

AATC 2024

PD Agent

AMD ASIA
TECHNICAL
CONFERENCE

AMD 
together we advance_innovation

Can't share any authors related information here or anywhere in the paper.

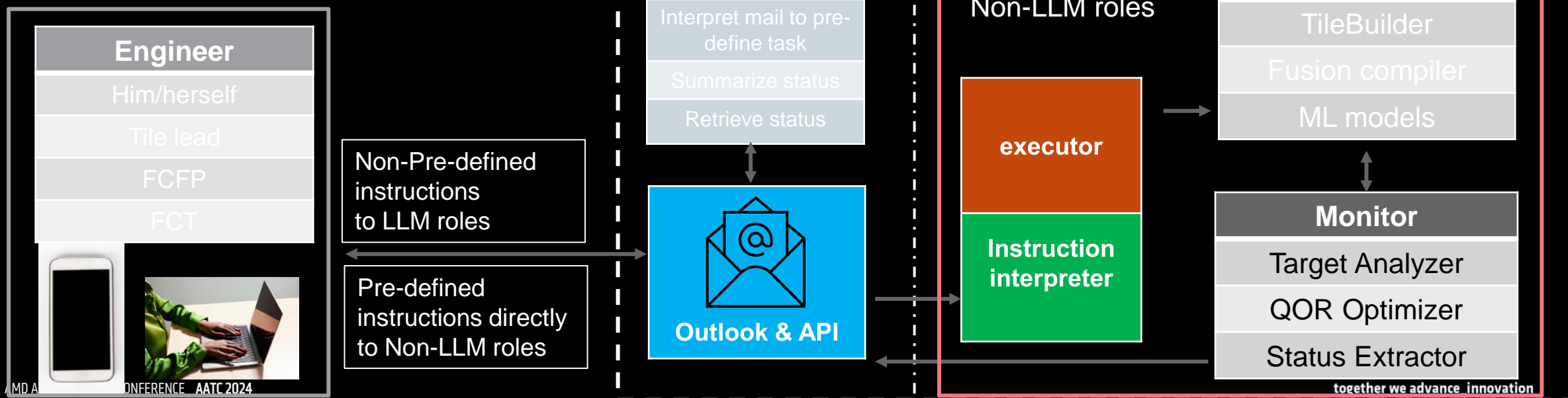
Abstract

- During IC design work, various scripts are developed to minimize manual tasks. However, the overall human resource savings are still limited due to the necessity of human involvement in email communication and script execution.
- This article proposes an unmanned technology - the PD agent. This agent can help physical design engineers save resources by completing the entire task cycle, from receiving and parsing emails to running and monitoring jobs. This includes starting TB runs, updating flows, rerunning targets, catching JIRA updates, releasing data, and monitoring runs in a Linux environment based on scripts/LLM. This technology can also be extended to other teams such as FEI, CAD, and DV for regression. It unifies communication through the Outlook API for human-to-agent and agent-to-agent interactions.
- With this technology, engineers can fully focus on complex tasks and be freed from simple and tedious jobs. Meanwhile, most requests to engineers will be responded to within 5 minutes to 1 hour, without delays due to engineers being offline or occupied with other tasks.
- Resources for easy tiles can be completely released in NLA~NLC stages. The need for a daily ECO FCT run owner can also be eliminated.
- **Keywords:** Agent, outlook API, LLM, unmanned technology.

Methodology

- PD agent consists of two parts, windows and Linux.
 - The Windows part includes Outlook & API, which receive, extract, and send emails from human engineers. The server accepts access from Linux non-LLM roles. The LLM translates non-pre-defined instructions into pre-defined instructions for non-LLM roles, or retrieves and summarizes the status on Linux, then sends it to engineers.
 - The Linux part accesses the email records in Windows through HTTP protocols. It interprets the pre-defined instructions in the records into scripts that are executed in Linux. It performs and monitors various PD jobs and returns the status to engineers via email.

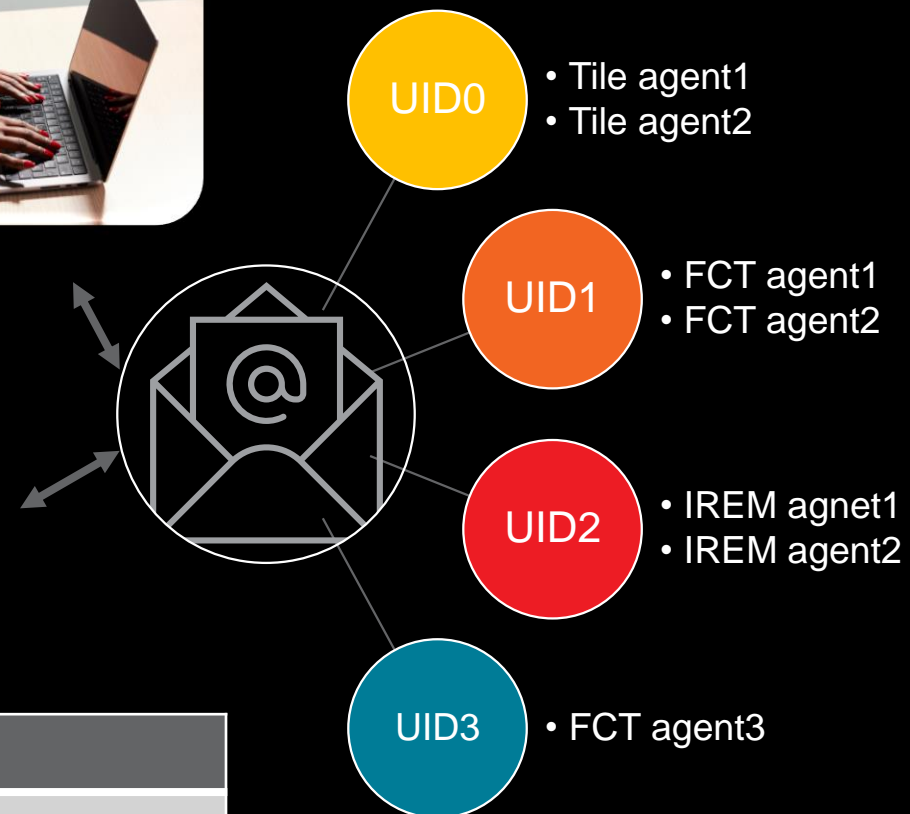
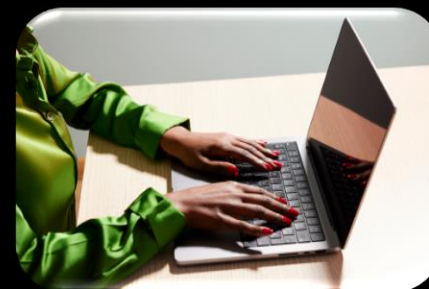
- Human —————
- Agent - - - - -



