AATC 2024

Fada-
A LLM based Farm Debug Agent Tool

AMD ASIA TECHNICAL CONFERENCE

AMD
together we advance_innovation

Abstract

- Farm is a critical validation process before silicon production, it validates APU/CPU and associate SW stack functionality and stability via a large scale of platforms running with different test cases circularly. Many tools and methodologies have been hatched up to reduce human operations in daily farm work, however, to ensure timely launch of the product, the accelerated convergence of farm issues is truly important.

- In history projects experience, accumulated tens of thousands of historical farm issue records. Team expecting these data can be well utilized by tools to provide guidance for solving issues in future projects.

- In this paper, will give a **Fa**rm **D**ebug **A**gent tool (called **Fada**) solution based on requirement and problem analyzing, discuss about the technical implement detail about RAG[1], embedding, data cleaning, agent design and prompt optimization, etc. A demo and conclusion will be provided at the end of this paper.

- Fada is an AI chatbot based on LLM[2], providing user a friendly web GUI. It can automatically retrieval previous farm history records best match user symptom description, analyze related JIRA ticket, provide debug clue, give next step suggestion for user reference. Now in trial run phase by Client validation team.

1. RAG = Retrieval-Augmented Generation
2. LLM = Large Language Model

Background

- Pain point from validation and debug team
  - Farm issue might be related to BIOS, driver, software/hardware stability, it's hard for junior tester find out what's next step for debug.
  - For issue triage and debug usually need multiple team collaboration, time/resource cost.
  - Validation team is seeking for a proper method to provide debug insight based on history record/configuration.

- Farm record database readiness
  - Validation team has accumulated millions of history farm test record in daily work.
  - Test record has been uploaded to web database and well managed.
  - Test record has failure description, configuration, debug logs and associated JIRA ticket number. Which can be used for new issue reference via similarity retrieval.

- Bloom of AI industry and the supportive strategy from company
  - Breakthroughs in NLP[1] especially LLMs based on Transformer[2] is changing life.
  - AMD developed excellent AI hardware and encouraged employee to leverage AI technology for daily work efficiency improvement.
  - IT provide AMD domain LLM API (e.g., Azure OpenAI, Google Gemini, Llama) service for AI tool development.

1. NLP = Natural Language Processing
2. Transformer = A deep learning model invented by Google Brain based on attention mechanism

Proposal

Background

With attention mechanism and Transformer model development, “LLM” such as GPT4 trained by rich knowledge material has excellent content generation capability based on context.

With mature of embedding and vector similarity search, local database can be “attached”, by providing related local knowledge and data along with user question to LLM, can get more accurate and targeted response.

End user need a friendly GUI to “talk” with AI. Easily access via network can avoid any installation.

AMD advantage

AMD LLM API Gateway provides a trusted, scalable and compliant access to public LLMs.

Tens of thousands of history fail record available
JIRA tickets, summary, solution and comments
Local debug knowledge.

Existing server can be used as webpage and service server.

Idea

Develop a tool based on AI, LLM as the “brain”.

By retrieval local database + proper prompt to let LLM generate debug proposal.

Friendly “ChatGPT” like webpage GUI with AMD NT authentication.

Prior work

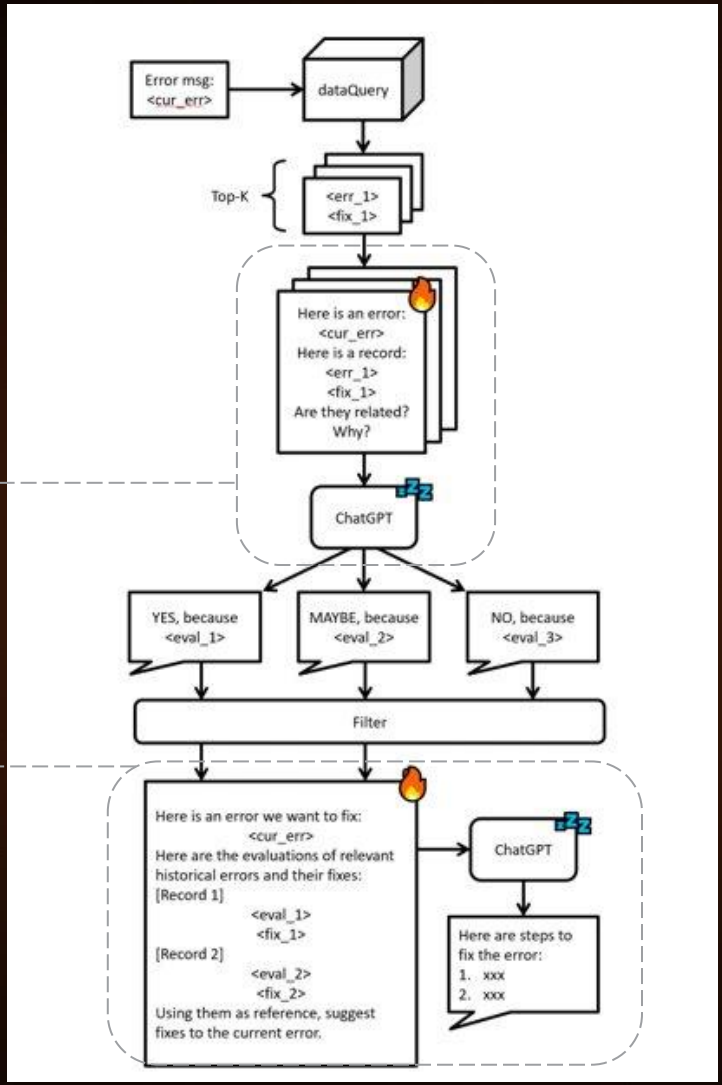
- MLDG group has some prior study based on data query and ChatGPT which given a solution to let AI provide error fix suggestion, methodology as left side block diagram shows.

- This is a very insightful RAG methodology but has below limitations.

  1. Need to ask GPT multiple times (K) to filter all truly related records. This will significantly increase the total time cost and API call cost.

  2. After filter process need to call ChatGPT one more time to provide final answer.

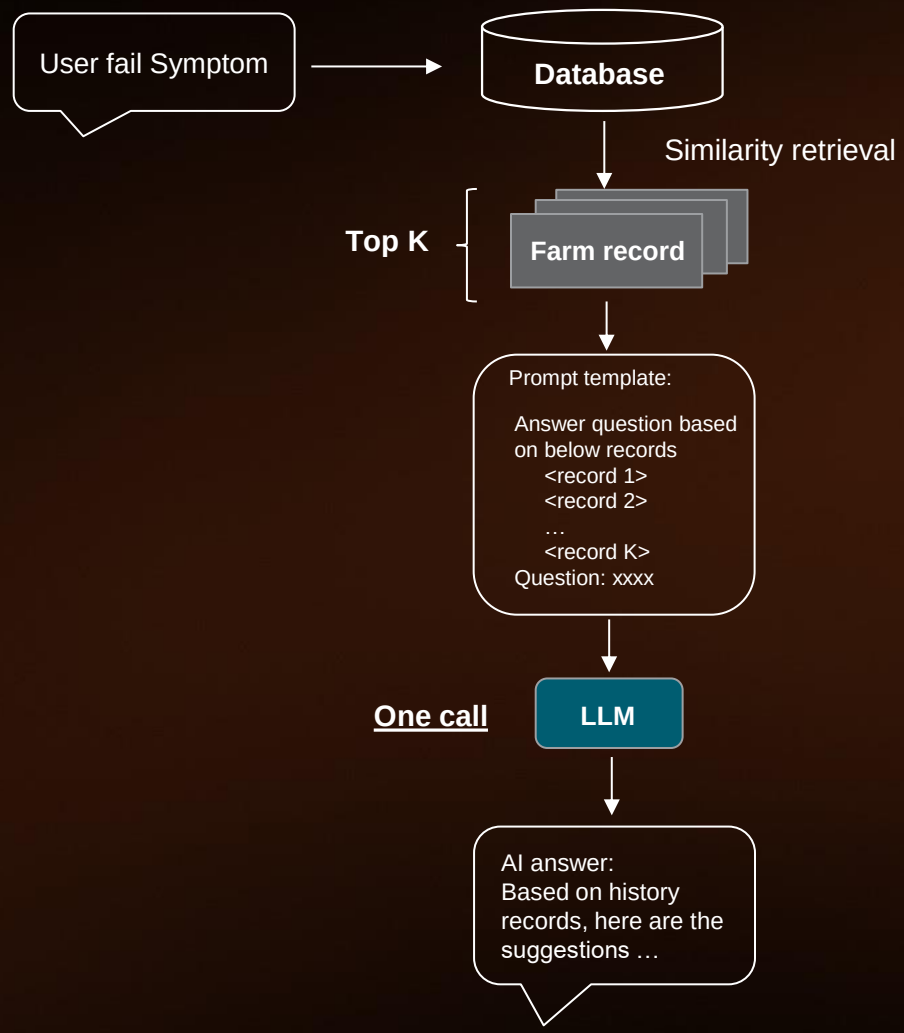
  3. Need to prepare <error> <fix> paired records as database but for use case discussed in this paper most of validation farm record don't have fix solution provided.



Simplified data retrieval structure

- By removing retrieval filter process in original methodology, structure can be simplified to only call LLM once (was K+1).

- To get as accurate answer as the solution with filter, need to do more test and improve from multiple action such as prompt engineering. Will discuss later in this paper.

