Android Quick Boot

Bin Yang, Software Engineer **O**pen **S**ource **T**echnology **C**enter, **S**oftware and **S**ervices **G**roup (OTC/SSG)

Intel Asia-Pacific Research & Development Ltd

NOTICE & DISCLAIMER

- Intel technologies' features and benefits depend on system configuration

and may require enabled hardware, software or service activation.

- Performance varies depending on system configuration.
- Intel, the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.

Agenda

- Background
- Measurement
- Methodology and Solution

Background

Video for Android Boot

Methods for Quick Boot

- Normal code boot
  - Fundamental boot flow
- Resume from S3
  - Similar concept to standby in laptop.
  - With a slight power consumption during standby.
- Resume from S4
  - High eMMC traffic involved when going to hibernation. Impacting eMMC lifetime.

Focus of This Presentation

- Normal cold boot
- Focus on AOSP
- No feature reduction (TEE, security boot, full disk encryption, full Android Open Source Project (AOSP*)
- Android 6.0

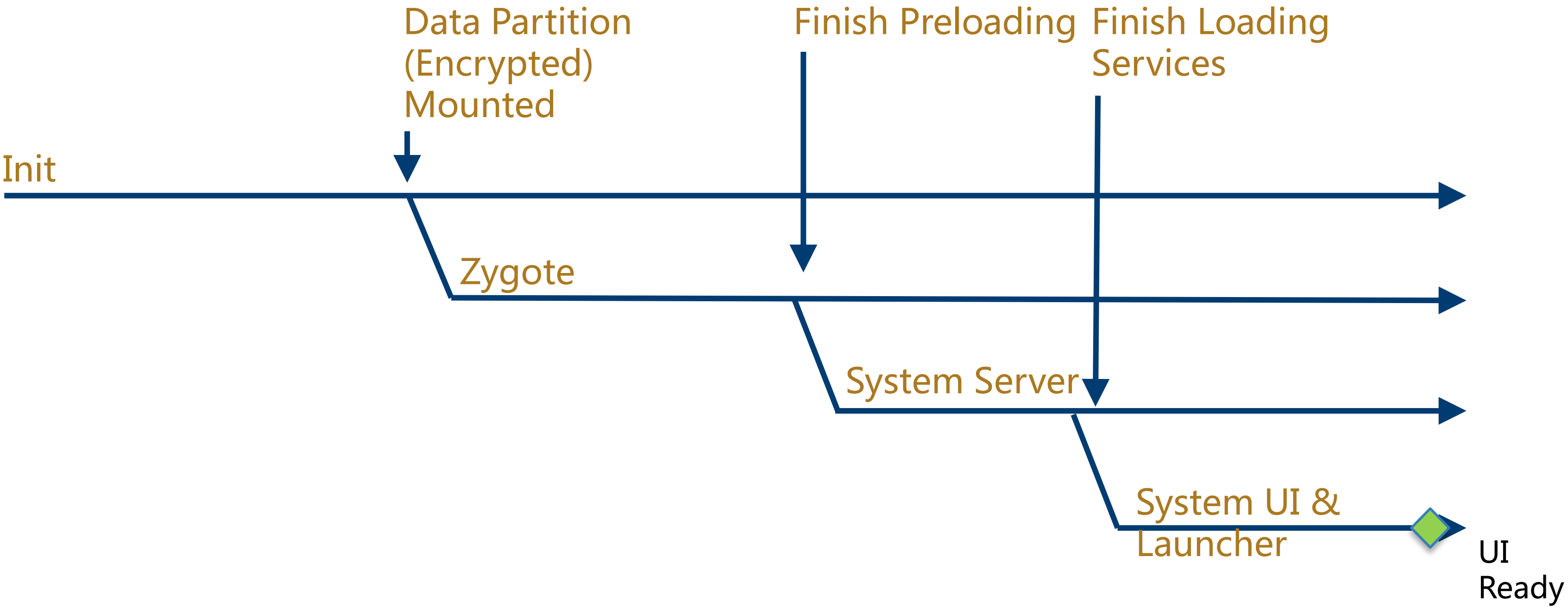
Measurement

Measurement

- Bootchart
- Systrace
- Automate regression check tool

Methodology and Solution

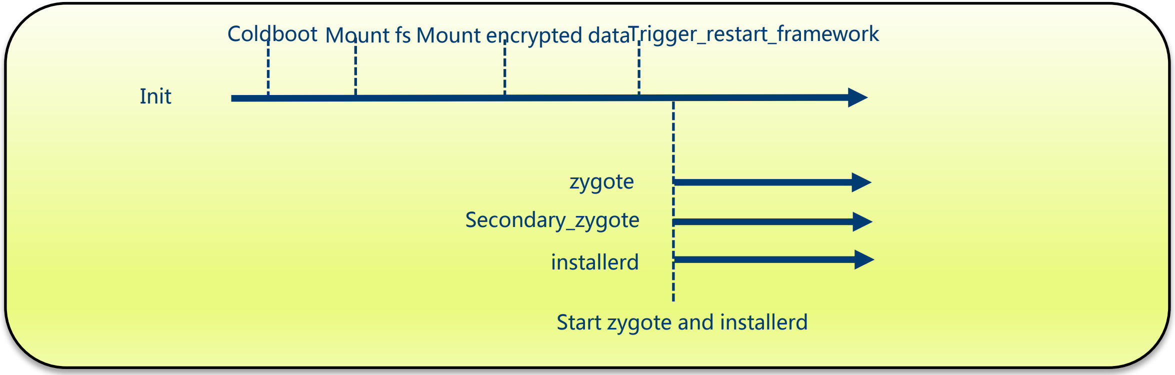
AOSP* Boot Sequence

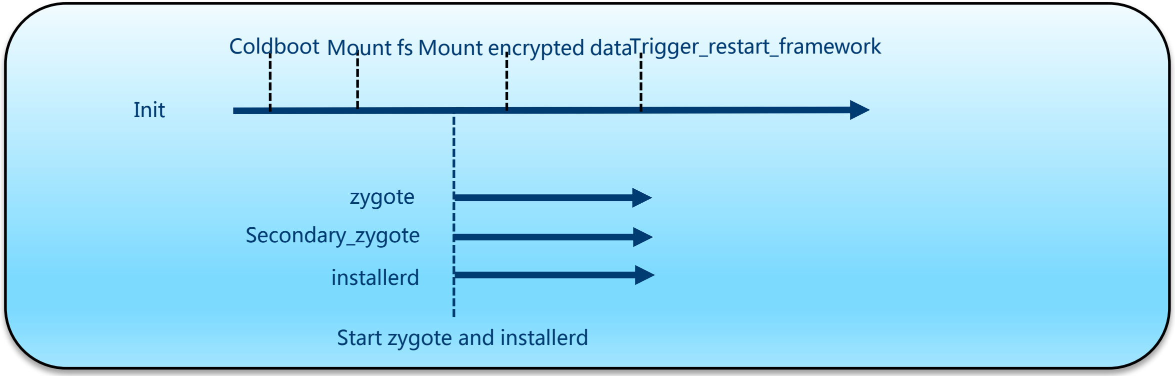


Methodology and Solution

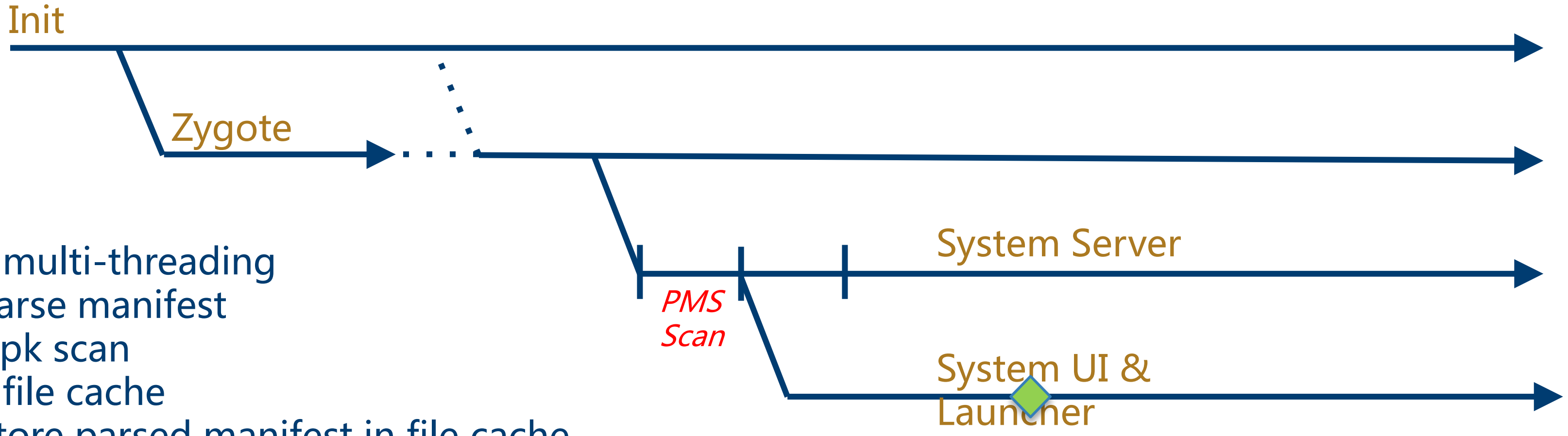
- Parallelization
  - Start Zygote ASAP
  - PMS Scan in Parallelization
- On Demand load
  - Preload class and resource on demand
  - Initialize Services Related to GUI
  - Split System UI
- Read ahead
- Reduce useless waiting operation