Role Definition and Instantiation

- In the context of PD agents, a role refers to a class that can perform certain functions, such as tile PNR or FCT. This role is instantiated with a name when it is invoked via email (TO0/TO1, FCT0/FCT1).
- Multiple roles share a single communication interface, which means that a single desktop can support multiple agents.
- The definition of a role allows developers to implement roles with different functions (like tile PNR/FCT), different models (such as program/LLM), different frameworks (like LangChain/AutoGPT), and different platforms (like Linux/windows). This provides virtually limitless possibilities.

Role	Name	Framework	Function	Platform
Tile owner	TO0	NA	Tile PNR	Linux
FCT	FCT0	NA	FCT	Linux
Manager	GPT0	GPT4-32k	Job assignment knowledge query	Windows
Tile owner	LC0	Langchain/GPT	Tile PNR	Linux



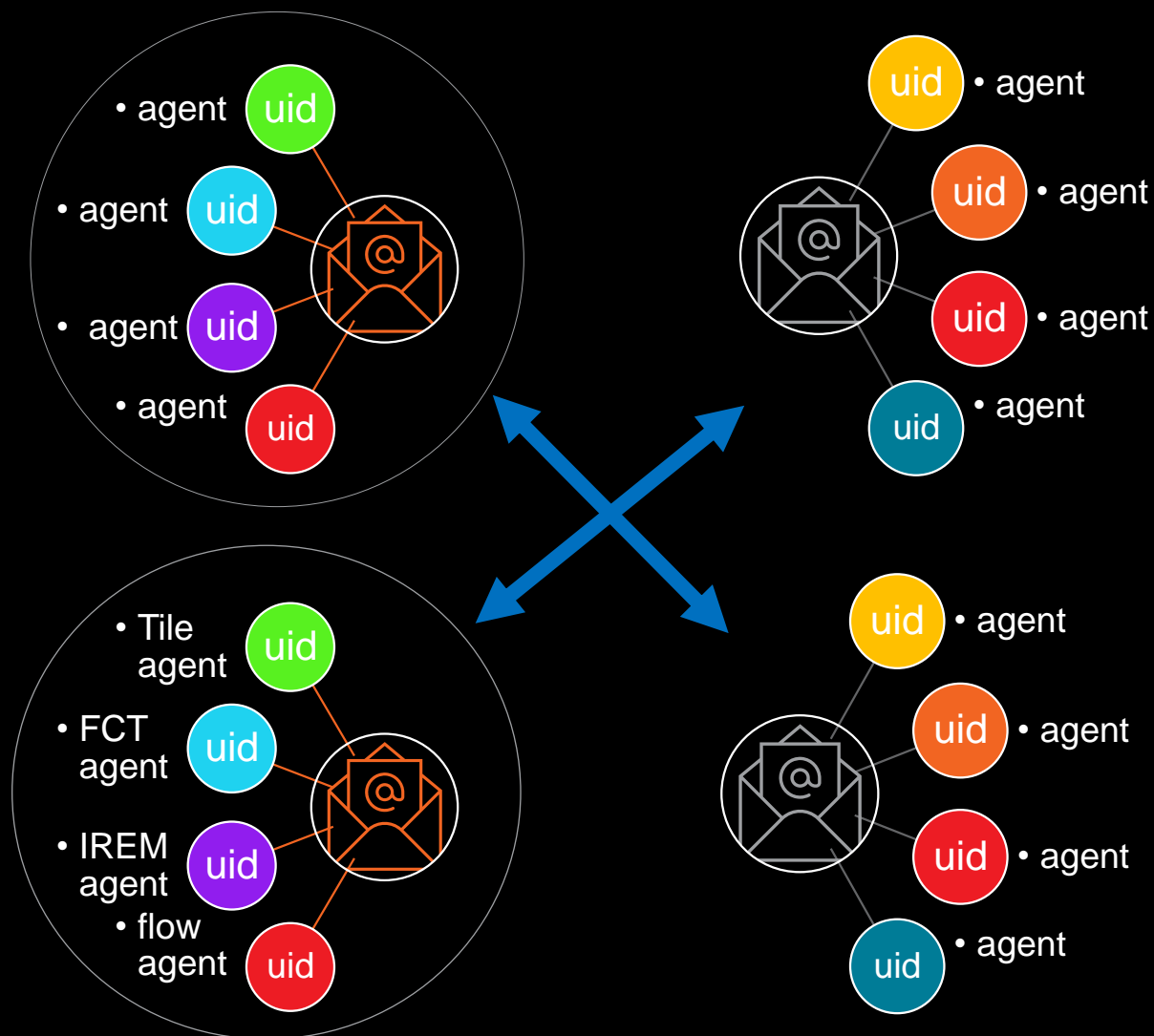
Communication Interface(Outlook API)

➤ Decoupling and Distributed

- Since the PD agent uses Outlook as an interface for human interaction or other types of agents, its boundary is very clear, allowing for decoupling. Each functional team can develop their agent independently, and different agents can communicate with each other.

➤ Traceability

- Outlook acts as an electronic contract, permanently recording and proving all requests and responses. Therefore, any important task can be assigned through Outlook.