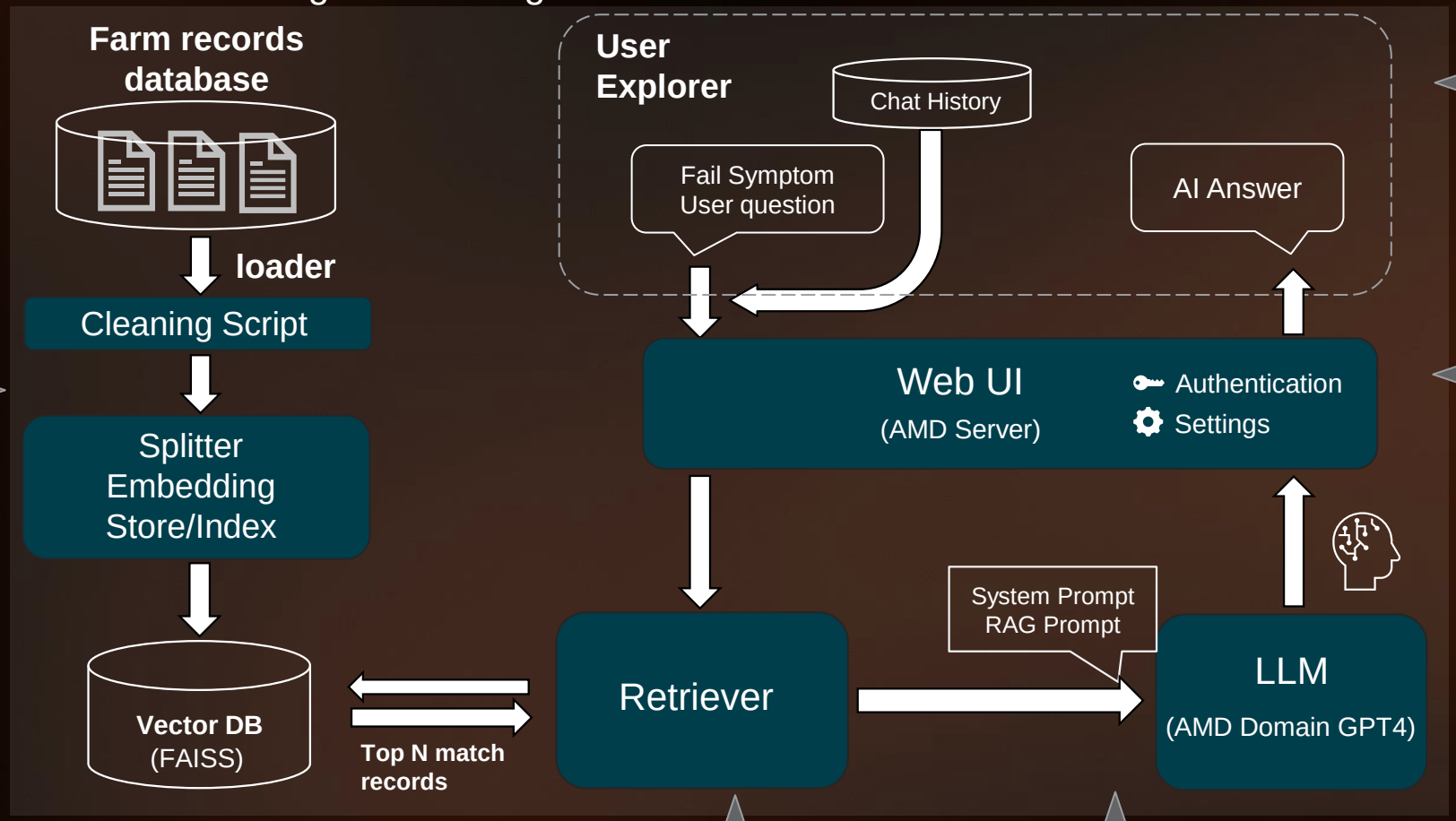


More problem discussion

Problem	Discussion
LLM has rich and wide knowledge based on training material crossing domains. How can we well leverage its knowledge on x86 system domain? How to let AI understand AMD specific terms and knowledge?	Most LLM provide system prompt mechanism, it can be behaved to a specific domain expert by properly system prompt setting. With attach external AMD specific knowledge database, AI can search the terms if it's not familiar with.
How these large quantity of history farm records can be well storage and recall in a most efficient way.	Traditional cosine similarity calculation seems cost more time especially for a large number of cycle. Need to find if there is any better solution for vector database storage and retrieval.
Which will be the best way to store user chat history?	Store at server side will increase server loading, also a problem to clean up user history if file over sized, store at user explore side could be a better approaching so user can self decide clear them or not.
Farm record consisted by a lot of information like configuration, BIOS/Driver version, user comment, debug comment, related JIRA etc. They are originally stored by table format, how to make it well recognized and understanding by AI?	Farm record data should be well re-organized, tagging and clean up for better understanding from semantical wise.

Proposed Fada Diagram

With a high efficiency RAG structure, by adding webpage front-end and database embedding portion, define the initialized infrastructure as below for archiving basic debug assistant function.



Farm data cleaning labeling and proper split for embedding for better semantical expression

Chat history stored at user explorer local storage for flexible user chat control

Login page for user authentication. Settings page provide a mean for Fada fine tune.

High efficiency RAG structure + efficient similarity search powered by FAISS[1]

System prompt form the LLM characteristic, RAG prompt template increasing the answer accuracy

1. FAISS is a library from Meta, it builds a data structure in RAM, retriever can perform high speed Euclidean distance calculation when given a target vector

Approaching - Data cleaning and consolidate

Original farm record has more than 20 columns, not well organized, very difficult for machine to understand.

Date	Category	Test Items	System No.	Status	Memory	Part	GPU	KVM IP	Wombat IP	OS	BIOS	Driver	SSD	Common Comments	ADS Report link/Dump file path/PSTReport	Detail comments	Debug Comments	Setting	Ticket	Windbg	Release date
5/12/2023	Coverage:RBI1001gC_DisWDTWL.Stress	3DMarkDandia(custom_timespy_windowmode (CUSTOM))*6h	PHX_FP8_Birman_LP5x_045_Navi33	Fail	P0 CHANNEL A 4 GB 7500 MHz P0 CHANNEL B 4 GB 7500 MHz P0 CHANNEL C 4 GB 7500 MHz P0 CHANNEL D 4 GB 7500 MHz	AMD Ryzen 7 7840HS w/ Radeon 780M Graphics	AMD Radeon 780M Graphics	kvm-sr8-3.amd.com	10.67.212.85	WIN11x64 10.0.22621.1555	RBI1001gC_DisWDTWL.FD	22.40.03.38-230428a-391535C-ATI	SAMSUNG MZVL2512HCJQ-00B00 476 GB	Black screen with PC:0000EC35 ,MP0 FW Status:D1420000.		Fail within 3min.	[Haiyan] PostCode: 0x0000EC35 PSP: hang 0x002EC653, FW Status: 0xD1420000 SMU: MP1_LX3_PDEBUGPC hang, 0x00029AF7 -> ReadHubIfNb() fch.c:195 SMU: MPCCX_LX3_PDEBUGPC hang, 0x00001e92 -> IdleLoop() rtos.c:82 GFX Info: GFX_IMU_MSG_FLAGS = 0x8000001F no gfx busy PwrMgtStatus: 0x50000000 CSTATEHISTCTLTOP: 2 LOWESTACPICSTATE: 2 No HWA. Hard lock, all cores no response. Create new jira and keep it for SMU team.	TDRLevel=0	PLAT-132997		

Reduced to 8 column by removing unused info and uniting similar terms, changed to more comprehensive column name.

Test date		Test Item	Test Bench	Test Result	Test Config								Fail Description			Debug Log	Associated Ticket No.			
Date	Category	Test Items	System No.	Status	Memory	Part	GPU	KVM IP	Wombat IP	OS	BIOS	Driver	SSD	Common Comments	ADS Report link/Dump file path/PSTReport	Detail comments	Debug Comments	Setting	Ticket	Windbg	Release date
5/12/2023	Coverage:RBI1001gC_DisWDTWL.Stress	3DMarkDandia(custom_timespy_windowmode (CUSTOM))*6h	PHX_FP8_Birman_LP5x_045_Navi33	Fail	P0 CHANNEL A 4 GB 7500 MHz P0 CHANNEL B 4 GB 7500 MHz P0 CHANNEL C 4 GB 7500 MHz P0 CHANNEL D 4 GB 7500 MHz	AMD Ryzen 7 7840HS w/ Radeon 780M Graphics	AMD Radeon 780M Graphics	kvm-sr8-3.amd.com	10.67.212.85	WIN11x64 10.0.22621.1555	RBI1001gC_DisWDTWL.FD	22.40.03.38-230428a-391535C-ATI	SAMSUNG MZVL2512HCJQ-00B00 476 GB	Black screen with PC:0000EC35 ,MP0 FW Status:D1420000.		Fail within 3min.	[Haiyan] PostCode: 0x0000EC35 PSP: hang 0x002EC653, FW Status: 0xD1420000 SMU: MP1_LX3_PDEBUGPC hang, 0x00029AF7 -> ReadHubIfNb() fch.c:195 SMU: MPCCX_LX3_PDEBUGPC hang, 0x00001e92 -> IdleLoop() rtos.c:82 GFX Info: GFX_IMU_MSG_FLAGS = 0x8000001F no gfx busy PwrMgtStatus: 0x50000000 CSTATEHISTCTLTOP: 2 LOWESTACPICSTATE: 2 No HWA. Hard lock, all cores no response. Create new jira and keep it for SMU team.	TDRLevel=0	PLAT-132997		

Consolidated each record to 1 row text (csv) by removing LF and adding labels, get ready for text split and embedding.

