Unifying sources of page hotness information

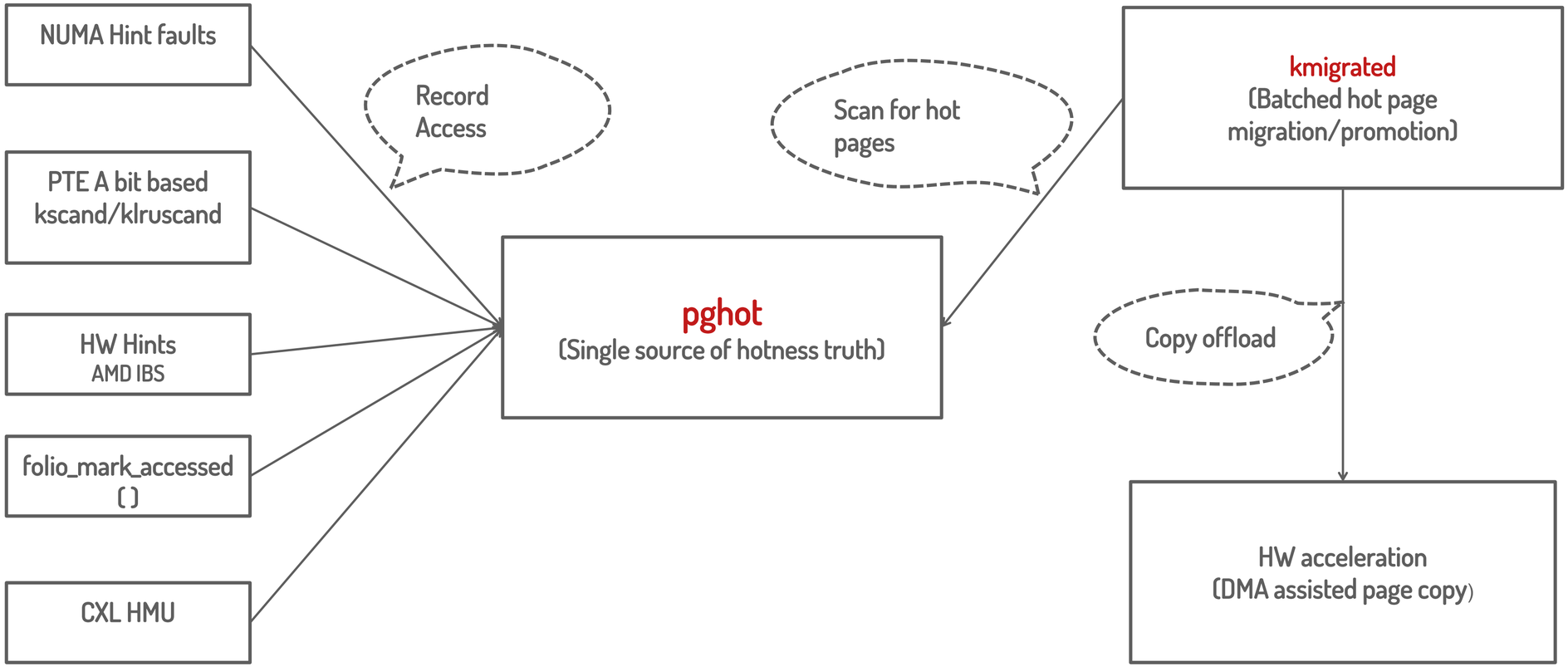
Device and Specific Purpose MC

Bharata B Rao, AMD

The problem space

- Multiple subsystems within the kernel that detect page hotness and act upon it
- Possibility of consolidating them into a dedicated hotness detection/monitoring subsystem
  - Single source of hotness truth
- Promotion is currently tied to NUMA Balancing/Scheduler and occurs in process context
  - Delink it
- Batched migration

pghot – Single source of page hotness truth



The solution – pghot subsystem

- API that subsystems can use to report a detected hot page
- Single place for hot page heuristics, promotion rate-limiting, etc.
- Per-PFN hot page record (unsigned long)
  - Part of mem_section
  - 2GB per TB of lower-tier memory (0.2%)
- Per-lower-tier kernel threads for promotion
  - Scan PFNs of sections containing ready-to-migrate pages

```
/**
 * pghot_record_access - Record page accesses from lower tier memory
 * for the purpose of tracking page hotness and subsequent promotion.
 *
 * @pfn - PFN of the page
 * @nid - Target NID to where the page needs to be migrated
 * @src - The identifier of the sub-system that reports the access
 * @now - Access time in jiffies
 *
 * Updates the NID, frequency and time of access and marks the page as
 * ready for migration if the frequency crosses a threshold. The pages
 * marked for migration are migrated by kmigrated kernel thread.
 *
 * Return: 0 on success and -EAGAIN on failure to record the access.
 */
int pghot_record_access(unsigned long pfn, int nid, int src, unsigned long now)
```