Keyword-based Instruction interpreter

- As PD agents are role-based, they natively support multiple instruction interpretations from natural language to instructions that can be executed in Linux or Windows.
- A keyword-based interpreter uses a lookup table algorithm to interpret the keywords in natural language.
- **Pros:** The advantage of this role is that it is 100% reproducible and explainable, which suits users who are very familiar with the overall instructions that the PD agent supports.
- **Cons:** If there is a typo in the user's input, it will fail to execute the task. For new users, it takes time to become familiar with the pre-defined instructions.

keyword.csv		argument.csv	
start	Kick off	df_tcdxa_t	tile
run		TILES_TO_RUN	params

instruction.csv	
Could you help start a run for df_tcdxa_t?	start_tb.csh \$tile \$params
TILES_TO_RUN = df_tcdxb_t	

Algorithm 1. One-hot Keyword Based Lookup Table Instruction Compiler

Procedure Func (mail_csv, keyword_csv, argument_csv, instruction_csv)

- ForEach** keyword **in** keyword_csv
- Keyword_dic[keyword] = [0, 0,...,1,...,0,0]
- ForEach** line **in** argument_csv
- data, argument = line
- argument_dict[data] = argument
- ForEach** instruction **in** instruction_csv
- ForEach** word **in** instruction
- If** word **in** Keyword_dic
- Instruction += Keyword_dic[keyword]
- Instruction_dic[instruction] = execution.csh
- ForEach** task **in** mail_csv
- ForEach** line **in** task
- ForEach** word **in** line
- If** word **in** keyword
- Instruction += Keyword_dic[word]
- If** word **in** argument_dic:
- argument_info[argument] = word
- If** instruction **in** Instruction_dic
- execution_csh = execution.csh
- Full_execution_csh = execution.csh argument_info[argument]

LLM-based Instruction interpreter

- The LLM-based instruction interpreter deploys the GPT4 gateway API, reads all the pre-defined instructions, and annotates them with descriptions.
- **Pros:** Users don't need to input instructions that exactly match the pre-defined instructions or even completely mismatch the instructions, as the LLM has an incredible generalization ability. In some cases, it can even infer the instructions implied in new tasks.
- **Cons:** The output of LLM is non-reproducible or inexplicable in some cases.

OpenAI API

```
SERVER = "https://llm-api.amd.com/azure"
HEADERS = {"Ocp-Apim-Subscription-Key": "your_key"}

def api_call(endpoint_name: str, body: dict, deployment_id: str):
    response = requests.post(
        url=f"{SERVER}/engines/{deployment_id}/{endpoint_name}",
        json=body,
        | headers=HEADERS)
    response.raise_for_status()
    return response.json()
```

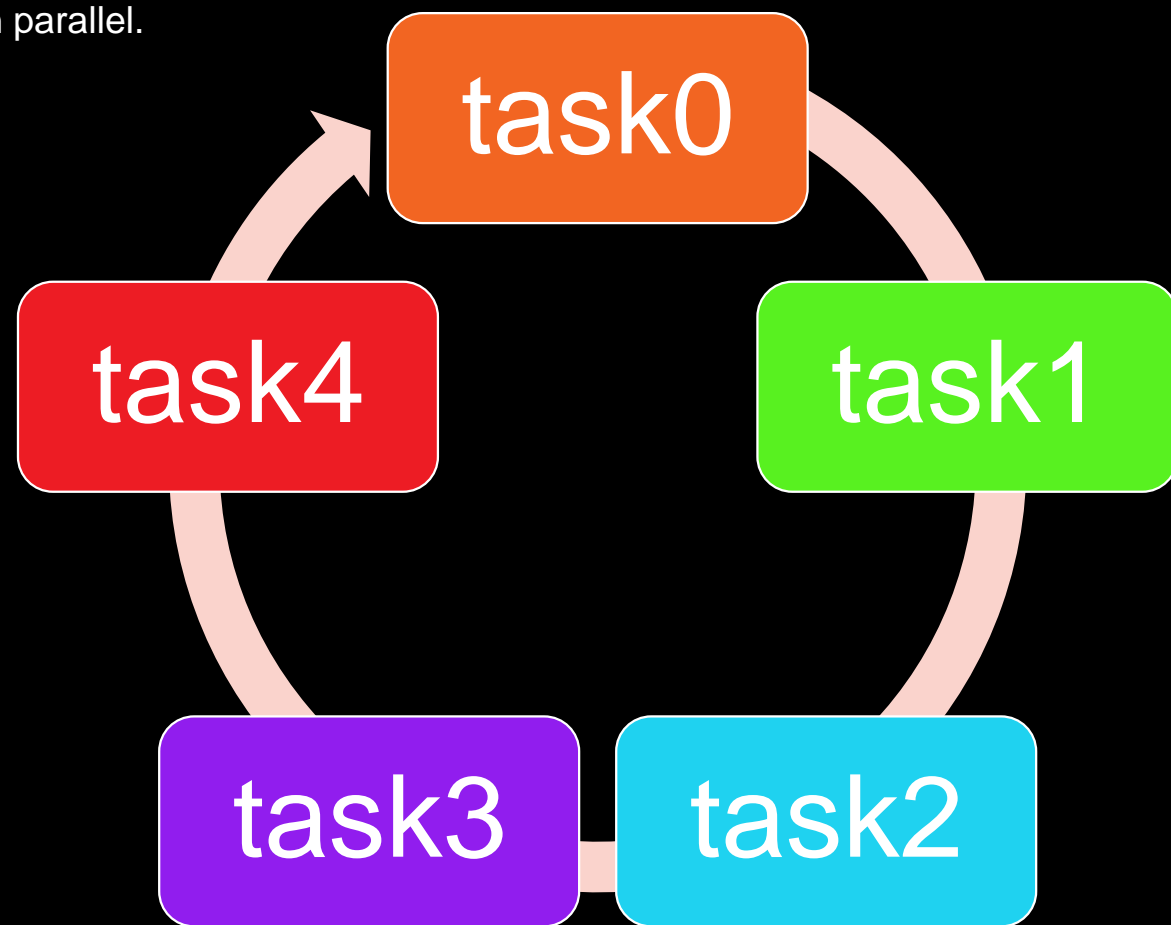
Context for GPT

description	PD agent instruction
branch	branch from \$target \$table with following params for \$tile \$run_dir
branch	TileBuilderBranch from \$target \$table with following params for \$tile \$run_dir
branch	TileBuilderBranch from \$target \$table for \$tile \$run_dir
check log	check your log logs/FxFpPlaceMacros.log.gz
open nlib	please open data/setup.nlib to check if you take this update correctly .
check log	Please check your log . make sure your use correct lib.
sync file	please sync flow to get this update
report tile status	what is the status for \$tile