Test Date:5/12/2023 Test Item:3DMarkDandia(custom_timespy_windowmode (CUSTOM))*6h Test Bench:PHX_FP8_Birman_LP5x_045_Navi33 Test Result:Fail Test Config:"BIOS Version:RBI1001gC_DisWDTWL.FD GPU Driver:22.40.03.38-230428a-391535C-ATI CPU:Ryzen 7 7840HS w/ Radeon 780M Graphics GPU:Radeon 780M Graphics OS:WIN11x64 10.0.22621.1555 Memory:CHA 4G 7500 CHB 4G 7500 CHC 4G 7500 CHD 4G 7500 SSD:SAMSUNG MZVL2512HCJQ-00B00 476 GB" Fail Description:"Black screen with PC:0000EC35 ,MP0 FW Status:D1420000." Debug log:"PostCode: 0x0000EC35 PSP: hang 0x002EC653, FW Status: 0xD1420000 SMU: MP1_LX3_PDEBUGPC hang, 0x00029AF7 -> ReadHubIfNb() fch.c:195 SMU: MPCCX_LX3_PDEBUGPC hang, 0x00001e92 -> IdleLoop() rtos.c:82 GFX Info: GFX_IMU_MSG_FLAGS = 0x8000001F no gfx busy PwrMgtStatus: 0x50000000 CSTATEHISTCTLTOP: 2 LOWESTACPICSTATE: 2 No HWA. Hard lock, all cores no response. Create new jira and keep it for SMU team. scan https://scanview/id/738320 MCA scan https://scanview.amd.com/analyze/report?id=738320&reportid=6763 " Associated Ticket No.:PLAT-132997

Approaching - Data split and embedding

- **Text embedding** is converting a chunk of text to high dimension vector via well trained embedding model. Therefore, retrieval module can find the most related chunks via calculating the vector distance between user question and embedded database.
- For accurate history record retrieval, need to split the farm data into chunks by each record. We can easily do that because in previous data cleaning phase, each record has been stored in text format and occupied 1 row. With this design, the splitter can recognize each record via the line break.

- It is also very important to select a proper **embedding model**, this can directly affect the retrieval result.
- We created some test query to verify the performance of embedding model. One of the query is search “FP8 Birman Hang OS with PC:B0000DB7”, expecting the retrieval result should be all fail records on FP8 Birman platform.
- By comparing 3 embedding model provided by OpenAI (Ada002, embedding 3L, embedding 3S), although “Ada” is not the latest embedding model, but it shows best accuracy than others. All retrieval result comes from FP8 Birman.

Embedding Model	FP8 Birman Record Retrieval Test[1]
text-embedding-ada-002	10/10
text-embedding-3-small	3/10
text-embedding-3-large	3/10

1. Retrieval top 10 fail records of FP8 Birman hang OS, check the percentage of return records precisely belong to FP8 Birman

Approaching – LLM selection and Prompt setting

- Per discussed in previous session, we chose a higher efficiency RAG structure which removed the k-times retrieval filter process. However, this brings in higher requirement to the only 1-time LLM's answer.
- Therefore, a powerful LLM model with larger context window is necessary. With that purpose, we selected OpenAI "GPT4-turbo" with up to 128K tokens context window. It provides sufficient room to accommodate all the top-K retrieval results (select K=20 in trail run, approx. 4k~6k tokens), meanwhile leaves room for chat history and long prompt accommodation.

LLM Model	Context Window (tokens)
GPT3.5 turbo (4k)	4k
GPT3.5-turbo (16k)	16k
GPT4 (8k)	8k
GPT4 (32k)	32k
GPT4 turbo	128k
GPT4o[1]	128k

- In another hand, prompt setting is even more important. System prompt can be well leveraged for LLM character setting.

System Prompt

Character Setting

Your name is Fada, you are a world class x86 PC debug expert, familiar with BIOS, Graphic driver, SMU, PSP, etc.
You are good at finding debug clue based on given history fail records.
You don't make up; you give analysis based on truth and knowledge.

Instructions and format requirements

Please make sure you complete the objective above with following rules:
1/ If user provide history record, you must cover all the records have same PC and MP0 status.
2/ Give related history record number, test bench, configuration and associate ticket number when user provided history records for analysis.
3/ Bold all the ticket number in your answer.
4/ Analyze the debug log and find clue.
5/ Provide the ticket hyperlink in conclusion, link address: https://ontrack-internal.amd.com/browse/PLAT-xxxxxx.

RAG Prompt

Check the issue symptom in question, analyze all related test record below to give debug clue, if there is no related record, say 'Can't find related information in database, here is my answer', then give an answer by yourself.

<record 1>
<record 2>
…
<record K>

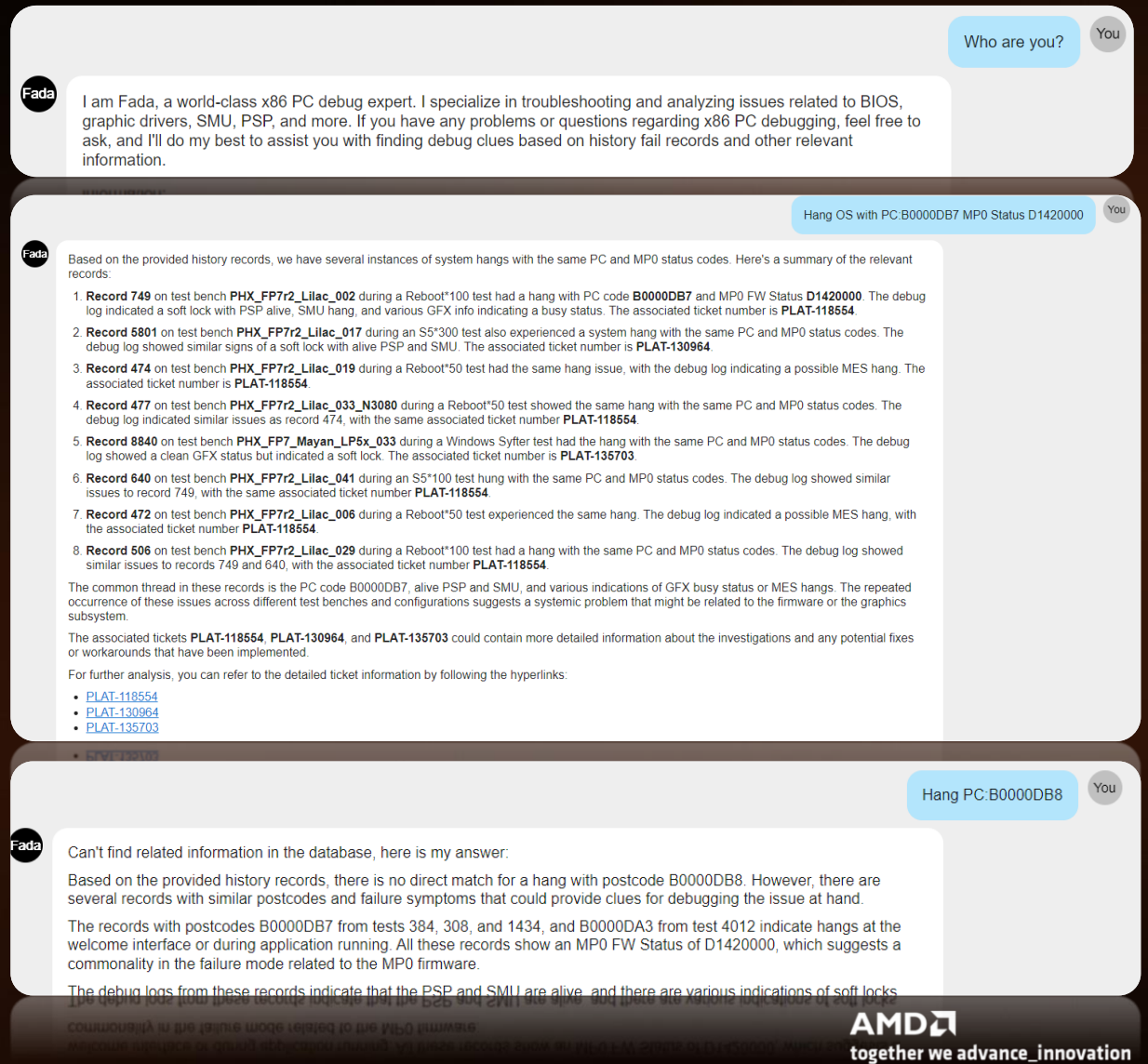
Question: <user fail symptom>

Guide LLM to only based on related records to provide suggestion

1. GPT4o is published after we finished Alpha test. So didn't select GPT4o model here. Even that, after we test GPT4o, GPT-turbo still more stable in handling long context.

