Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

Unification of RAS feature control - Enhancing EDAC

Shiju Jose (Huawei), Vandana Salve (Micron), Jonathan Cameron (Huawei)

LINUX
PLUMBERS
CONFERENCE Vienna, Austria / Sept. 18-20, 2024

Scope

- What do we mean by RAS features?
- Why in kernel?
- Use cases
- Open questions.

Aim for today:

- Discussion of open issues,
- Identify blockers / additional use cases

RAS Features to Control? (CXL etc)

Examples *

- Patrol Scrub
- Error Check Scrub
- Memory Sparing
- hard and soft post package repair
- Error reporting threshold control. (General, or CXL specific?)

Other topic.

- Memory capacity reduction.

(*) Details provided in backup slides.

Why in kernel? (not fwctl)

- Generalized interface - whatever hardware provider: same kernel interface.
  - Existing examples, but no interface unification: NVDIMM ARS, EDAC scrub.

- Operations only safe under some circumstances*
  - e.g. sPPR on devices where accesses must be quiesced.

Enhancing EDAC for RAS feature control

- Why EDAC? It’s where related stuff is!
- Path to enhancement.
  - Existing EDAC model is ‘unusual’ wrt to the Linux Device model.
  - Pure sysfs interface, reporting independent (current paths)
  - Here, conventional subsystem bus with feature instances under

```
/sys/bus/edac/devices/cxl_mem0/scrub0
/sys/bus/edac/devices/cxl_region0/scrub0
/sys/bus/edac/devices/acpi_ras2_mem0/scrub0
…
```

- Some features naturally associate with CXL regions, some with device instance, some both!

For detailed architecture see the patches.

RASDaemon based Scrub PoC

- RASDaemon - Widely used RAS error logging and control.
- Just an example - No idea what the real policy will be yet!

1. Correctable Error Rate > Threshold

2. Use predefined ‘bad device’ scrub rate

Real system would probably use a more complex control policy!

https://github.com/shijujose4/rasdaemon/commits/ras_feature_control/

Issues

- Some specifications are very vague in definitions
  - RAS2 scrub rate has no units!
  - In theory specifications allows multiple scrub ranges, but no discoverability.

Open Questions

- Does this meet everyone’s requirements?
  - Virtualization use case?
  - What else is in scope?
- Worth including existing interfaces? NVDIMM ARS etc.
- How much to expose here vs let user space figure out? Address ranges, NUMA nodes
- Attributes in driver vs callbacks?
  - Sanity checks in the core (can I repair this memory?)
  - Interface generality enforcement.
- Need policy control on exposing controls?
  - Ability to turn off scrub, set minimums levels etc.
  - (Or is this a hardware / firmware problem?)

References

- CXL Specification: https://computeexpresslink.org/cxl-specification/
- ACPI Specification: https://uefi.org/specifications
- Blog: Augmenting EDAC for controlling RAS Features https://gitee.com/shijujose/openEuler-portal/blob/edac_enhancement_ras_features/app/en/blog/ras/edac-enhancement-for-control-ras-features.md
- [PATCH v12 00/17] EDAC: Scrub: introduce generic EDAC RAS control feature driver + CXL/ACPI-RAS2 drivers https://lore.kernel.org/linux-cxl/20240911090447.751-1-shiju.jose@huawei.com/T/#t

Backup slides. The details

Patrol Scrub

What is it?

- Memory controller periodically reads every DRAM row.
- ECC checks performed.
- Corrected and uncorrectable errors reported via normal paths.

How might an Admin use it?

- Linux may elect to take immediate action, or use this for information only; perhaps allowing safe offlining of the memory.
- Differentiated reliability - trade off performance vs stability (Direct admin control)
  - Early fixing of corrected errors that ‘might’ degrade to uncorrectable.
- Mitigation against known failing part (RASDaemon PoC)
- Policy application to hotplugged device (UDev Script)

Error Check Scrub (ECS)

What is it?

- ECS feature is defined in JEDEC DDR5 SDRAM Specification.
- Allows the DRAM to internally read, correct single-bit errors, and write back corrected data bits to the DRAM array while providing transparency to error counts.
- The ECS control feature allows to configure ECS parameters: ECS threshold count, mode of operation, reset the ECS counter etc.
- ECS log may be available.

How might and Admin use it?

- DDR5 ECS is able to handle corrected errors only. When a corrected error is fixed by ECS, the device shall generate an event record.
- Linux may elect to take immediate action, or use this for information only perhaps allowing safe offlining of the memory.

Memory Sparing

What is it?

- Memory sparing is a repair function that replaces a portion of memory (spared memory) with a portion of functional memory at that same DPA.
- Cacheline/row/bank/rank memory sparing replaces full cacheline/single DDR row/entire bank/entire DDR rank respectively.

How might and Admin use it?

- User space tool, e.g. rasdaemon, may request the sparing operation for a given address for which uncorrectable error is reported.

Post Package Repair (PPR)

What is it?

- PPR is maintenance operation requests the memory device to perform a repair operation on its media.
- Hard PPR (hPPR), for a permanent row repair, and Soft PPR (sPPR), for a temporary row repair.
- During PPR, memory device, may or may not retain data and may or may not be able to process memory requests correctly.

How might and Admin use it?