Executor

- The Executor periodically iterates through each task in the CSV, launching instructions (in the form of *.csh files) from the interpreter if the status is empty (not executed).
- Each instruction is assumed to be independent, so they can be run in parallel.

Task.csv			
Time	mailBody	Instruction	Status
2023-5-20 10:39:28.30	Could you start a new run for df_tcdxa_t?	Start_new_run.csh df_tcdxa_t	running
2023-5-20 11:39:28.30	Could you send the timing status for df_tcdxa_t?	Report_timing_stat us df_tcdxa_t	finished
...



Monitor

- The PD agent has a monitor that can complete internal tasks. These tasks are not triggered by email, but by real-time status. The monitor includes the following classes:
 - Target analyzer, this class checks the target logs and takes correlated actions.
 - JIR tracer, this class reads the JIRA in the mailbox, extracts the solution, and applies it to each run.
 - QOR optimizer, this class extracts the current run status and launches the next run with the correlated recipe.

Target analyzer

issue	action
Segmentation fault encountered Stack trace for crashing thread	switch_version.csh
No such file sdc	report.csh fct
No such file or directory	report.csh flowLead
Cannot find tech layer	report.csh flowLead:librarian
Can't find in data/flow.xml	make.csh
Distributed run failed. error code: Operation cancelled	retrace.csh

JIR tracer

Zhang, Cinney updated an issue

Chen, Dongsheng closed an issue as **Fixed**

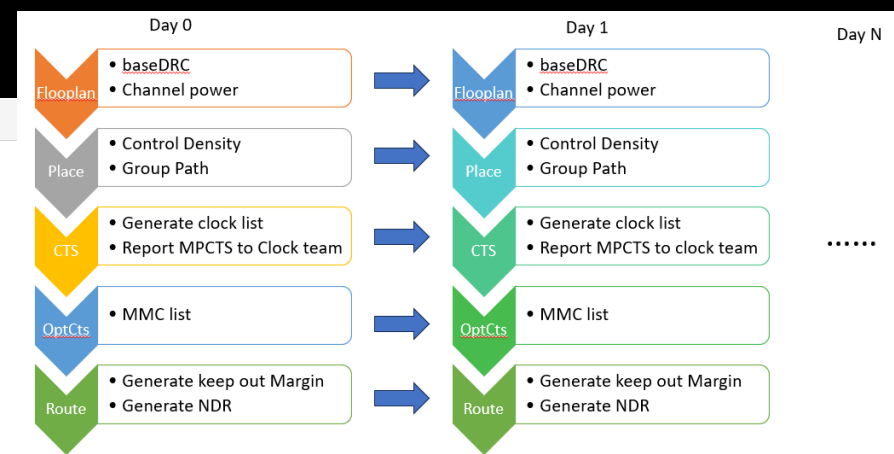
Zhang, Cinney assigned an issue to Zhang, Cinney

Westelk / DEWESTELK-159

GenRouteFill fail

Issue Type: Defect
Severity: normal
Status: Opened
Assignee: Zhang, Cinney
Reporter: Xie, Ruochen
Issue Category: Tile Flow [All; Floorplan; Place; CTS; Route & DFM]
Defect Type: Functional
Tech Bench: N/A
Variant: [N/A]
Issue Description:
FxGenRouteFill fail after switching tool version. Could you help to check the issue?
ICV_MODULE = icvalidator/2022.12-SP3

QOR optimizer



Scenario -- Kick-off/Monitor Tile Run/Update Flow

Mail of start new run from Flow lead

Hi All ,

NLB_2nd netlist and def is ready , please start a new TB for this milestone .
SDC is not ready , FCT team will update sdc at 2024/01/09.

Setup project MI350AID:

```
source ./tools/aticad/1.0/src/sysadmin/cpd.cshrc
setenv FAMILY supra
setprj mi350aid
TileBuilderStart --params override.params --controls override.controls | & tee tbs.log
```

override.params:

```
TILES_TO_RUN = vcn_tb_0_t (Tile owner can modify)
NICKNAME      = test(Tile owner can modify)
```

```
TILEBUILDER_FREEZEFLOWRELEASE = WEK_PD_FLOW_NLB_1st
TILEBUILDER_FREEZESTACKDATARELEASE = WEK_PD_DATA_NLB_1st
TILEBUILDER_SEPARATE_STACK_PARAMS_LABEL = WEK_PD_DATA_NLB_1st
```

Mail from agent after execution

Hi

The TB run for dfx_dft_aid_t vcn_s_0_t df_tcdxgi_t mmhub_sub_t nbio_l1_pcie_x16_t has been started at:

tile	runDir
dfx_dft_aid_t	/proj/wek_pd_irem1/VTO0/dfx_dft_aid_t/240108124204608/main/pd/tiles/dfx_dft_aid_t_240108124204608
vcn_s_0_t	/proj/wek_pd_irem1/VTO0/vcn_s_0_t/240108124204608/main/pd/tiles/vcn_s_0_t_240108124204608

Rerun failed targets

```
# check /proj/cmb_pnr_vol21/vto7/cdci_iohub_df_mid_t/240415
## check failed target TsPlaceGenTSDBView
[1] 2503
# TsPlaceGenTSDBView rerun count > 2, not rerun to save it
## check failed target TsReRouteGenTSDBView
[2] 2557
# TsReRouteGenTSDBView rerun count > 2, not rerun to save it
# Rerun norun targets.
# check reported
# check reported
```

Skip finished target

```
# check /proj/cmb_pnr_vol21/vto7/secip_mp1_t/240507151724214_s
# skip finished target OaGenPlaceDb.
# skip finished target FxXtrcttyprc100cPlace.
# skip finished target FmEqvPlaceVsBase.
[1] 22470
[2] 22474
[3] 22478
# skip finished target FxPlace.
```

Mail of Updating flow from Flow lead

Hi all,

Flow has two updates based on //gpuE/mi350/aid/pd/tiles/params/supra/personality/mi350aid/a0/placenroute.params#298.