Zygote Start Early





PMS Scan in Parallel



Use multi-threading

- Parse manifest
- Apk scan

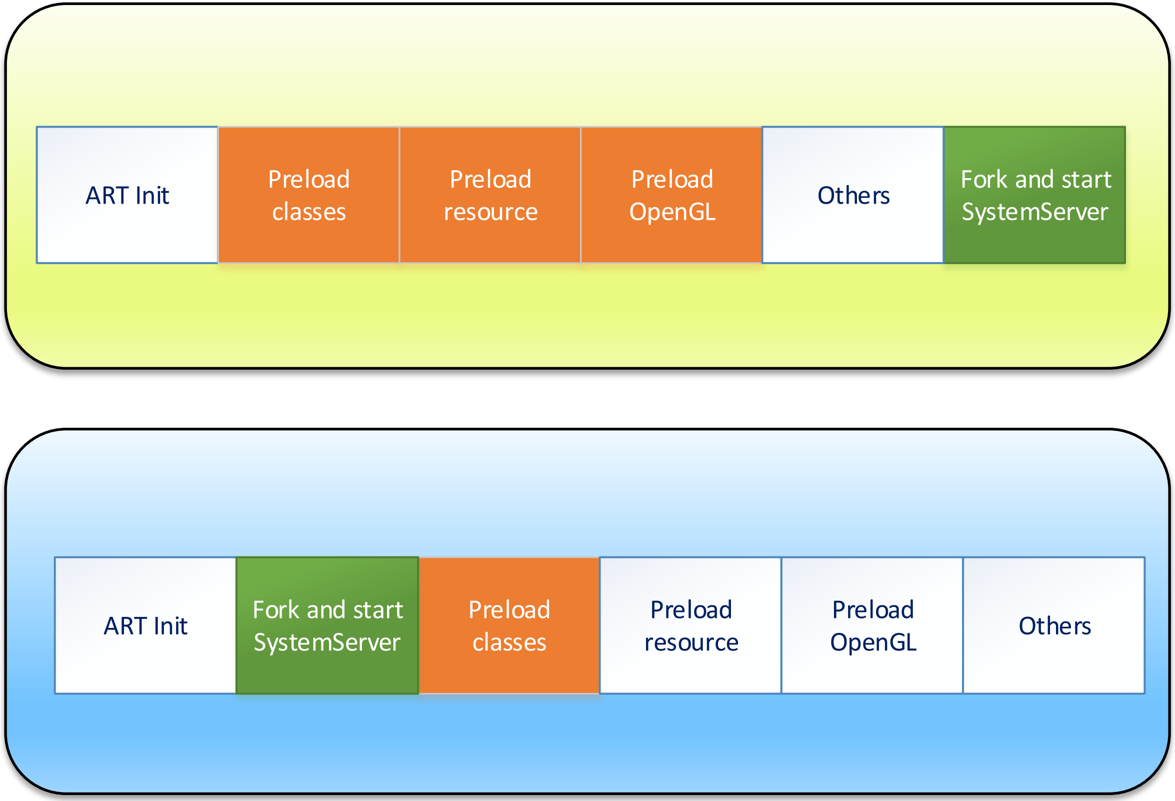
Use file cache

- Store parsed manifest in file cache.
- Reparse only when cache is outdated.

