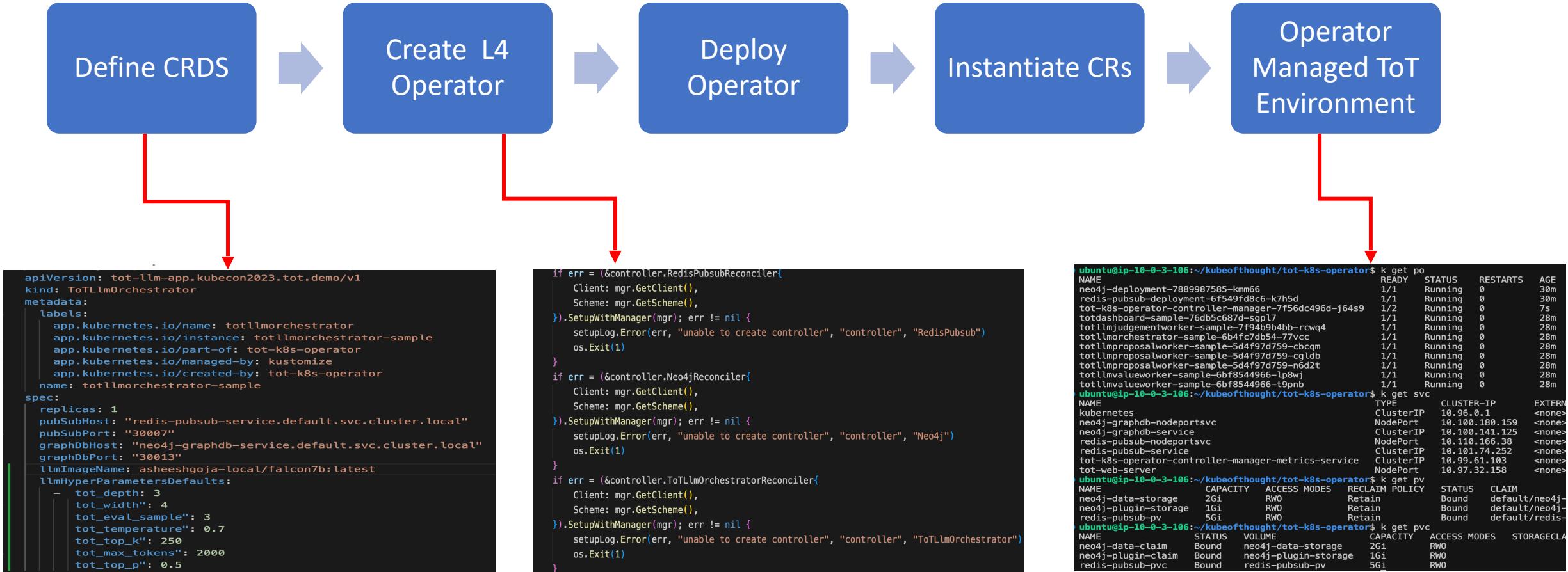
Custom KEDA Scaler

- Scaling ToT workloads - Automatically scale ToT workloads up or zero in by monitoring its custom metrics.
- Event-Driven scaling - Uses multiple event sources and metrics, for event-driven scaling
- Optimal Resource Utilization - Ensures that resources are allocated based on actual demand, improving efficiency