Impact tiles of update 1
Impact tile:
All tiles

1. Update DFP_TAPCELL_PITCH = 75.012.

How to get this update

- Skip all green targets, sync flow environment
source \$ALLTILES_DIR/scripts/TButility/skip_green_target.csh

Mail from agent after execution

Re:RE: [MI350 AID]NLC flow update

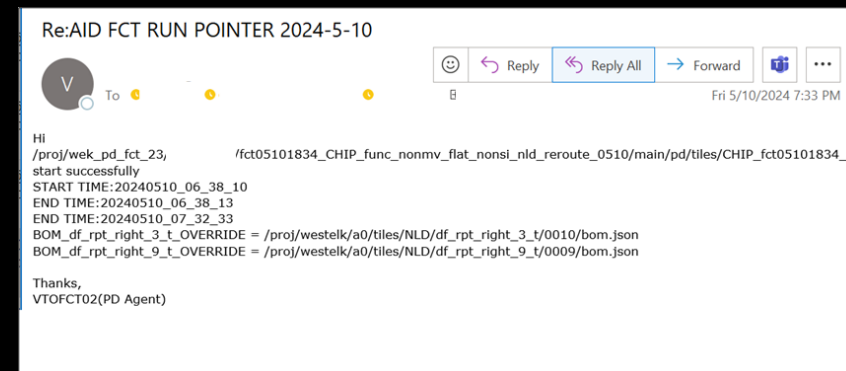
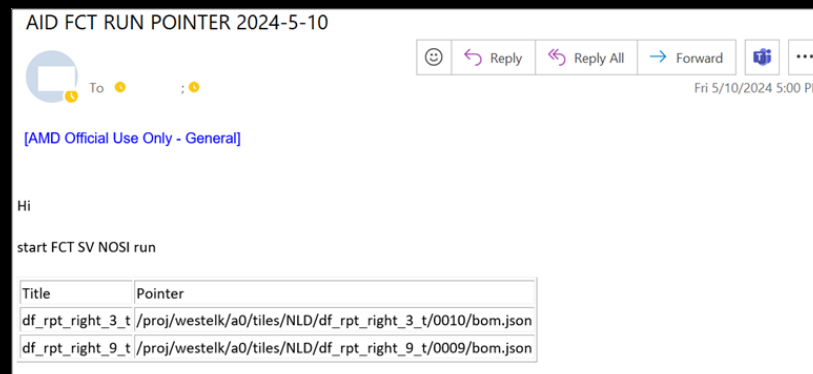
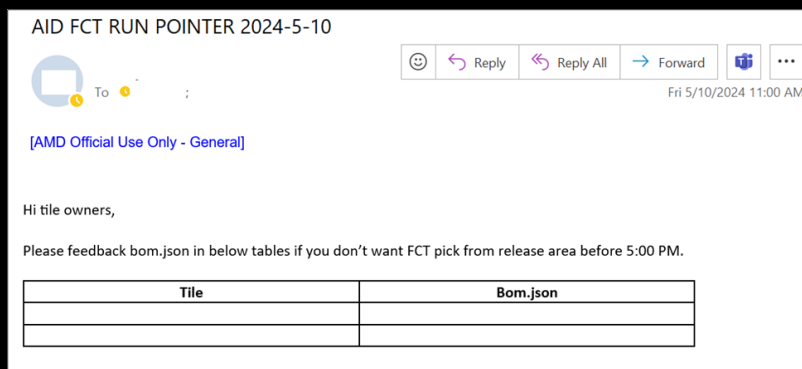
Hi

The environment has been updated:

Tile	runDir	Comment
dfx_dft_aid_t	/proj/wek_pd_irem1/VTO0/dfx_dft_aid_t/240312145351905/main/pd/tiles/dfx_dft_aid_t_240312145351905_sub0_TileBuilder_Mar12_0257_32294_GUI	SyncAllFlowEnvironment
dfx_dft_aid_t	/proj/wek_pd_irem1/VTO0/dfx_dft_aid_t/240110182914743/main/pd/tiles/dfx_dft_aid_t_240110182914743_TileBuilder_Jan10_0532_55980_GUI	SyncAllFlowEnvironment
vcn_s_0_t	/proj/wek_pd_irem2/VTO0/vcn_s_0_t/240312145351905/main/pd/tiles/vcn_s_0_t_240312145351905_sub0_TileBuilder_Mar12_0312_30545_GUI	SyncAllFlowEnvironment

Scenario -- Kick-off Daily ECO FCT Run

- The FCT agent provides an example of how communication occurs between agents and humans, as well as between different agents.
 - During the ECO stage, agent0 sends an email to all tile owners. The tile owners respond to the email by filling in the pointers in the table. At a specific moment, agent0 sends the command to start the run to agent1, along with the collected pointers.
 - Upon receiving the email from agent0, agent1 initiates the FCT run with the specific pointer. After the run has started, agent1 sends an email to the FCT team.
 - The FCT agent is capable of executing one full chip timing run and one partial full chip timing run daily.
- Mail of Pointer collection sent by agent0
- Mail of order of starting run sent by agent0
- Mail from agent1 after starting FCT run