Methodology and Solution

- Parallelization
  - Start Zygote ASAP
  - PMS Scan in Parallelization
- On Demand load
  - Preload class on demand
  - Initialize Services Related to GUI early
  - Split System UI
- Read ahead
- Reduce useless waiting operation

Preload Resources on Demand



ART Init | Preload classes | Preload resource | Preload OpenGL | Others | Fork and start SystemServer

ART Init | Fork and start SystemServer | Preload classes | Preload resource | Preload OpenGL | Others

Heap Split

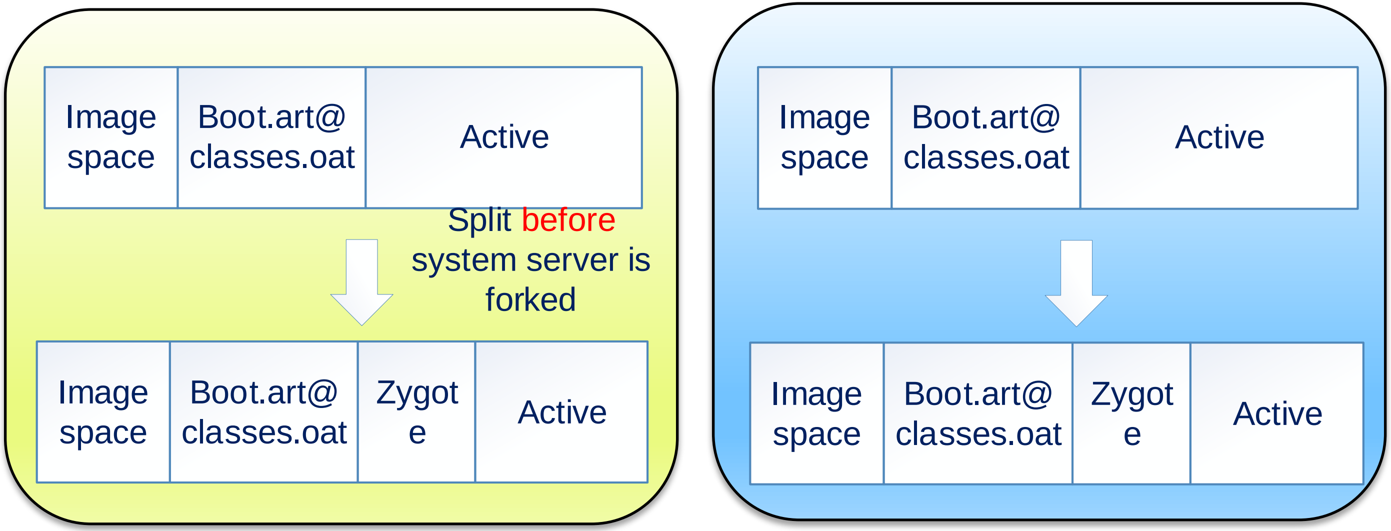
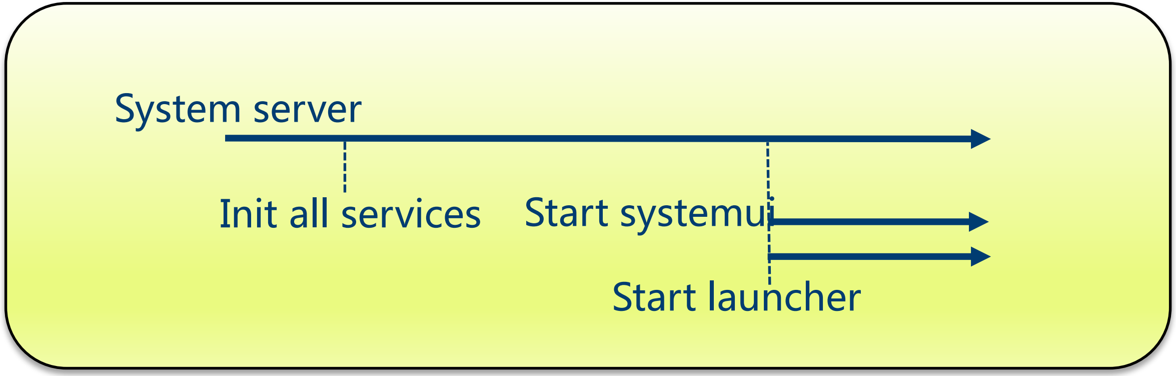
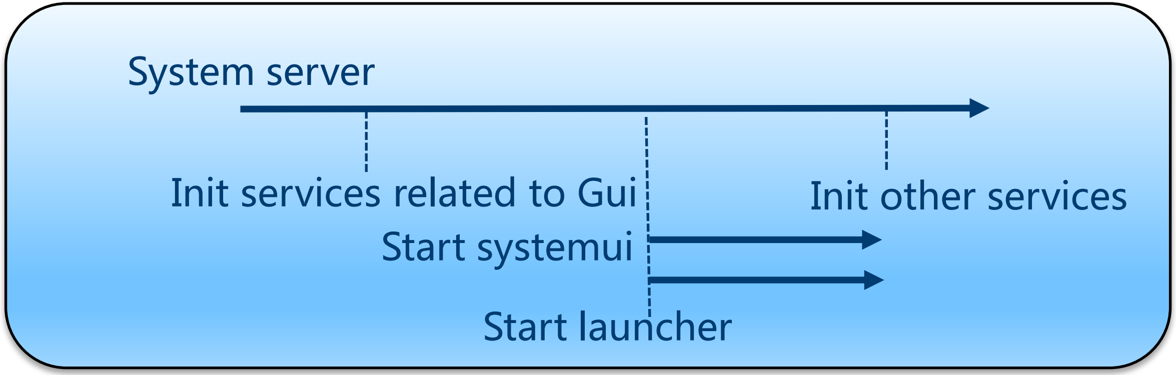