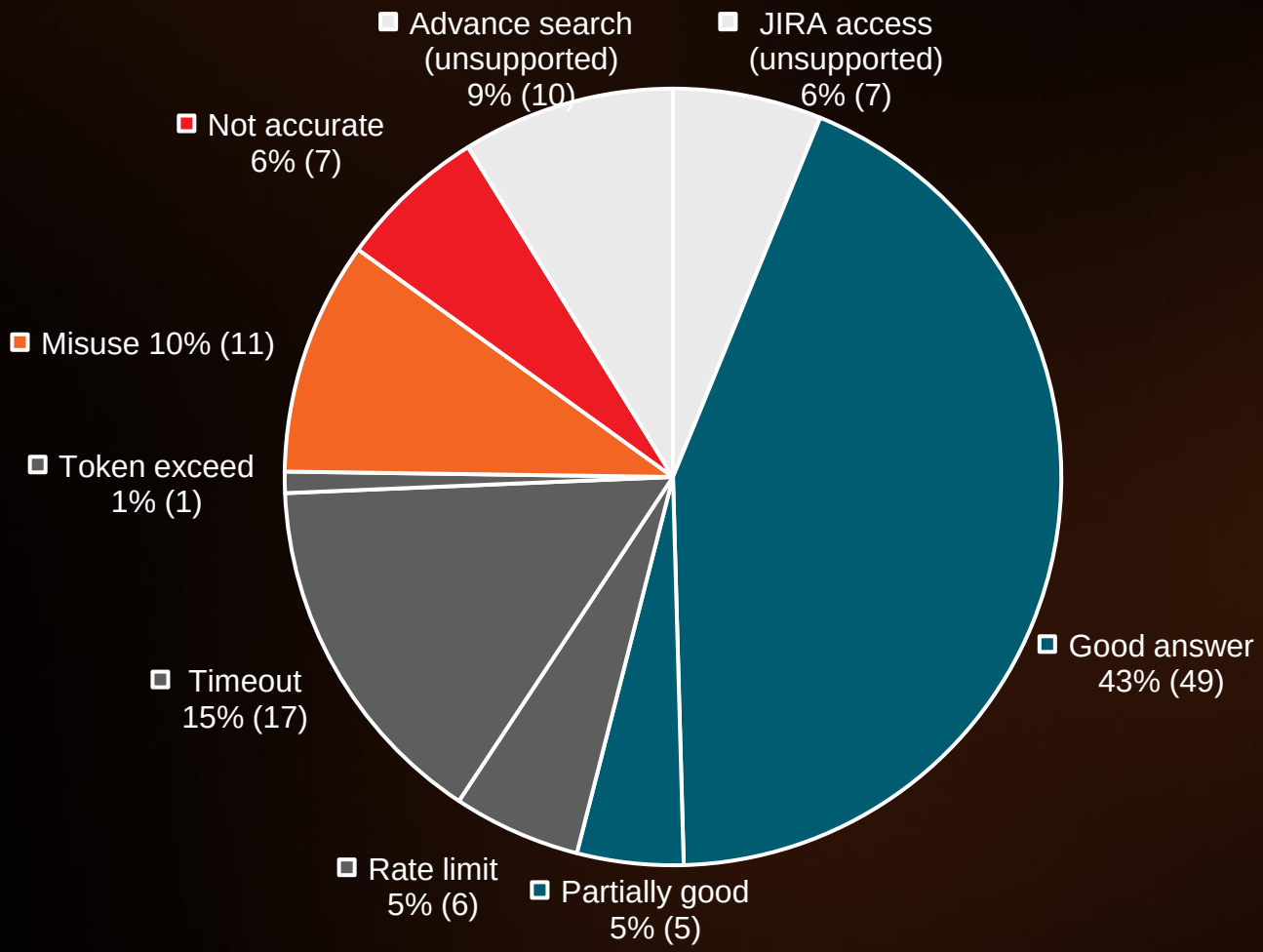
Approaching – LLM selection and Prompt setting

- After setting system prompt, LLM can recognize itself as a debug expert to help find debug clue based on history records. This can guide LLM well leverage its knowledge in x86 debug domain.
- System prompt + RAG prompt makes LLM can retrieval, filter and provide suggestions in only one call. With properly format and well-organized structure.
- With the requirement in system prompt, LLM will filter and only keep the failure records precisely match provided Postcode and MP0 Status in history record.
- LLM also summarized and provided all related JIRA tickets hyperlink at the end of its answer.
- LLM after prompt tuning will not make up, if there is no match record, it will let user know, and provide more clues based on analysis by itself for user reference.



Approaching - Fada Alpha Test Feedback

After 1st phase development, Fada open Alpha test to end user for feedback collection, over 100 feedback collected

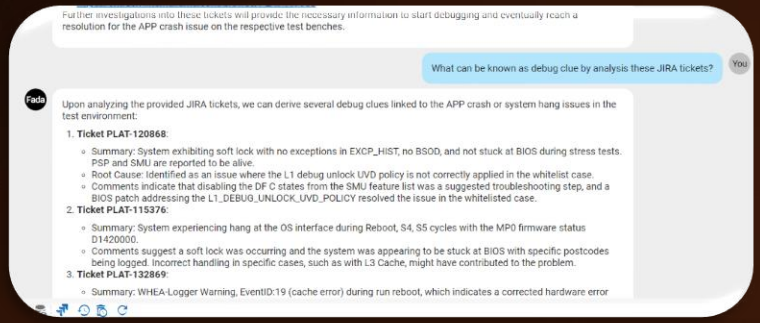


- 48% answer get satisfactory feedback
- 21% is IT limitation and issue
- 15% categorized as unsupported feature
- 10% can be improved via user usage
- 6% not accurate - need to improve

Approaching - Fada Feedback Analysis (Unsupported feature)

We carefully analyzed the user feedback and given solution and idea proposal for next step optimization plan.

JIRA access requirement from end user	Solution/Idea
希望能根据JIRA ticket里的信息，归类各种可能的root cause	• + JIRA access and summary function • JIRA data merge in and further feature
希望可以列出完整的jia信息	
希望把JIRA具体描述一起贴上	
列出Root cause与xxx相关的issue	

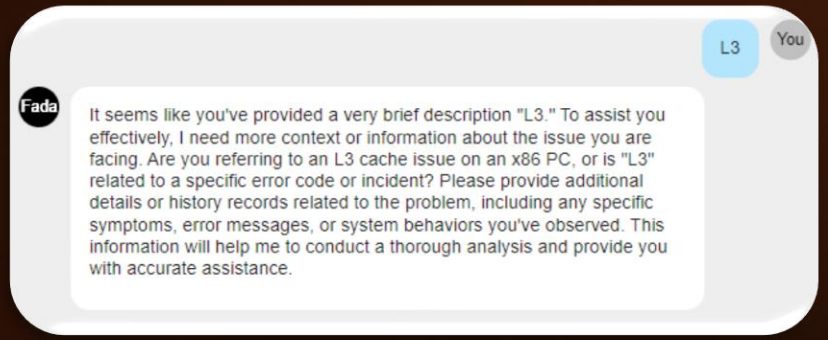


Advance search features requirement	Solution/Idea
Which 5 test items are the highest fail rate?	• Database pre-processing • Multidimensional clustering via ML • Agent + script • Hybrid search: SQL + semantic • 3rd party tool integration
List all fail units?	
Does this system fail several times while running heaven?	

Approaching - Fada Feedback Analysis (Usage)

User Operation Miss	Solution/Idea
Forget to turn on the knowledge DB function	• Auto access local DB or JIRA according to context

Short prompt or AMD specific term	Solution/Idea
“L3”	• AI guide user for more comprehensive question
“Setting”	
“TCP PF”	
“RZ616 Lost”	• More local knowledge help AI understanding



Approaching - Fada Feedback Analysis (Accurate Improvement)

Problem:
Hard to retrieval short and weak-semantic content

Such as “PLAT-135729” “HWA36”
Limited by vector similarity, hard to match by semantic due to weak-semantic and flooding effect

Solution/Idea

- Database Tagging, search by field to mitigate the flooding effect
- Hybrid Retrieval to mitigate the limitation of vector similarity

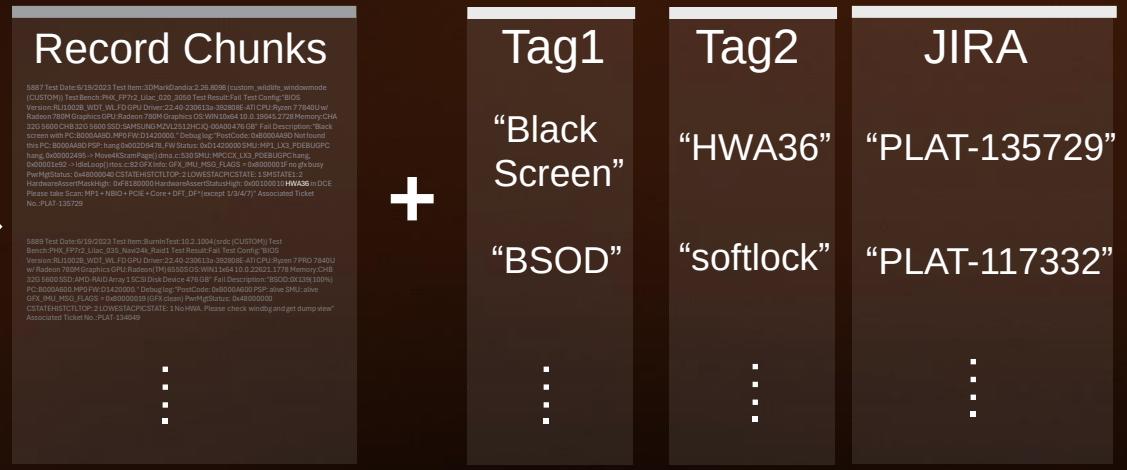
Record Chunks

⋮

5887 Test Date:6/19/2023 Test Item:3DMarkDandia:2.26.8098 (custom_wildlife_windowmode (CUSTOM)) Test Bench:PHX_FP7r2_Lilac_020_3050 Test Result:Fail Test Config:"BIOS Version:RLI1002B_WDT_WL.FD GPU Driver:22.40-230613a-392808E-ATI CPU:Ryzen 7 7840U w/ Radeon 780M Graphics GPU:Radeon 780M Graphics OS:WIN10x64 10.0.19045.2728 Memory:CHA 32G 5600 CHB 32G 5600 SSD:SAMSUNG MZVL2512HCJQ-00A00 476 GB" Fail Description:"Black screen with PC:B000AA9D.MP0 FW:D1420000." Debug log:"PostCode: 0xB000AA9D Not found this PC: B000AA9D PSP: hang 0x002D9478, FW Status: 0xD1420000 SMU: MP1_LX3_PDEBUGPC hang, 0x00002495 -> Move4KSramPage() dma.c:530 SMU: MPCCX_LX3_PDEBUGPC hang, 0x00001e92 -> IdleLoop() rtos.c:82 GFX Info: GFX_IMU_MSG_FLAGS = 0x8000001F no gfx busy PwrMgtStatus: 0x48000040 CSTATEHISTCTLTOP: 2 LOWESTACPICSTATE: 1 SMSTATE1: 2 HardwareAssertMaskHigh: 0xF8180000 HardwareAssertStatusHigh: 0x00100010 **HWA36** in DCE Please takeScan: MP1 + NBIO + PCIE + Core + DFT_DF*(except 1/3/4/7)" Associated Ticket No.: **PLAT-135729**