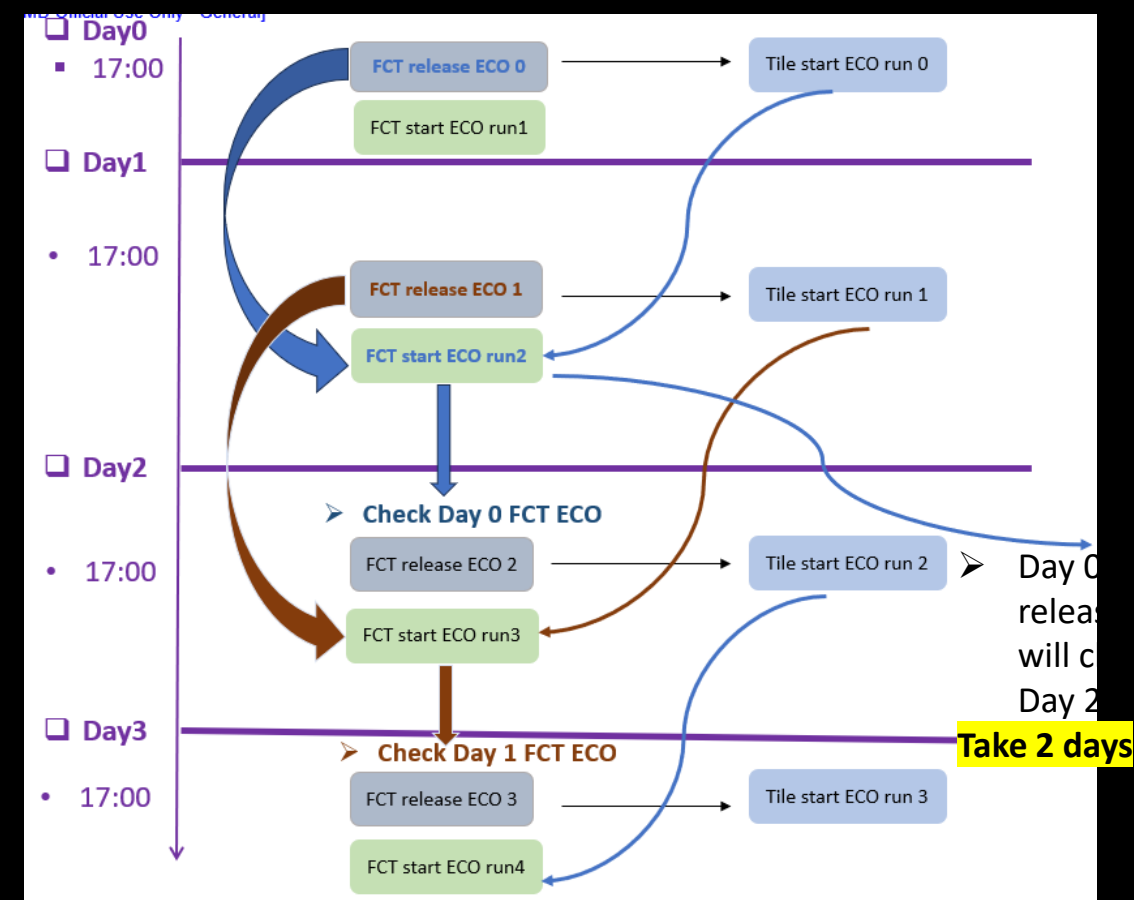


- The reason why two agents are separated is: agent0 represent FCT team to collect pointer, but it has no ER permission to read ER data, agent1 has ER permission but don't want to receive many tile owner mails.