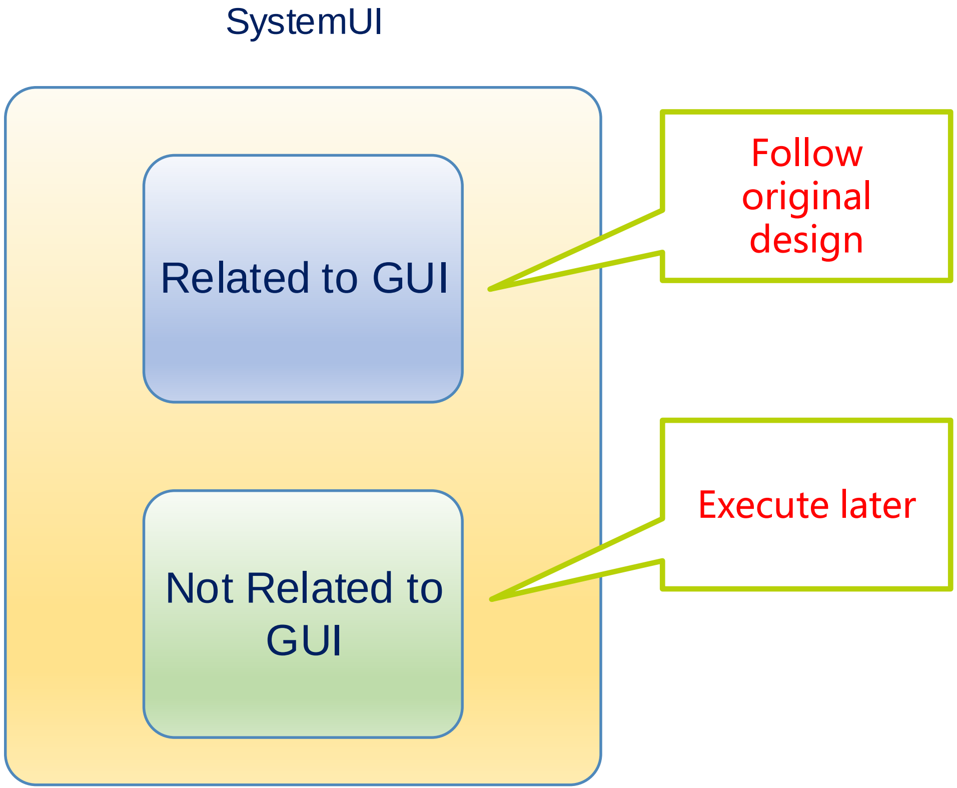
Image space
Boot.art@ classes.oat
Active
Split before system server is forked
Image space
Boot.art@ classes.oat
Zygot e
Active
Image space
Boot.art@ classes.oat
Active
Image space
Boot.art@ classes.oat
Zygot e
Active