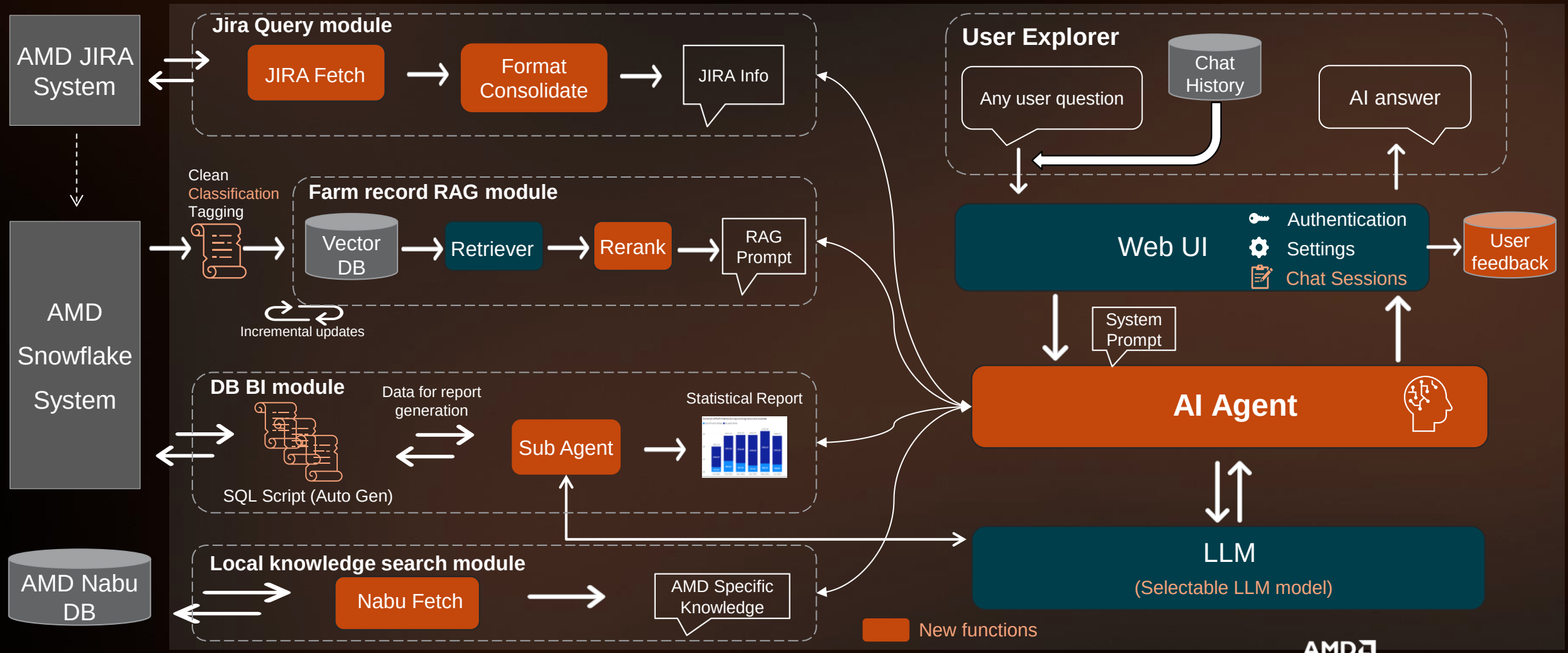
5888 Test Date:6/19/2023 Test Item:BurnInTest:10.2.1004 (srdc (CUSTOM)) Test Bench:PHX_FP7r2_Lilac_035_Navi24k_Raid1 Test Result:Fail Test Config:"BIOS Version:RLI1002B_WDT_WL.FD GPU Driver:22.40-230613a-392808E-ATI CPU:Ryzen 7 PRO 7840U w/ Radeon 780M Graphics GPU:Radeon(TM) 6550S OS:WIN11x64 10.0.22621.1778 Memory:CHB 32G 5600 SSD:AMD-RAID Array 1 SCSI Disk Device 476 GB" Fail Description:"BSOD:0X139(100%) PC:B000A600.MP0 FW:D1420000." Debug log:"PostCode: 0xB000A600 PSP: alive SMU: alive GFX_IMU_MSG_FLAGS = 0x80000019 (GFX clean) PwrMgtStatus: 0x48000000 CSTATEHISTCTLTOP: 2 LOWESTACPICSTATE: 1 No HWA. Please check windbg and get dump view" Associated Ticket No.:PLAT-134049

⋮



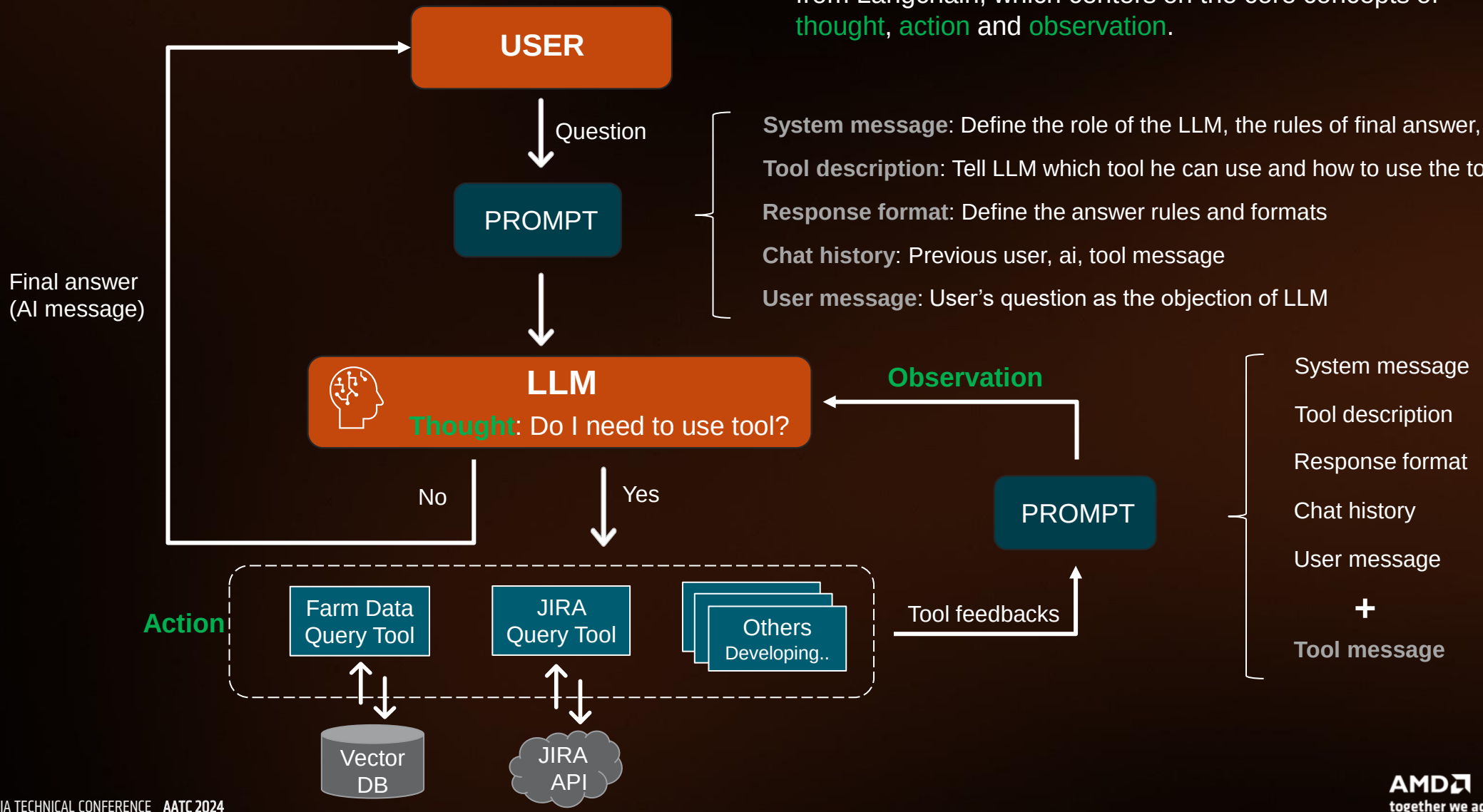
Improved FADA infrastructure

Per feedback analysis, gives an improved Fada infrastructure to provide more powerful and ease of use solution. New infrastructure introduced AI agent as the brain of Fada to support all kinds of user questions and intelligently assign to sub modules.



Approaching – AI Agent

- Agent's architecture is designed around the ReAct model from Langchain, which centers on the core concepts of thought, action and observation.