63	32-62	13-31	10-12	0-9
Ready	Unused	Time	Freq	NID
1bit	31bits	19bits	3bits	10bits

Per-PFN hot page record– unsigned long

Source 1 – NUMA hint faults

- NUMA_BALANCING_MEMORY_TIERING mode reports accesses to pghot sub-system
  - Address space scanning and hint faulting mechanism remain
  - Accesses from hint fault handler are reported to pghot
  - Batched misplaced migration by kmigrated
- Hot page heuristics moved to pghot
  - Dynamic hotness threshold and rate limiting

Source 2 – PTE Accessed bit scanning

- Two approaches: kscand and MGLRU-based klruscand
- Kscand
  - Scans process address space of all processes looking for Accessed bit
  - Target NID detection heuristics, scan rate limiting, per-process opt-out
- Klruscand
  - Extension to MGLRU's page table walk that provides hooks to obtain Accessed bit information
  - Kernel daemon that periodically invokes MGLRU page table walk

Source 3 – HW hints (AMD IBS)

- IBS – HW facility in AMD processors to gather metrics related to instruction fetch and execution
  - Independent of PMU
  - Execution sampling for memory access profiling
  - Provides PA, VA and data source (Cache, DRAM, remote CXL) for each access
- Arch-specific IBS driver feeds access information to pghot

Source 4 – folio_mark_accessed(), FMA

- Detecting accesses on folios that aren't mapped to process address space
- Targeting unmapped page cache folios
- Feeds access info to pghot sub-system

Results: Memory access – Promotion only

Benchmark details

- 3 Node AMD Zen 5 system with two regular nodes (0, 1) and a CXL node (2)
- Multi-threaded application with 64 threads running on Node 0, memory provisioned on CXL node 2 using **mmap(MAP_POPULATE)**
- Fixed number of random and repetitive accesses resulting in promotion
- Time to complete the accesses is the benchmark score, lower is better

(*) With promotion on 2nd access:

Pghot-NUMAB2 = 105,636,591 us

Time taken (us), lower is better

Source	Base	Pghot	Change
NUMAB0	118,986,471	116,240,187	-2.3%
NUMAB2	104,025,651	103,120,089	-0.8% **(*)**
pgtscan	NA	110,800,511	+6.5% wrt NUMAB2
hwhints	NA	100,442,082	-3.4% wrt NUMAB2

Pages migrated

Source	Base	Pghot	Change
NUMAB0	0	0	0%
NUMAB2	2097152	2097152	0%
pgtscan	NA	2097152	NA
hwhints	NA	1232876	NA

Results: File access– Promotion only

Benchmark details

- 3 Node AMD Zen 5 system with two regular nodes (0, 1) and a CXL node (2)
- Multi-threaded application with 64 threads running on Node 0, memory provisioned on CXL node 2 using **mmap(fd, MAP_POPULATE | MAP_SHARED)**
- Fixed number of random and repetitive accesses resulting in promotion
- Time to complete the accesses is the benchmark score, lower is better

(*) With promotion on 2nd access:

Pghot-NUMAB2 = 84,999,971 us

Time taken (us), lower is better

Source	Base	Pghot	Change
NUMAB0	113,352,595	110,053,021	-2.9%
NUMAB2	72,339,008	72,117,399	-0.3% **(*)**
pgtscan	NA	66,189,266	-8.5% wrt NUMAB2
hwhints	NA	71,644,577	-0.9% wrt NUMAB2

Pages migrated

Source	Base	Pghot	Change
NUMAB0	0	0	0%
NUMAB2	2097152	2097152	0%
pgtscan	NA	2097152	NA
hwhints	NA	1232876	NA

Results: Memory access – Promotion with demotion

Benchmark details

- 3 Node AMD Zen 5 system with two regular nodes (0, 1) and a CXL node (2)
- Single-threaded application running on Node 1, memory provisioned on DRAM node 1 and CXL node 2 using **mmap(MAP_POPULATE).** CXL memory is divided into 1G regions
- Fixed number of random and repetitive accesses into CXL regions resulting in demotion from DRAM and promotion from CXL
- Time to complete the accesses is the benchmark score, lower is better

(*) With promotion on 2nd access:

