QiMeng-PerceptOS: Semantic-Aware Kernel Optimization for OS-Intensive Workloads via Hardware-Software Alignment

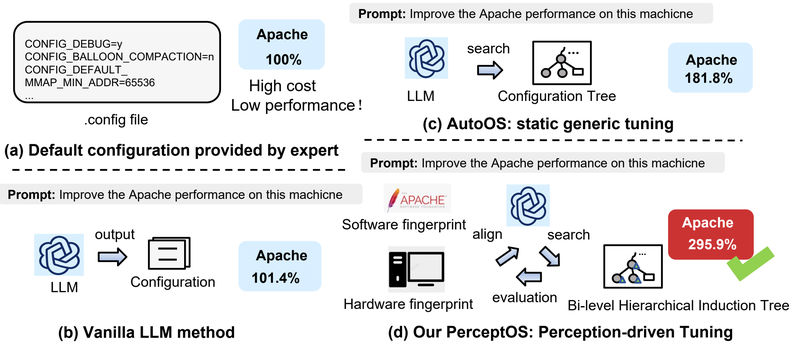
Huilai Chen, Yuanbo Wen, Liangfeng Li , Shaohui Peng, Jingzhe Zhu, Jun Bi, Xuzhi Zhang, Qi Guo, Ling Li, Yunji Chen

ICML International Conference On Machine Learning

Introduction

Optimizing OS kernels for certain workload is critical yet challenging due to over 15,000 compile-time options, 40–70 minute evaluation costs, and boot failure risks. Traditional methods fails. Existing LLM methods for optimization follow a static and heuristic open-loop paradigm, blind to dynamic runtime behaviors, causing a fundamental semantic mismatch.

Addressing these challenges, we propose QiMeng-PerceptOS, a closed-loop perception framework that aligns raw hardware-software telemetry with LLM reasoning through three collaborative modules. Experiments across diverse workloads show that it achieves significant performance breakthroughs by optimizing kernel configurations.



Problem Formulation

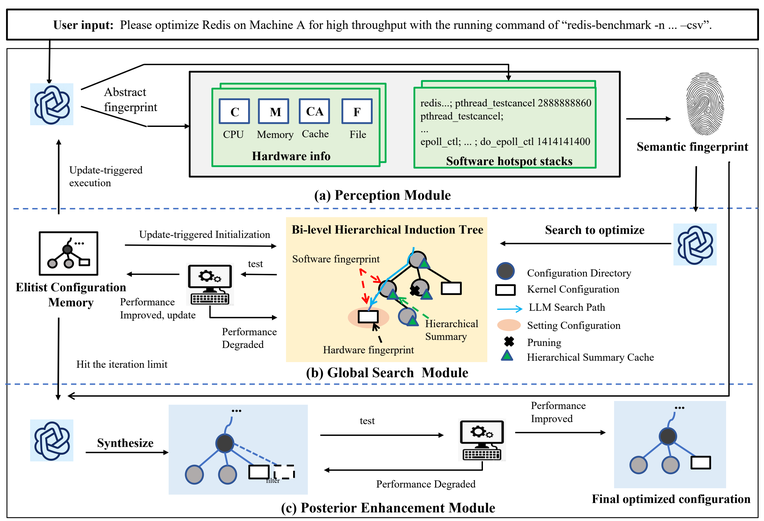
Optimize linux configurations represented as a tree (T = (N, E)) for specitic workload:

- N : Nodes representing configuration options and directories.
- E : Relationships between nodes.
- k : application k with evaluation metric f_k (2^N -> R)
- T_k : optimal configuration achieving optimal performance.
- T_0 : default configuration.
- H and S_k : denote hardware and software information.

Our goal: Find $T_k{}^*$ = A (T_0, H, S_k) , where $f_k(T_k{}^*) = max_T\ f_k$(T).

In essence, this entails operating system kernel tuning for specific applications within the context of high-dimensional, high-cost, and high-risk tasks.
Alignment between the kernel configuration and the application is required.