Approaching – AI Agent

Prompt & History

Question

Thought

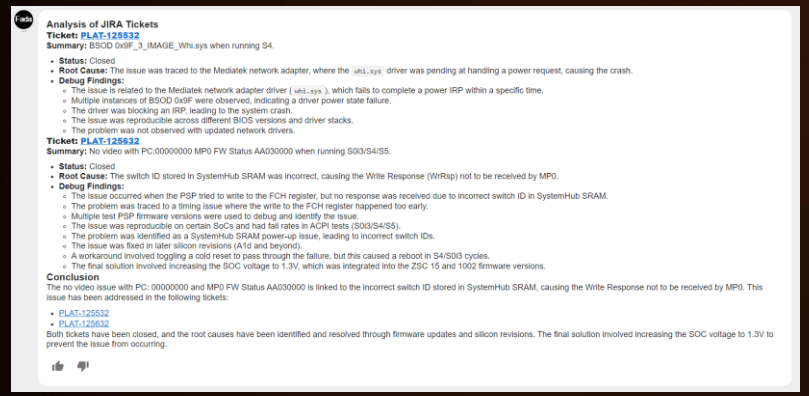
Action & Observation

Answer

```
system: Your name is Fada, you are a world class x86 PC debug expert, familiar with BIOS, Graphic driver, SMU, PSP, etc.
You are good at finding debug clue based on given history fail records.
You don't make up, you give analysis based on truth and knowledge.
Please make sure you complete the objective above with following rules:
1/ If use provide history record, you must cover all the records have same PC and MP0 status.
2/ Give related history record number, test bench, configuration and associate ticket number when user provided history records for analysis.
3/ Bold all the ticket number in your answer.
4/ Analyze the debug log and find clue.
5/ Provide the ticket hyperlink in conclusion, link address: https://ontrack-internal.amd.com/browse/PLAT-xxxxxx.
system: You have access to FarmInfoQueryTool, JiraQueryTool. You can only use 1 tool for 1 time before response to human. If the user's description is too
human: I have a no video issue with PC: 00000000 when farm test, please help me on the debug
ai: Based on the history records for PC: 00000000, there are multiple instances of the no video issue during farm tests. Here are the relevant records:
...
You can find more information on these tickets by following these links:
- [PLAT-125532](https://ontrack-internal.amd.com/browse/PLAT-125532)
- [PLAT-125632](https://ontrack-internal.amd.com/browse/PLAT-125632)
human: Search the JIRA to find clues.
```

```
Invoking: `JiraQueryTool` with `PLAT-125532 PLAT-125632`
```

```
JiraQueryTool query: PLAT-125532 PLAT-125632
['PLAT-125532', 'PLAT-125632']
PLAT-125532:  {'summary': '[PHX_FP7r2/FP7/FP8][SSV_Farm]BSOD 0x9F_3_IMAGE_Whi.sys when run S4.F/R:1/10U.', 'statu
PLAT-125632:  {'summary': '[PHX_FP7r2/FP7][SSV_Farm]No video with PC:00000000 MP0 FW Status AA030000 when run S0i
```



Analysis of JIRA Tickets

Ticket: PLAT-125532

Summary: BSOD 0x9F_3_IMAGE_Whi.sys when running S4.

- **Status:** Closed
- **Root Cause:** The issue was traced to the Mediatek network adapter, where the `whi.sys` driver was pending at handling a power request, causing the crash.
- **Debug Findings:**
  - The issue is related to the Mediatek network adapter driver (`whi.sys`), which fails to complete a power IRP within a specific time.
  - Multiple instances of BSOD 0x9F were observed, indicating a driver power state failure.
  - The driver was blocking an IRP, leading to the system crash.
  - The issue was reproducible across different BIOS versions and driver stacks.
  - The problem was not observed with updated network drivers.

Ticket: PLAT-125632

Summary: No video with PC:00000000 MP0 FW Status AA030000 when running S0i3/S4/S5.

- **Status:** Closed
- **Root Cause:** The switch ID stored in SystemHub SRAM was incorrect, causing the Write Response (WrRsp) not to be received by MP0.
- **Debug Findings:**
  - The issue occurred when the PSP tried to write to the FCH register, but no response was received due to incorrect switch ID in SystemHub SRAM.
  - The problem was traced to a timing issue where the write to the FCH register happened too early.
  - Multiple test PSP firmware versions were used to debug and identify the issue.
  - The issue was reproducible on certain SoCs and had fail rates in ACPI tests (S0i3/S4/S5).
  - The problem was identified as a SystemHub SRAM power-up issue, leading to incorrect switch IDs.
  - The issue was fixed in later silicon revisions (A1d and beyond).
  - A workaround involved toggling a cold reset to pass through the failure, but this caused a reboot in S4/S0i3 cycles.
  - The final solution involved increasing the SOC voltage to 1.3V, which was integrated into the ZSC 15 and 1002 firmware versions.

Conclusion

The no video issue with PC: 00000000 and MP0 FW Status AA030000 is linked to the incorrect switch ID stored in SystemHub SRAM, causing the Write Response not to be received by MP0. This issue has been addressed in the following tickets:

- PLAT-125532
- PLAT-125632

Both tickets have been closed, and the root causes have been identified and resolved through firmware updates and silicon revisions. The final solution involved increasing the SOC voltage to 1.3V to prevent the issue from occurring.

- In this case, user ask question about “No video” and got some records, then user ask Fada to search Jira to find more clue.
- The latest user question together with system message, chat history, tool descriptions are then sent to LLM.
- For OPENAI ‘s LLM, since it support function calling by API. Tool description could be separated with user & system message.
- For those LLMs don’t support function calling. Tool description and answer formats are prompted into user message, which consumes more token per chat.

- LLM accepts chat history, question and tool description then give feedback to use “JiraQueryTool”

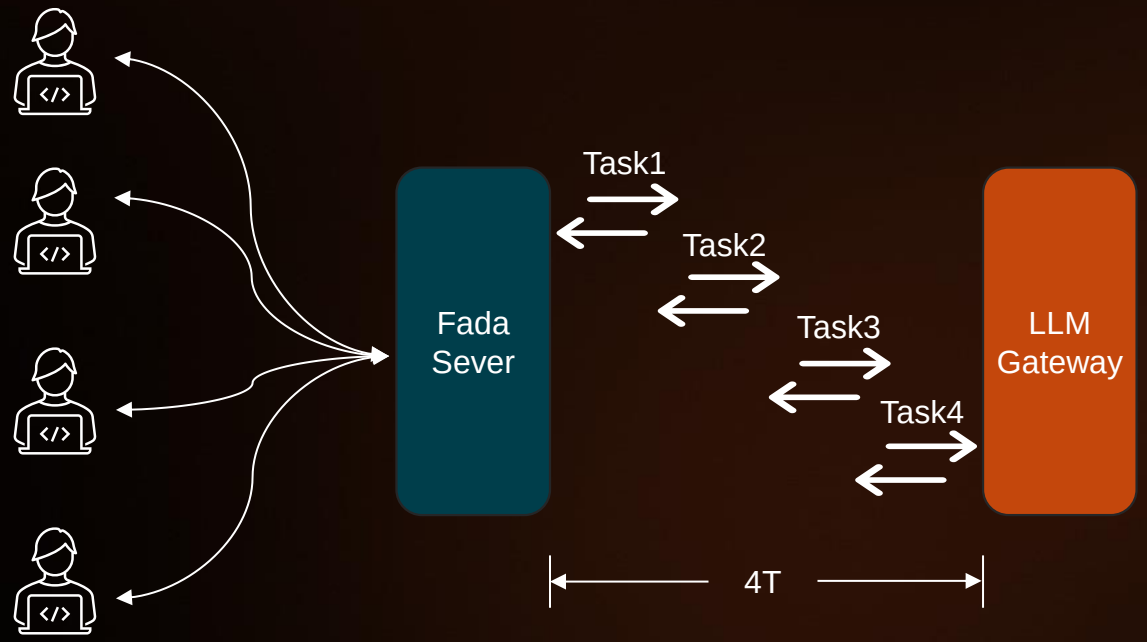
- Agent got LLM’s tool calling message then execute JiraQueryTool to Jira info
- Jira info (title, status, root cause, comments, etc.) is formatted as JSON string then sent back to LLM
- All logic handled by Langchain’s tool calling agent

- LLM got the Jira info together with previous user question. Then decided to response to human.
- The tool calling results are also stored as chat history and will be passed to LLM in next question. So, user could continue asking questions about Jira info while agent is no need to search again.

Approaching – asynchronous design for performance

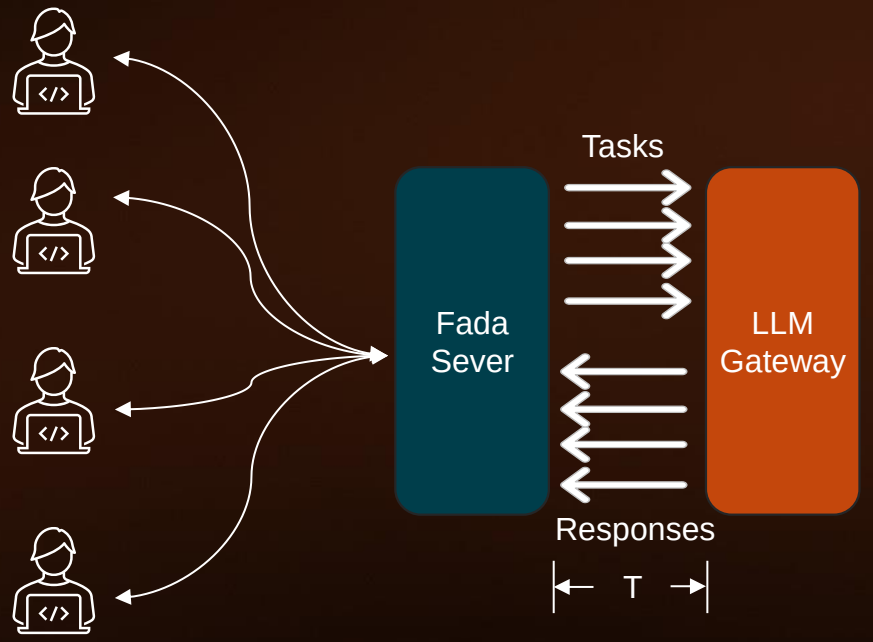
Problem

Fada response slows down when accessed simultaneously by multiple users because the server processes requests in a sequential manner.