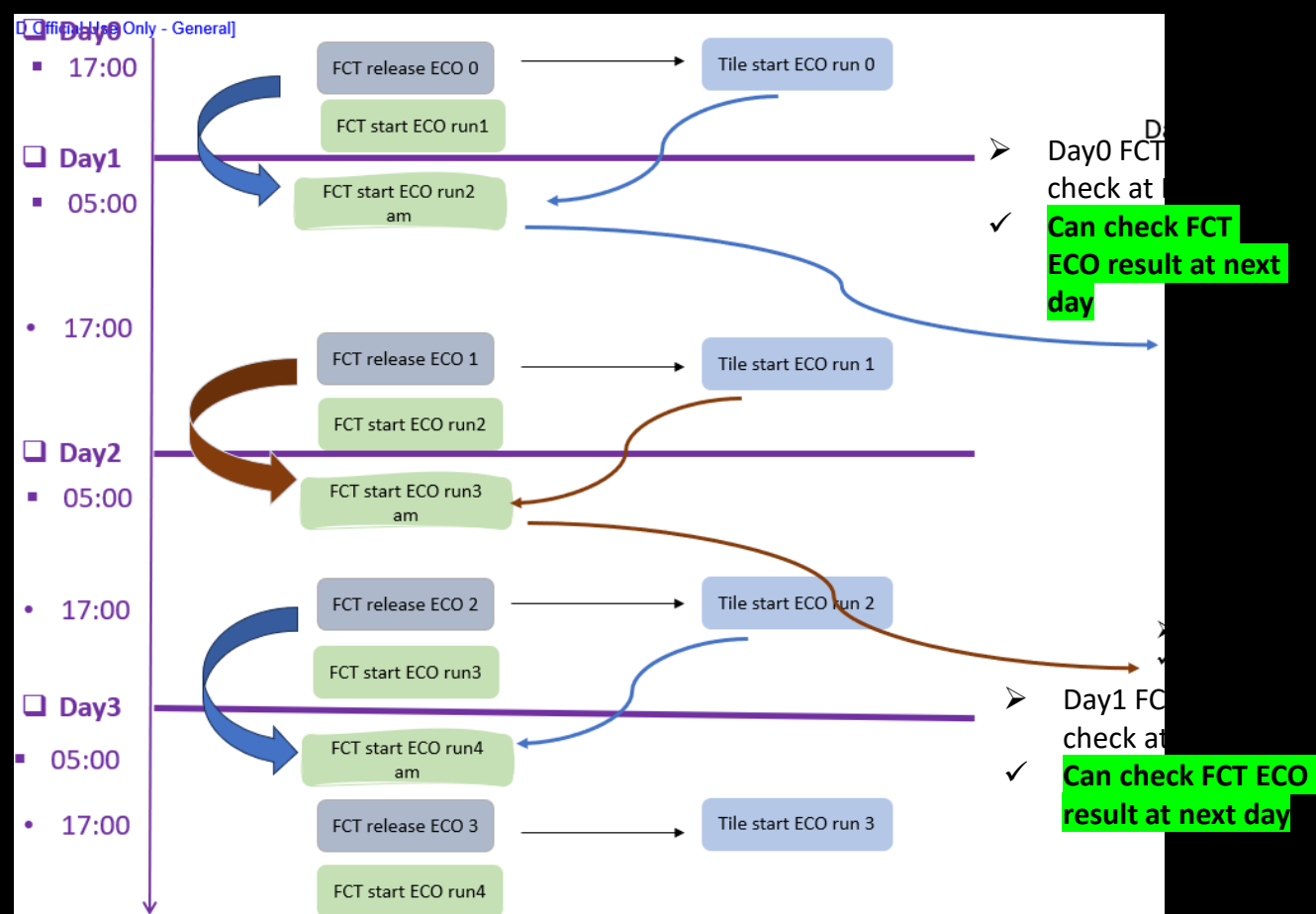
Scenario -- Kick-off Daily ECO FCT Run

- With agent, FCT can achieve daily ECO iteration by starting run at 5:00 AM, mean while saving FCT owner.

Regular ECO FCT^[1]



Agent ECO FCT^[1]



Result -- LLM capability

- LLM does a perfect job in table generation, issue assignment (e.g., SDC issue is assigned to the FCT team), branch run instruction interpretation, and run directory report. It is amazing that LLM can even identify the macro-overlap issue and propose the correct location.
- However, LLM does a poor job in timing report analysis and Macro placement.

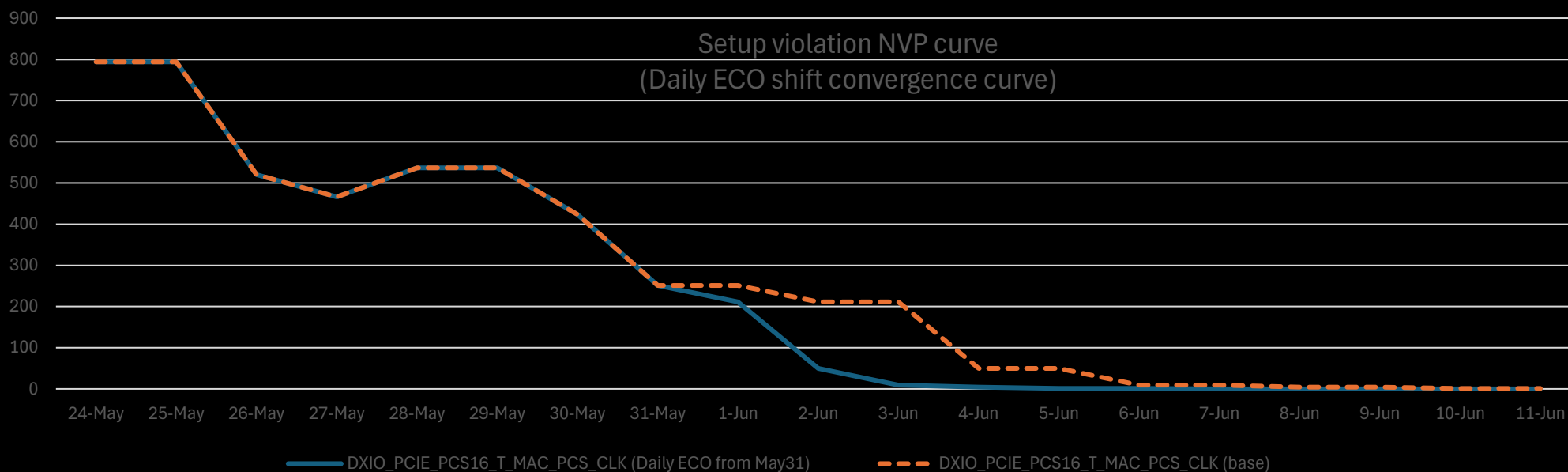


Result – Daily ECO Speedup

- Daily ECO will speed up setup fixing by 50%^[2]
- For hold fixing, FCT is using PtFixECO with 2 days runtime, no see improvement.^[2]

	Setup fixing (iterations per week)	Hold fixing (iterations per week)
Legacy ECO	3	2
Daily ECO	4.5 (speed up 50%)	2

- Daily ECO shift FCT convergence curve
 - DXIO_PCIE_PCS16_T_MAC_PCS_CLK enables Daily ECO from May31, the convergence curve is shift left by 5 days.