The design of Qimeng-PerceptOS



1. Perception Module: Distills unstructured hardware-software telemetry into semantic fingerprints, aligning low-level telemetry with LLM reasoning.

2. Global Search Module: Performs LLM-driven depth-first traversal over BHIT, exposing both local options and hierarchical subdirectory summaries (employs a cache-on-first-access strategy) to dynamically prune search space and align high-dimensional configurations with LLM reasoning.

3. Posterior Enhancement Module: Batch-reaudits historically modified options by LLM in elite memory M to synthesize trajectories, suppress hallucinations, and recover semantic mismatches between telemetry and configurations.

Evaluation

We compare QiMeng-PerceptOS with defaults, LLM-Vanilla, and AutoOS across 3 OS , 5 workloads, and 3 LLMs, with 13 search iterations and 2 posterior enhancement rounds under the same total budget as baselines.

Optimization results (throughput) across different environments compared to default.

Environment	LLM	Setting	Redis	Apache	AES	PostgreSQL	RAG
PC (AMD) Ubuntu 22.04	GPT-4o-mini	Vanilla	104.8%	101.4%	100.4%	101.1%	102.0%
		AutoOS	207.3%	181.8%	140.8%	142.8%	113.4%
		QiMeng-PerceptOS	**278.5%**	**295.9%**	**154.0%**	**154.5%**	**124.3%**
		Improv.	↑173.7%/↑71.2%	↑194.5%/↑114.1%	↑53.6%/↑13.2%	↑53.4%/↑11.7%	↑15.6%/↑10.9%
	DeepSeek-V3.2	Vanilla	226.3%	142.7%	101.2%	130.2%	116.7%
		AutoOS	220.9%	196.4%	140.6%	142.7%	117.2%
		QiMeng-PerceptOS	**294.0%**	**306.3%**	**143.5%**	**157.1%**	**128.4%**
		Improv.	↑67.7%/↑73.1%	↑163.6%/↑109.9%	↑42.3%/↑2.9%	↑26.9%/↑14.4%	↑11.7%/↑11.2%
	Gemini-3-pro-preview	Vanilla	109.0%	225.7%	141.4%	139.7%	122.2%
		AutoOS	264.0%	254.5%	150.8%	157.3%	118.4%
		QiMeng-PerceptOS	**296.6%**	**323.5%**	**157.0%**	**164.9%**	**128.7%**
		Improv.	↑187.6%/↑32.6%	↑97.8%/↑69.0%	↑15.6%/↑6.2%	↑25.2%/↑7.6%	↑6.5%/↑10.3%
PC (Intel) Fedora 42	GPT-4o-mini	Vanilla	108.9%	116.3%	105.5%	100.6%	111.3%
		AutoOS	108.2%	122.5%	117.7%	106.5%	102.2%
		QiMeng-PerceptOS	**147.0%**	**139.7%**	116.8%	**112.3%**	**133.8%**
		Improv.	↑38.1%/↑38.8%	↑23.4%/↑17.2%	↑11.3%/↑-0.9%	↑11.7%/↑5.8%	↑22.5%/↑31.6%
	DeepSeek-V3.2	Vanilla	132.5%	117.8%	106.6%	100.0%	136.5%
		AutoOS	121.6%	137.2%	111.9%	107.2%	109.7%
		QiMeng-PerceptOS	**143.7%**	**147.9%**	**118.4%**	**116.0%**	133.7%
		Improv.	↑11.2%/↑22.1%	↑30.1%/↑10.7%	↑11.8%/↑6.5%	↑16.0%/↑8.8%	↑-2.8%/↑24.0%
	Gemini-3-pro-preview	Vanilla	107.3%	121.5%	107.1%	109.7%	130.5%
		AutoOS	123.7%	149.1%	121.2%	113.9%	116.5%
		QiMeng-PerceptOS	**145.6%**	**165.7%**	118.8%	**115.5%**	**135.5%**
		Improv.	↑38.3%/↑21.9%	↑44.2%/↑16.6%	↑11.7%/↑-2.4%	↑5.8%/↑1.6%	↑5.0%/↑19.0%
Workstation Ubuntu 24.04	GPT-4o-mini	Vanilla	100.1%	102.3%	102.4%	104.2%	102.3%
		AutoOS	100.3%	100.0%	102.3%	103.4%	100.3%
		QiMeng-PerceptOS	**123.8%**	**128.9%**	**108.0%**	**109.2%**	**119.7%**
		Improv.	↑23.7%/↑23.5%	↑26.6%/↑28.9%	↑5.6%/↑5.7%	↑5.0%/↑5.8%	↑17.4%/↑19.4%
	DeepSeek-V3.2	Vanilla	108.1%	114.8%	103.0%	102.8%	115.9%
		AutoOS	100.0%	100.0%	101.5%	107.9%	100.0%
		QiMeng-PerceptOS	**124.6%**	**133.4%**	102.6%	**115.6%**	**120.1%**
		Improv.	↑16.5%/↑24.6%	↑18.6%/↑33.4%	↑-0.4%/↑1.1%	↑12.8%/↑7.7%	↑4.2%/↑20.1%
	Gemini-3-pro-preview	Vanilla	118.4%	114.6%	102.2%	109.3%	116.9%
		AutoOS	119.1%	109.4%	110.1%	101.4%	107.1%
		QiMeng-PerceptOS	**128.9%**	**139.3%**	**112.6%**	**113.4%**	**117.9%**
		Improv.	↑10.5%/↑9.8%	↑24.7%/↑29.9%	↑10.4%/↑2.5%	↑4.1%/↑12.0%	↑1.0%/↑10.8%