Pghot-NUMAB2 = 63,087,940 us

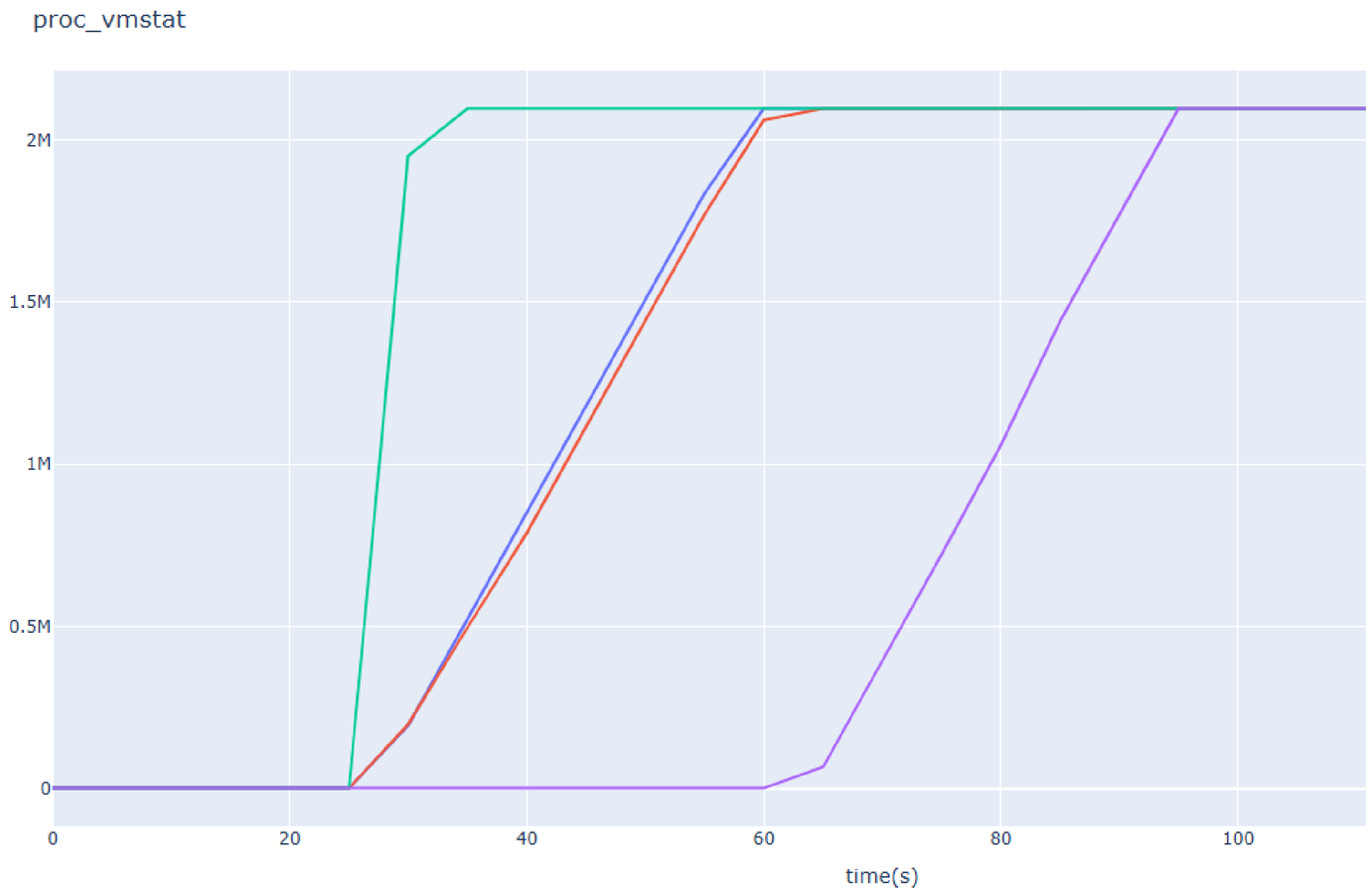
Time taken (us), lower is better

Source	Base	Pghot	Change
NUMAB0	61,537,418	59,165,269	-3.8%
NUMAB2	62,070,563	61,909,849	-0.2% **(*)**
pgtscan	NA	66,886,552	+7.7% wrt NUMAB2
hwhints	NA	63,354,394	+2.0% wrt NUMAB2

Pages migrated

Source	Base	Pghot	Change
NUMAB0	0	0	0%
NUMAB2	0	1186242	
pgtscan	NA	6481483	NA
hwhints	NA	304	NA