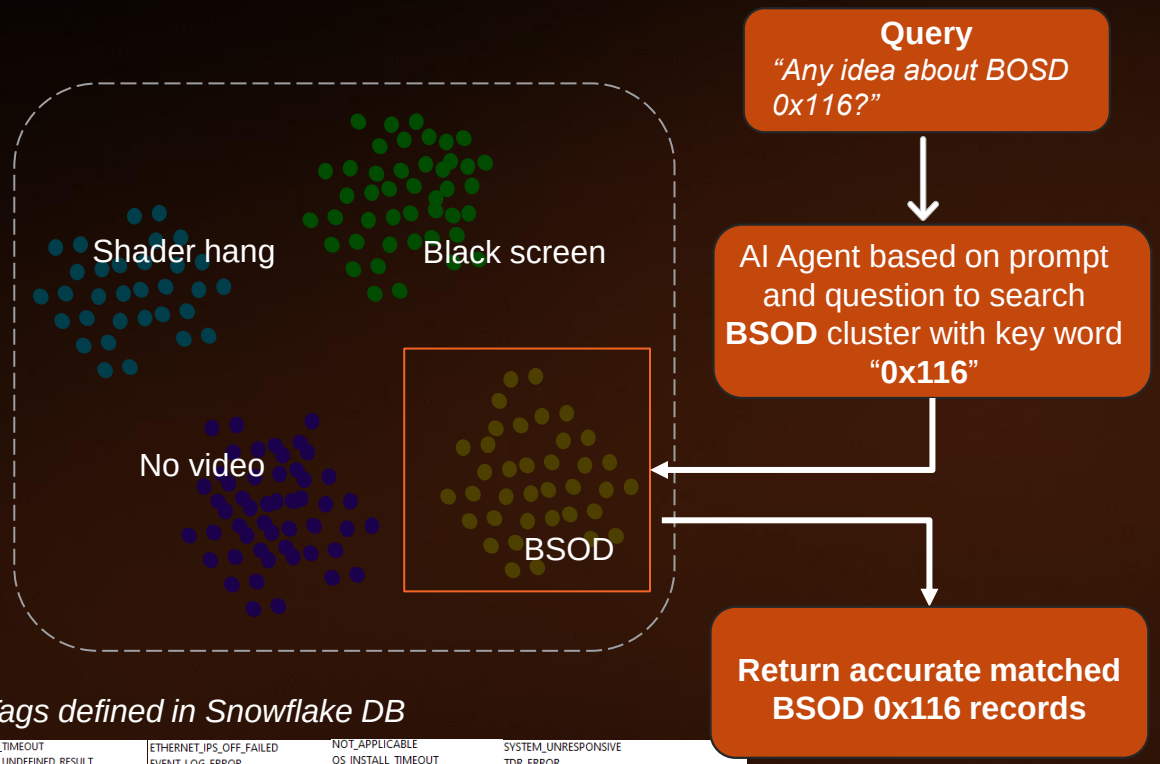
Solution

Utilizing coroutines for asynchronous request handling to decrease the IO wait time in LLM.



Approaching – Clustering for Accuracy and Performance

- Cluster the database by tags or user-defined clustering_id, like issue description “black screen”, “no video”, “shader hang”, “BSOD”, etc.
- When conducting a search, only need to find the cluster closest to the query vector and then perform the search within this cluster.
- Or specify to search multiple clusters. When doing this way, we need to figure out the balance point between Accuracy and Performance. As searching more multiple clusters means more searching time but higher accuracy.
- AI agent will base on user description to determine to performing text search or vector search on which cluster(s) with recognized information.

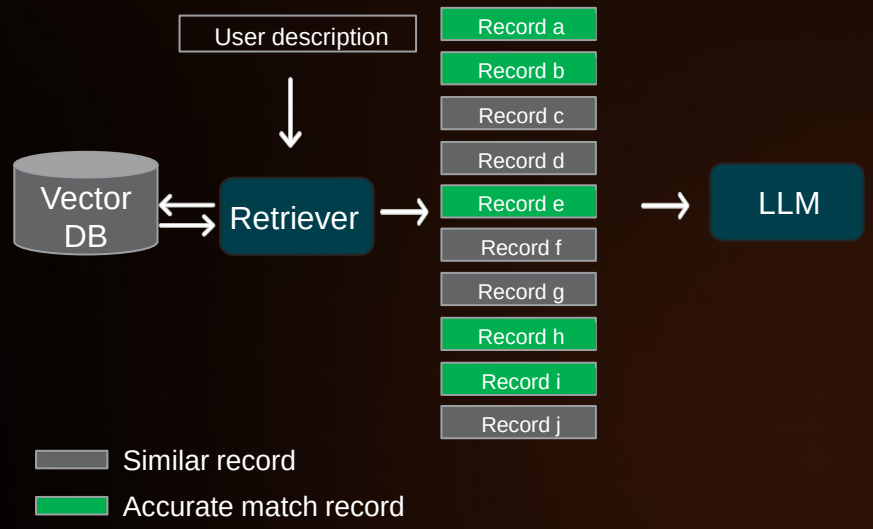


Tags defined in Snowflake DB

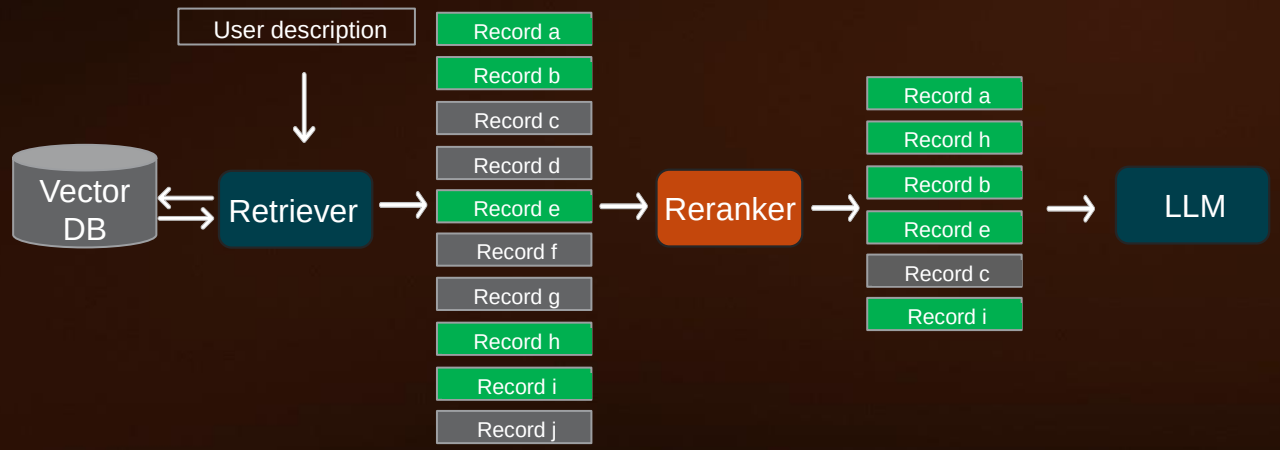
ADS_TIMEOUT	ETHERNET_IPS_OFF_FAILED	NOT_APPLICABLE	SYSTEM_UNRESPONSIVE
ADS_UNDEFINED_RESULT	EVENT_LOG_ERROR	OS_INSTALL_TIMEOUT	TDR_ERROR
APP_CRASH	EXCEPTION	PAPI_FAILED	THERMAL_CHAMBER_FAILED
APP_CREDENTIALS_UNAVAILABLE	EXECUTOR_INIT_FAILED	PARAMETER_ERROR	UNABLE_TO_SET_TEST_CONDITION
APP_INSTALL_ERROR	EXECUTOR_UNRESPONSIVE	Pass	UNEXPECTED_POWER_STATE
APP_LOG_ERROR	FUSE_FAILED	PCIE_SPEED_CHANGE	UNEXPECTED_REBOOT
APP_NOT_INSTALLED	HARD_LOCK	PDU_START_FAILED	UNKNOWN_ERROR
APP_RESULT_ERROR	HDT_INIT_FAILED	PSP_UNLOCK_FAILED	UNMAPPED_EXIT_CODE
APP_STARTUP_ERROR	HELPER_SCRIPT_ERROR	PWR_STATE_ENTRY_FAIL	UNSAFE_STORAGE_SHUTDOWN
ARX_TWEAK_FAILED	HYPERVISOR_ERROR	PWR_STATE_EXIT_FAIL	WAIT_FOR_SUT_REBOOT_AND_RESTART_WRAPPER
AUTOMATION_ERROR	INSTALL_SCRIPT_MISSING	PXE_BOOT_FAIL	WARNINGS_DETECTED
BCT_EXECUTION_ERROR	INSUFFICIENT_DISK_SPACE	PXE_BOOT_TIMEOUT	WHEA_ERROR
BIOS_FLASH_ERROR	INSUFFICIENT_MEMORY	REBOOT_TIMEOUT	WORKLOAD_DETECTED_TIMEOUT
BSOD	INVALID_CONFIG	REQUEST_HOST_REBOOT_SUT_AND_	WORKLOAD_TIMEOUT
COMMAND_TIMEOUT	INVALID_SETUP	REQUEST_HOST_REBOOT_SUT_AND_	
CORE_FEATURE_CHECK	MCA_ERROR	SCHEMA_VALIDATION_FAILED	
DAQ_START_FAILED	MISSING_DAQ_LOG	SECURE_UNLOCK_FAILED	
DCH_ERROR	MISSING_RESOURCE	STDERR_DETECTED	
DEVICE_MISCOMPARE	MISSING_WORKLOAD	STOP	
DOWNLOAD_ERROR	MULTIPLE_FAILURES	SUT_IP_ADDRESS_CHANGED	
DRIVER_INSTALL_ERROR			

Approaching - Rerank

- To further increasing RAG retrieval accuracy, we also introduced a re-rank model (Reranker) right after retrieval function. Reranker is essentially an opensource lite-weight language model can run locally on server, it screen and score the retrieval result and feed to LLM with a renew order.
- With Reranker, we can feed more accurate record to LLM. This can also help saving the LLM tokens.
- However, Reranker will also increasing the reaction time of Fada, we allow user to determine enable this feature or not.