Average search time and LLM cost per iteration for three runs in the global search module on the workstation

LLM	Metric	Redis	Apache	AES	PostgreSQL	RAG
GPT-4o-mini	Time (min)	24.2	33.2	25.7	34.1	22.1
	Cost ($)	0.22	0.33	0.23	0.30	0.17
DeepSeek-V3.2	Time (min)	78.2	75.1	33.1	105.2	60.9
	Cost ($)	0.25	0.33	0.13	0.48	0.10
Gemini-3-pro-preview	Time (min)	159.6	175.6	153.1	171.5	132.4
	Cost ($)	3.91	4.38	3.14	4.65	3.74

Ablation of Perception and Global Search modules in GPT-4o-mini-based QiMeng-PerceptOS for Apache on Ubuntu workstation

	Perception Module		Search Module		Result
	HW	SW	Hierarchy Summary	Close Loop	
Ours	✓	✓	✓	✓	**128.90%**
Ablation 1	×	✓	✓	✓	114.34%
Ablation 2	✓	×	✓	✓	116.41%
Ablation 3	✓	✓	×	✓	121.70%
Ablation 4	✓	✓	✓	×	116.70%

Ablation study of posterior enhancement module (PE) for Apache on Ubuntu workstation (35% probability of gain across all test)

Setting	Redis	Apache	AES	PostgreSQL	RAG
13 search	260.95%	295.89%	151.93%	149.88%	124.25%
15 search	260.95%	295.89%	152.19%	149.88%	124.25%
13 search+PE	**278.53%**	295.89%	**154.00%**	**154.50%**	124.25%

References

Huilai Chen, Yuanbo Wen, Limin Cheng, Shouxu Kuang, Yumeng Liu, Weijia Li, Ling Li, Rui Zhang, Xinkai Song, Wei Li, et al. AutoOS: make your os more powerful by exploiting large language models. In Forty-first International Conference on Machine Learning, 2024.

Kai Mei, Xi Zhu, Wujiang Xu, Wenyue Hua, Mingyu Jin, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. AIOS: Llm agent operating system. arXiv preprint438 arXiv:2403.16971, 2024

Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis Ioannidis, Karthik Subbian, Jure Leskovec, and James Y Zou. Avatar: Optimizing llm agents for tool usage via contrastive reasoning. Advances in Neural Information Processing Systems, 37:25981–26010, 2024