- User space tool, e.g. rasdaemon, may request the PPR operation for a given address, for which uncorrectable error or excess corrected errors is reported.
- At boot apply sPPR to known set of bad memory addresses (emulating hPPR)

Error Reporting Threshold Control

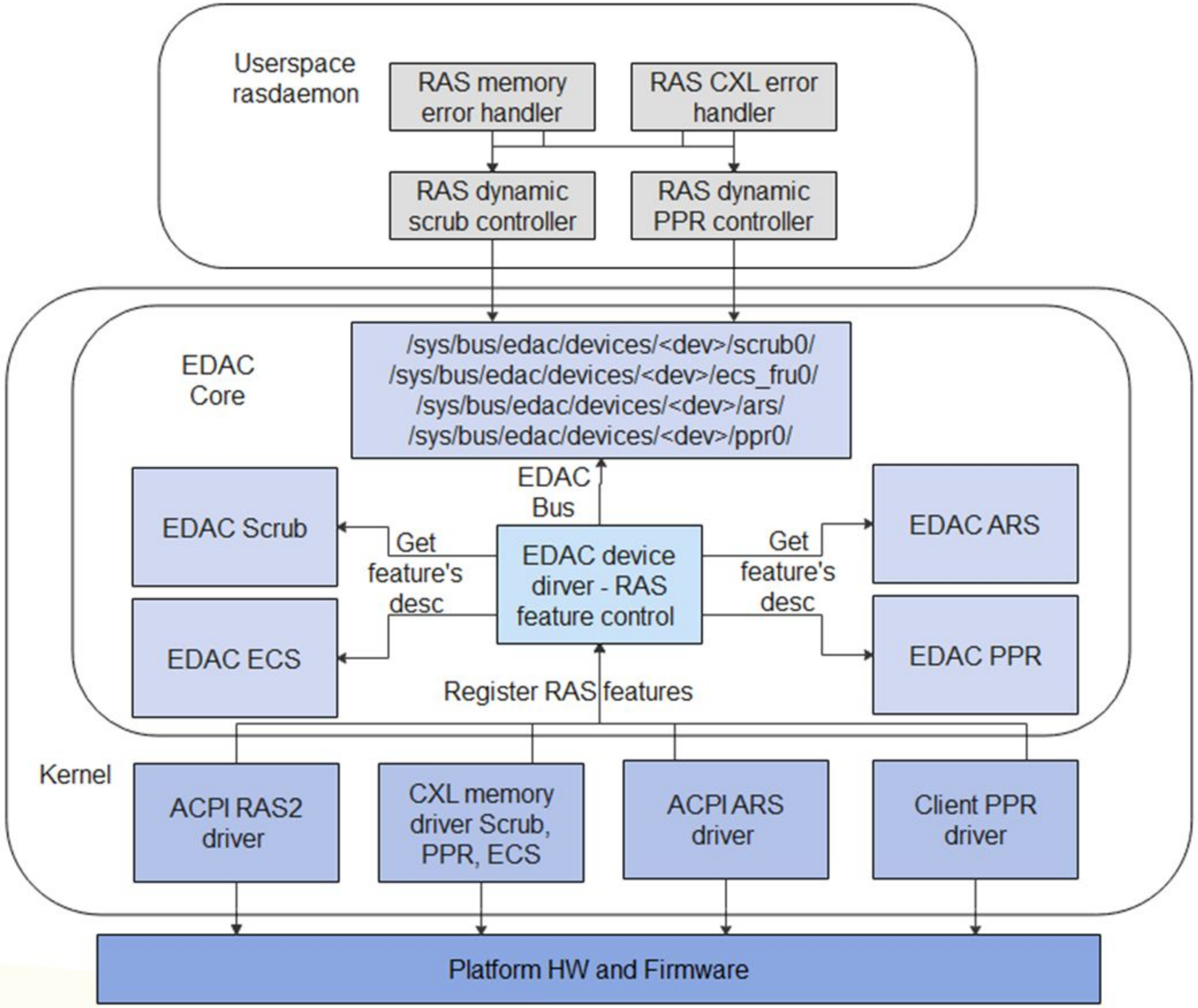
What is it?

- Corrected Volatile Memory Error (CVME) Threshold: When enabled, the device shall maintain separate counters and thresholds for CVMEs detected during patrol scrub and non-patrol scrub accesses.
- CVME counters used by patrol scrub CVME thresholds are automatically reset at the end of every patrol scrub cycle.
- Threshold Granularity : Full memory/per memory media FRU/per rank.

How might and Admin use it?

- Likely hotplug only - so configuring a new device to match fleet wide settings.

Proposed Software Architecture


Userspace rasdaemon
RAS memory error handler
RAS CXL error handler
RAS dynamic scrub controller
RAS dynamic PPR controller
EDAC Core
/sys/bus/edac/devices/<dev>/scrub0/
/sys/bus/edac/devices/<dev>/ecs_fru0/
/sys/bus/edac/devices/<dev>/ars/
/sys/bus/edac/devices/<dev>/ppr0/
EDAC Bus
EDAC Scrub
EDAC ECS
Get feature's desc
EDAC device dirver - RAS feature control
Get feature's desc
EDAC ARS
EDAC PPR
Register RAS features
Kernel
ACPI RAS2 driver
CXL memory driver Scrub, PPR, ECS
ACPI ARS driver
Client PPR driver
Platform HW and Firmware