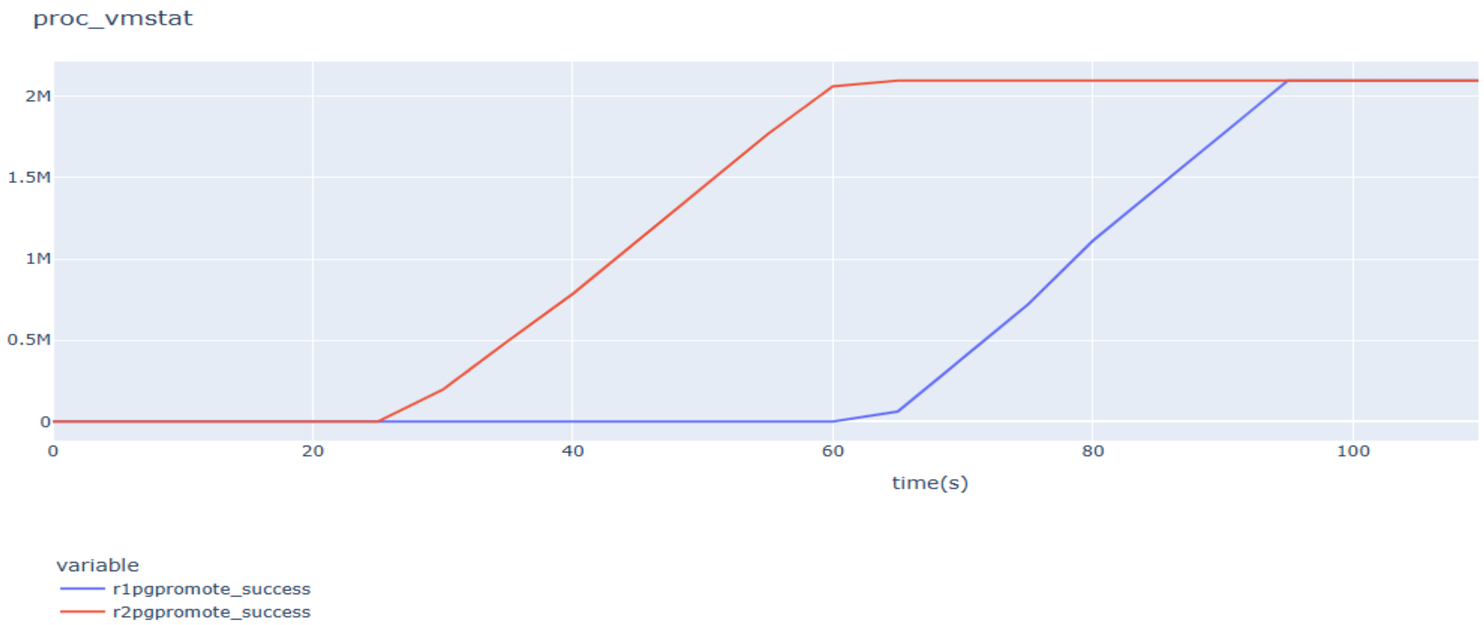
Results – Rate of migration



pgpromote_success

- blue: base-NUMAB2
- red: pghot-NUMAB2 (1st access promotion)
- violet: pghot-NUMAB2 (default 2nd access promotion)
- green: pghot-pgtscan

Results – Full PFN scan vs hot section scan, NUMAB2



pgpromote_success, 1st access promotion

- r1-blue: Full PFN scan (pghot-v3)
- r2-red: Hot section scan (pghot-v4)

```
kmigrated2-2863    [063] .....   281.592612: kmigrated_do_work: scanned 4094 skipped 2046 hot 3 cold 2045
kmigrated2-2863    [063] .....   281.594188: kmigrated_do_work: scanned 4094 skipped 2046 hot 1 cold 2047
kmigrated2-2863    [063] .....   282.587755: kmigrated_do_work: scanned 4094 skipped 2046 hot 3 cold 2045
kmigrated2-2863    [063] .....   283.467105: kmigrated_do_work: scanned 4094 skipped 2046 hot 3 cold 2045
```

Results – Promotion of unmapped page cache folios

Benchmark details

- 3 Node AMD Zen 5 system with two regular nodes (0, 1) and a CXL node (2)
- Single-threaded application running on Node 0 provisions a 2G file on CXL node initially
- Fixed number of random and repetitive accesses**(pread)** to the file resulting promotion from CXL
- Time to complete the accesses is the benchmark score, lower is better

Time taken (us), lower is better

	Base	Pghot – FMA disabled	Pghot – FMA enabled
Time taken (us)	96,511,260	119,332,436	82,807,865
Pages migrated	0	0	524242

Discussion points

- Recheck on the need for a centralized hotness monitoring subsystem
- Access reporting API: single and bulk reporting
- Hot page records organization
  - Space and time overhead
- kmigrated
  - In-time vs delayed and batched migration
  - Ability to migrating hottest pages first
  - Direct API for migration bypassing pghot
- Ways to classify a page as hot
- Different hotness sources
  - NUMAB2, HW Hints (IBS), PTE A bit, folio_mark_accessed()
- Benchmarks to test

Links

- pghot RFC v4 patchset
- kscand patchset

COPYRIGHT AND DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.