Start Services Related to GUI Early





Split SystemUI into Two Parts



Methodology and Solution

- Parallelization
  - Start Zygote ASAP
  - PMS Scan in Parallelization
- On Demand load
  - Preload class on demand
  - Initialize Services Related to GUI early
  - Split System UI
- Read ahead
- Reduce useless waiting operation

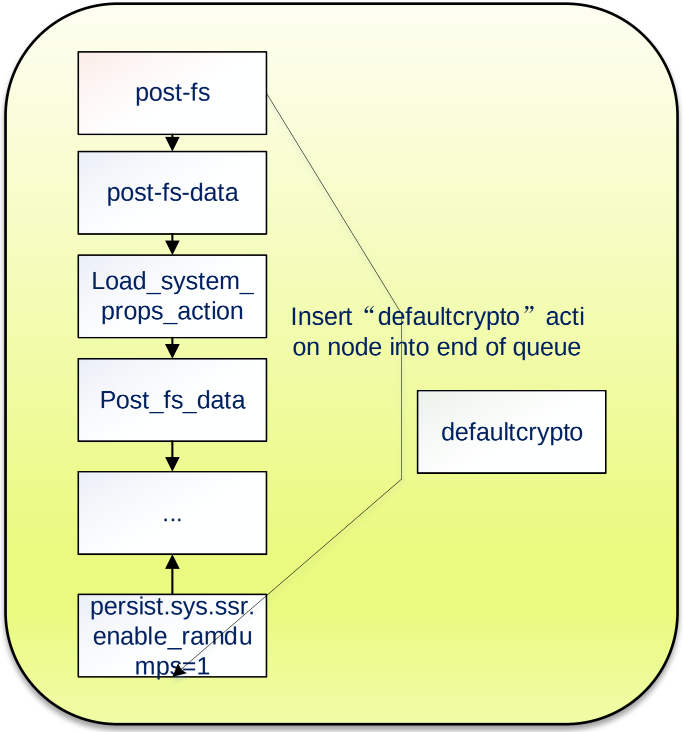
Read Ahead

- Boot.oat/Boot.art
- libLLVM.so
- Libart.so
- framework-res.apk
- libskia.so
- libandroid_runtime.so
- libicuuc.so

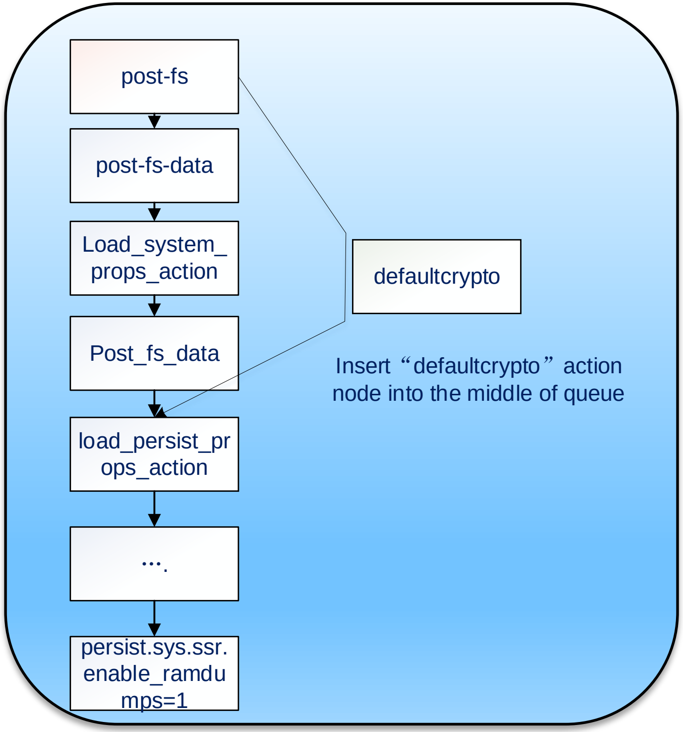
Methodology and Solution

- Parallelization
  - Start Zygote ASAP
  - PMS Scan in Parallelization
- On Demand load
  - Preload class on demand
  - Initialize Services Related to GUI early
  - Split System UI
- Read ahead
- Reduce useless waiting operation

Reduce Useless Waiting



Insert "defaultcrypto" acti on node into end of queue



Insert "defaultcrypto" action node into the middle of queue

Q & A

intel®

Software