E.g., without a Reranker LLM only got 5/10 accurate records



with a Reranker LLM get 5/6 accurate records, 40% token saved

Result – Answer satisfaction improvement status

Via solutions implementation in 2nd stage development (still ongoing), the satisfaction has below status update

- 48% answer get satisfactory feedback → 80%[1]
- 21% is IT limitation and issue →
  - 15% timeout issue is mitigated via providing LLM change options to Gemini, IT is still working on OpenAI timeout issue fix.
  - 5% Rate limit will be fixed with production version API permission.
- 15% categorized as unsupported feature →
  - 6% JIRA access requirement has been satisfied with new feature enablement.
  - 9% BI function under development.
- 10% can be improved via user usage →
  - Improved with more intelligent AI agent. (auto call sub tool, query user for more info)
  - Local knowledge query function underdevelopment .
- 6% not accurate - need to improve →
  - Partially improved with data clustering and rerank. Hybrid retrieval function under development

1. 80% is calculated based on below improvement item implement status

Result and conclusion

- The development and implementation of Fada, a high-efficiency farm issue debug chatbot, have demonstrated significant advancements in leveraging AI for the validation and debugging processes. The tool's integration of an optimized Retrieval-Augmented Generation (RAG) structure with a powerful LLM AI agent has yielded impressive results in terms of accuracy, efficiency, and user satisfaction.

- **Results**
  - 1. Increased Efficiency:
    - 48% of responses received satisfactory feedback** from users, (improved to 80% via action items) indicating a substantial improvement in debugging efficiency and effectiveness.
    - The streamlined data retrieval process, which eliminates the need for multiple retrieval calls, has reduced both time and computational resources.
  - 2. Enhanced Accuracy:
    - The implementation of embedding and vector similarity search has improved the accuracy of retrieved historical records.
    - The use of advanced LLM models, such as GPT-4 turbo with a 128K tokens context window, has ensured that the AI provides precise and contextually relevant suggestions.
  - 3. User Feedback and Iteration:
    - Based on over 100 pieces of feedback collected during the Alpha test phase, 48% of responses were satisfactory, 15% were categorized as unsupported features, 10% could be improved through user usage, 21% faced IT limitations, and 6% were not accurate.
    - Iterative improvements are continuously being made based on user feedback to enhance Fada’s performance and address identified limitations.

- **Conclusion :** Fada provides a robust solution for farm issue debugging by combining historical data analysis with state-of-the-art AI technology. This methodology not only enhances the efficiency and accuracy of the validation process but also serves as a model for future AI-based tools within the company.
- In summary, Fada stands as a testament to the power of AI in transforming validation and debugging processes, offering significant benefits to projects, products, and the company as a whole. Its ongoing development and optimization promise even greater efficiencies and innovations in the future.

Benefit and Relevance for AMD

- **For the Project:**
  - Reduced Debug Time: Accelerates the identification and resolution of issues, significantly reducing debug time.
  - Automated Guidance: Provides junior testers with automated, expert-level debugging insights, improving the efficiency of the validation team.
  - Enhanced Collaboration: Facilitates better communication and collaboration among multiple teams by providing centralized and easily accessible debug information.

- **For the Product:**
  - Improved Quality: Ensures higher product quality by identifying and resolving issues more efficiently during the validation phase.
  - Timely Launches: Accelerates convergence on farm issues, supporting timely product launches.
  - Data-Driven Insights: Utilizes vast historical data to provide insights and predictions, leading to more robust and reliable products.

- **For the Company:**
  - Cost Efficiency: Reduces costs associated with prolonged debugging processes and late-stage issue resolutions.
  - Leverage AI Technology: Demonstrates the AMD's commitment to leveraging AI technology to improve operational efficiency.
  - Competitive Edge: Enhances the AMD's competitive position by ensuring faster time-to-market and superior product quality.

Next Step Plan (Possible Opportunities)

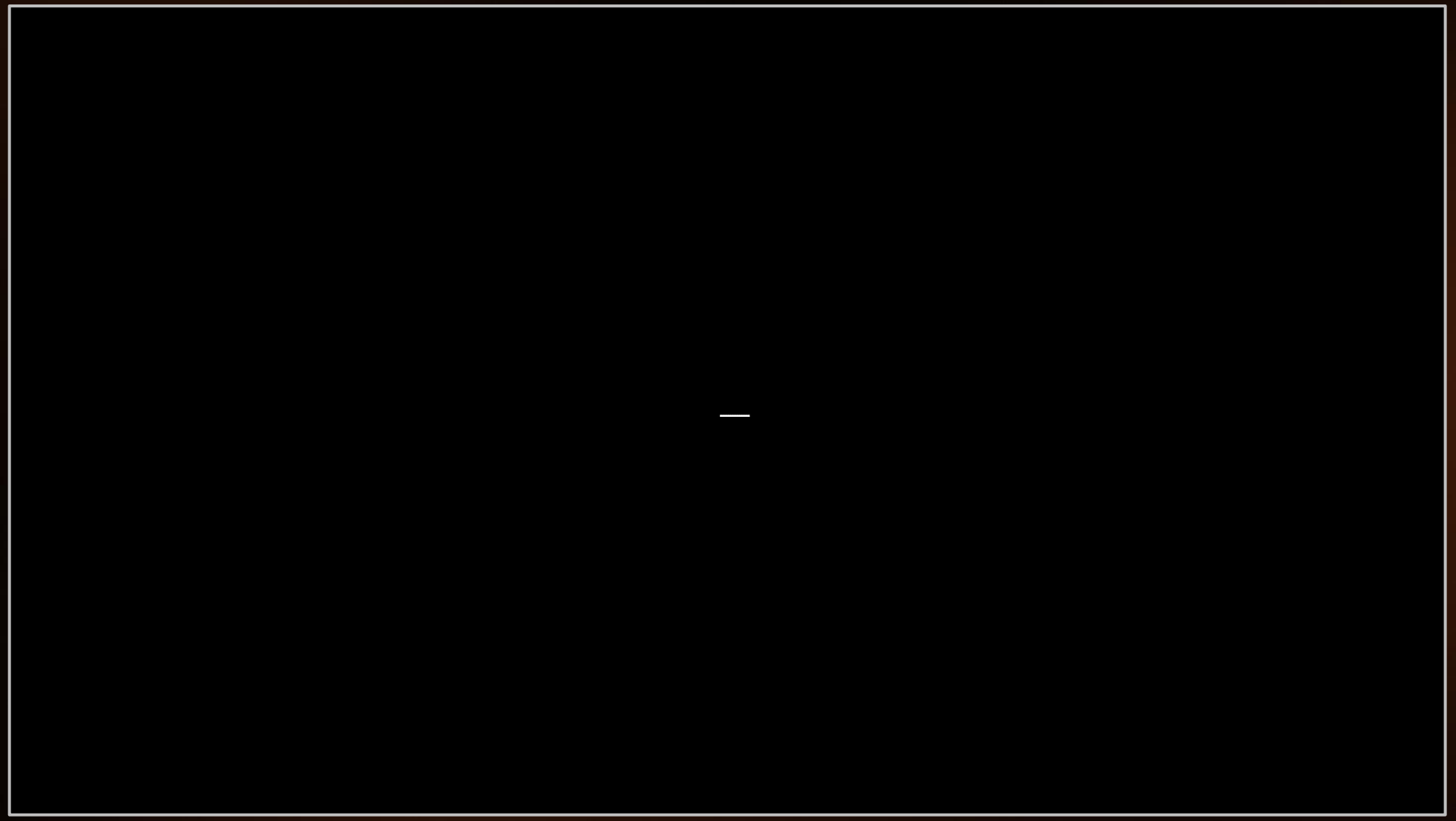
- **Expansion of Fada's Capabilities:**
  - BI Module Development: Incorporate Business Intelligence (BI) modules to provide detailed analytical reports and insights.
  - Local Knowledge Integration: Enhance local knowledge query functionalities to improve AI understanding of AMD-specific terms and processes.

- **Optimization and Testing:**
  - Performance Optimization: Implement asynchronous request handling to improve system responsiveness, especially under heavy load.
  - User Feedback Iteration: Continuously gather and incorporate user feedback to refine and enhance Fada's functionalities.

- **Wider Deployment and Integration:**
  - Broadening Scope: Extend Fada's application to other areas within the company, such as software development and customer support.
  - Integration with Other Systems: Integrate with other internal systems like JIRA and Snowflake to provide a more comprehensive and unified debugging solution.

- **Exploring New Opportunities:**
  - AI Agent Development: Develop AI agents for other use cases within the company, leveraging the successful model of Fada.
  - External Market Potential: Explore opportunities to offer Fada or similar AI solutions to external clients, potentially creating new revenue streams.

Demo




AMD