Result -- QOR comparison

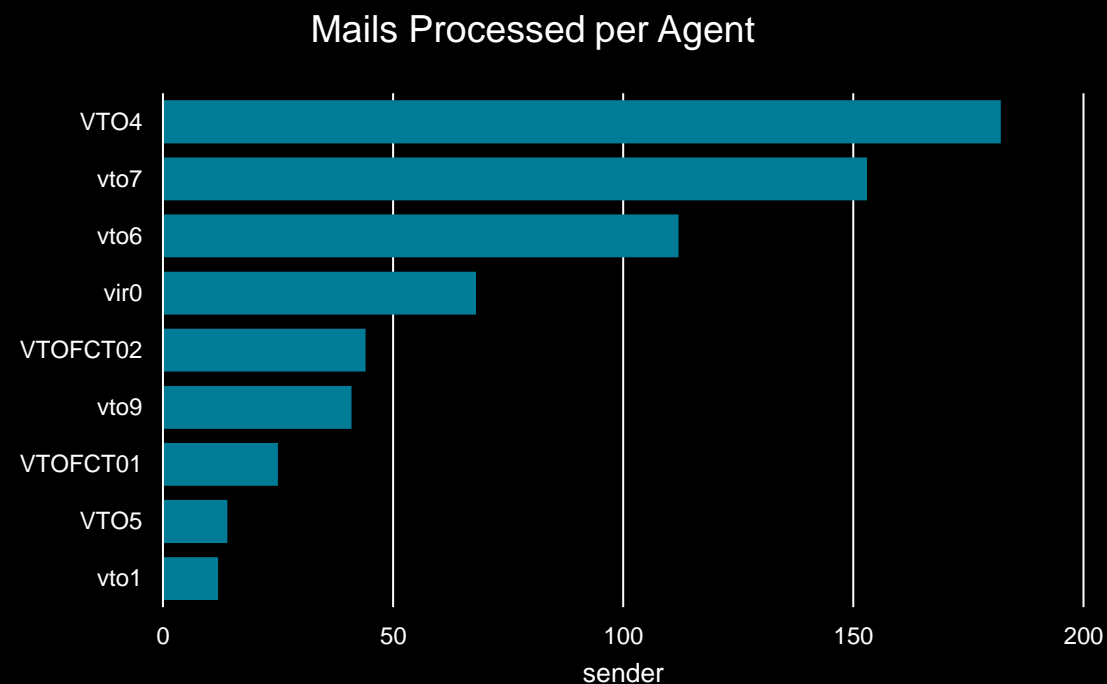
- The QOR running from Agent is comparable with human tile owner..

Run:	Status			Util			Pretiming	Place	OptCts	ReRoute	ReRoute DRC
	Syn	Timing	Routing	Init	Place	End	Stp: wns/tns/nvp Hld: wns/tns/nvp	WNS(ps)/TNS(ns)/NVP/Cong	WNS(ps)/TNS(ns)/NVP/Cong	WNS(ps)/TNS(ns)/NVP	DRC/Shorts
Tile owner run				58.9%	59.1%	58.6%	Setup: FuncTT0p9v DCLK: 0/0/0 Hold: FuncFFG1p05v DCLK: -62.9ps/57.6ns/1652	Congestion : 0.22% Setup: FuncTT0p9v DCLK: -12.77ps/-0.2ns/131 clock_gating_default 0/0/0 Setup: FuncTT0p65v DCLK: -0.42ps/-0.0004ns/1 clock_gating_default 0/0/0	Congestion : 0.48% Setup: FuncTT0p9v DCLK: -14.76ps/-0.016ns/2 clock_gating_default -16.65ps/-0.029ns/2 Setup: FuncTT0p65v DCLK: -47.2ps/-12.1ns/2394 clock_gating_default -30.3ps/-0.087ns/13	Setup: FuncTT0p9v DCLK: 0/0/0 DCLK_halfcycle: 0/0/0 Setup: FuncTT0p65v DCLK: 0/0/0 DCLK_halfcycle: 0/0/0	256962/59
Pd Agent				60.9%	62.9%	61.6%	Setup: FuncTT0p9v DCLK: 0/0/0 Hold: FuncFFG1p05v DCLK: -62.9/57.6/1652	Congestion : 0.26% Setup: FuncTT0p9v DCLK: -0.21ps/-0.0002ns/1 clock_gating_default -0.09ps/-0.00009ns/1 Setup: FuncTT0p65v DCLK: -18ps/-0.13ns/19 clock_gating_default 0/0/0	Congestion : 0.41% Setup: FuncTT0p9v DCLK: -6.2ps/-0.04ns/15 clock_gating_default -19ps/0-0.022ns/3 Setup: FuncTT0p65v DCLK: -31.7ps/-2.2ns/235 clock_gating_default -19ps/-0.038ns/10	Setup: FuncTT0p9v DCLK: 0/0/0 DCLK_halfcycle: 0/0/0 Setup: FuncTT0p65v DCLK: 0/0/0 DCLK_halfcycle: -97ps/0.6ns/8	252920/56

Result -- PD Resource Saving by Agent

- Below chart show the 651 mails that has been executed during 30 days.
- These agents needs process 29.59 mails(not just read, need accomplish in Linux) per working days.
- Generally, and PD process 3~4 mails which need real execution; thus, 7.3~9.8 PD resource is saved with existed agents.

Agent Name	Job Description
VTO4	FCT agent, responsible for TB restart/file existence check
vto7	tile owner agent, responsible for TB start/flow update, status report/monitoring
vto6	tile owner agent, responsible for TB start/flow update, status report/monitoring
vir0	EMIR agent, responsible for TB start/restart/monitoring
VTOFCT02	FCT agent, responsible for daily FCT start/monitoring
vto9	tile owner agent, responsible for TB start/flow update, status report/monitoring
VTOFCT01	FCT agent, responsible for daily FCT start/monitoring
VTO5	FCT agent, responsible for TB restart/file existence check
vto1	FCT agent, responsible for TB restart/file existence check



Conclusion

- The PD agent can independently take over the entire task, from receiving emails to executing tasks in Linux using various tools (EDA/script). This helps save 7.3~9.8 total resources in PD, rather than just saving effort while resources are not reduced.
- Through role definition, the PD agent can be extended with different functions (tile/FCT/EMIR), different models (LLM/non-LLM), different frameworks (langchain/AutoGPT...), and different platforms (Linux/Windows).
- The PD agent can completely cover easy tiles for starting/monitoring runs, updating flows, and releasing data. Meanwhile, the Quality of Results (QOR) is comparable to that of humans, thus freeing up tile owner resources from NLA~NLC.
- The PD agent can completely cover the FCT daily ECO run for pointer collection and starting/monitoring runs, meanwhile speedup the setup timing fix by 50%, help DXIO clock cleaning shift left 5 days.
- Non-LLM is still suitable for executing most tasks involving tool calls; LLM is suitable for queries, such as instruction interpretation, status reporting, and issue assignment at the current stage due to its accuracy limitations.