Kubified ToT Environment

Thought Operator – Under the hood

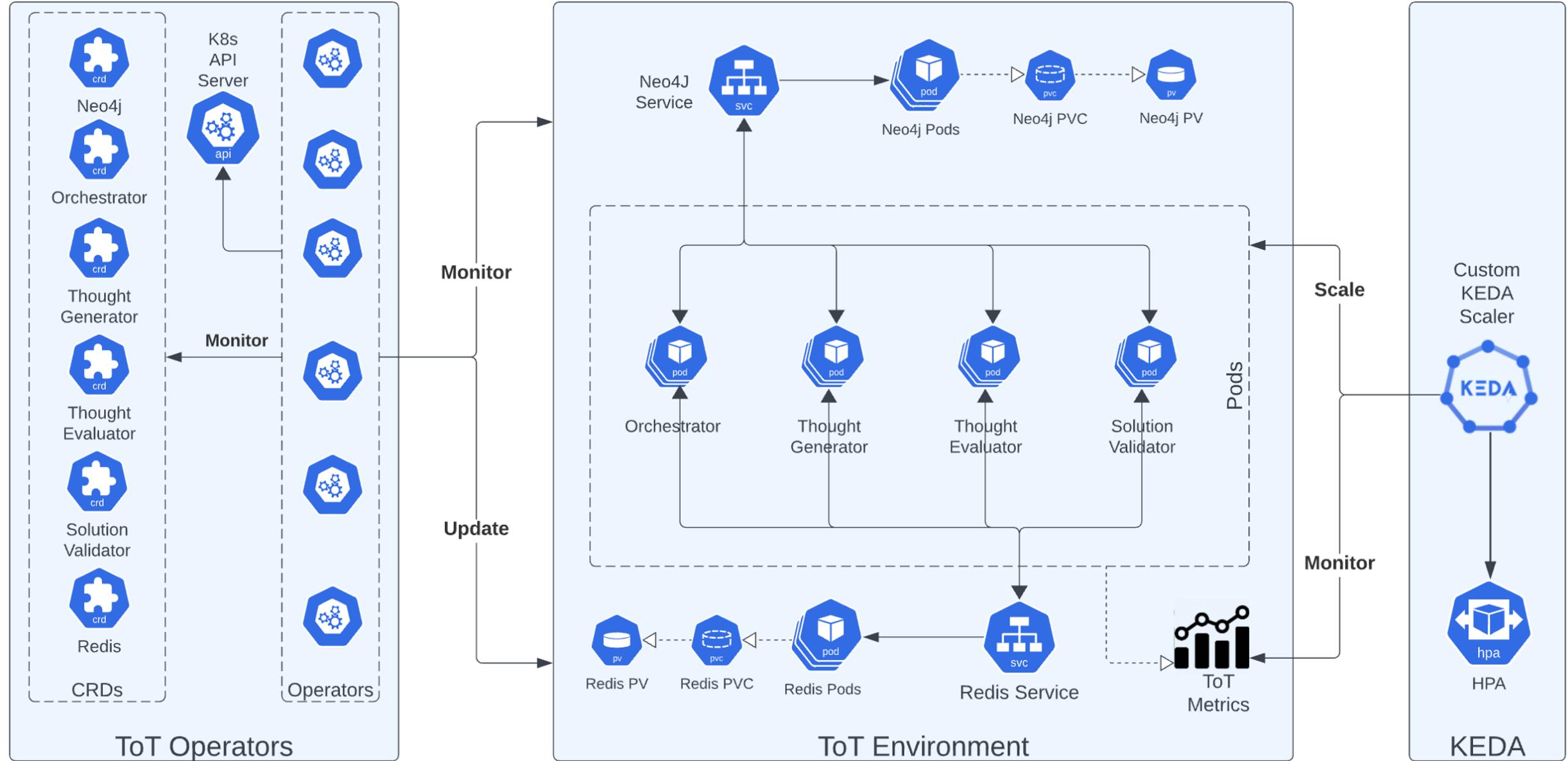


- Orchestrator CRD
- Thought Generator CRD
- Thought Validator CRD
- Solution Validator CRD
- Redi CRD
- Neo4j CRD

- Orchestrator Controller
- Thought Validator Controller
- Thought Validator Controller
- Solution Validator Controller
- Redi Controller
- Neo4j Controller

- Deployment
- Load balancing
- Service discovery
- Health check and failover
- Resource Isolation
- High Availability
- Storage Management

Overall Architecture - Operator + KEDA



See it in action – Demo



Kubernetes Dashboard x KubeofThought - Kul x + v ← → ↻

localhost:30018/demo.html ☆ Incognito ::

ToT Hyperparameters:

Tree Depth: Branching Factor:
Request Batch Size: Eval Threshold:
Traversal Algorithm: Concurrency Model:

LLM Parameters:

Temperature: Max Tokens:
top_k: top_p:

Problem:

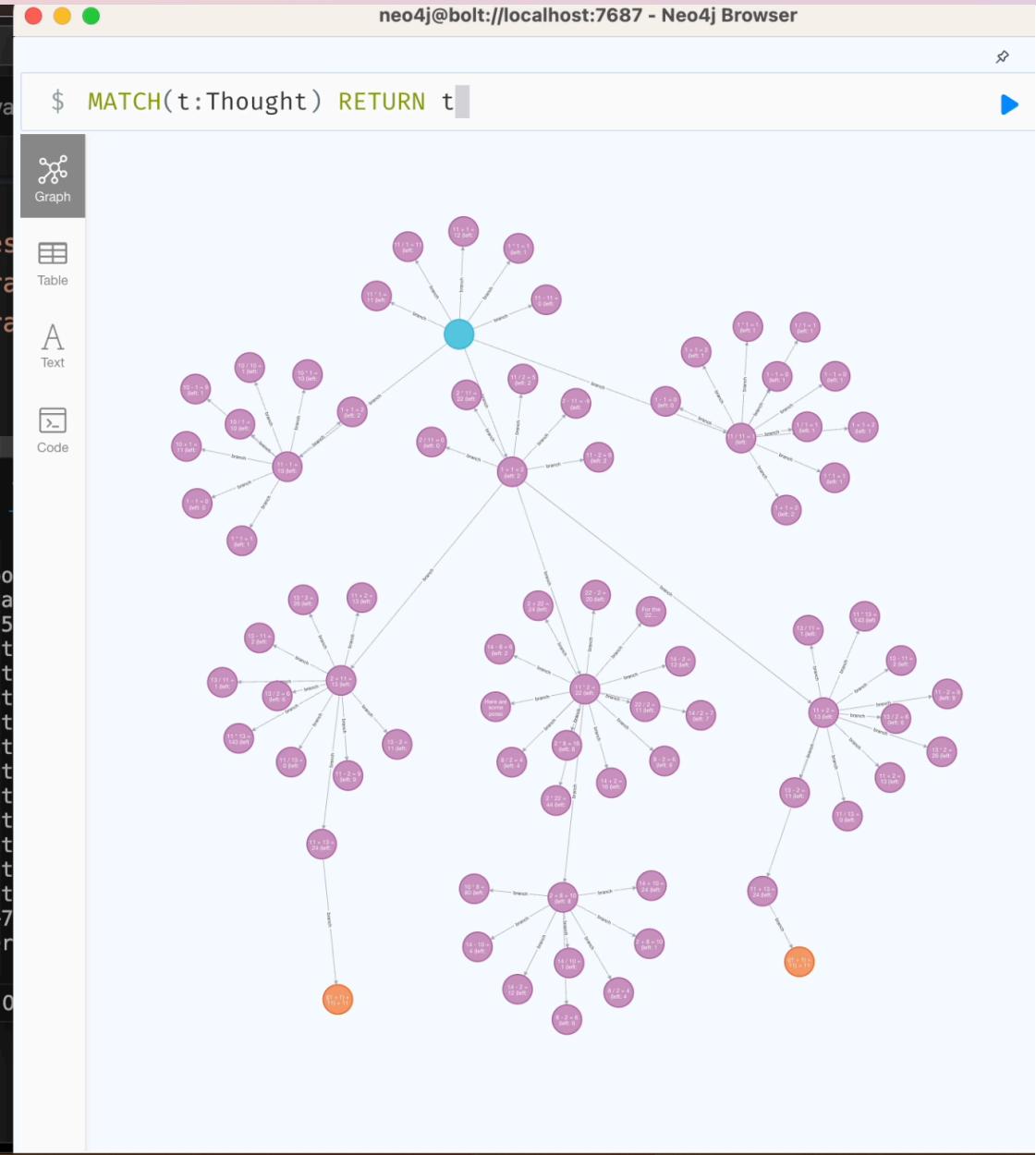
1 1 1 1 1 1

Solution:

```
1 + 1 = 2 (left: 2 1 1 1)
2 + 11 = 13 (left: 11 13)
11 + 13 = 24 (left: 24)
((1 + 1) + 11) + 11 = 24
```

Thought Logs:

```
tot-orchestrator-pod-cb4bb5c89-xw6cm - 2023-11-06 03:54:21.2
tot-orchestrator-pod-cb4bb5c89-xw6cm - 2023-11-06 03:54:21.1
tot-orchestrator-pod-cb4bb5c89-xw6cm - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-7hwxp - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-7hwxp - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-1mbvz - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-1mbvz - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-mm6ww - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-mm6ww - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-g8pn2 - 2023-11-06 03:54:21.1
tot-thought-evaluator-pod-558d5c8dbf-g8pn2 - 2023-11-06 03:54:21.1
```



Future direction

- Graph of Thoughts
 - Improve efficiency
 - Reduce Cost
- RL using ToT Style self-play
 - Generate new problem solving strategies
- ToT counterfactual decision making
 - Use to fine-tuning LLMs
- P=NP ???

Questions



References

- [Tree of Thoughts: Deliberate Problem Solving with Large Language Models](#)
- [Large Language Model Guided Tree-of-Thought](#)
- [Do large language models need sensory grounding for meaning and understanding?](#)
- [An In-Depth Look at the Transformer Based Models](#)
- [SoC – CS](#)
- [MIT-Notes on the Complexity of Search](#)
- [Graph of Thoughts: Solving Elaborate Problems with Large Language Models](#)
- [Large Language Model for Science: A Study on P vs. NP](#)