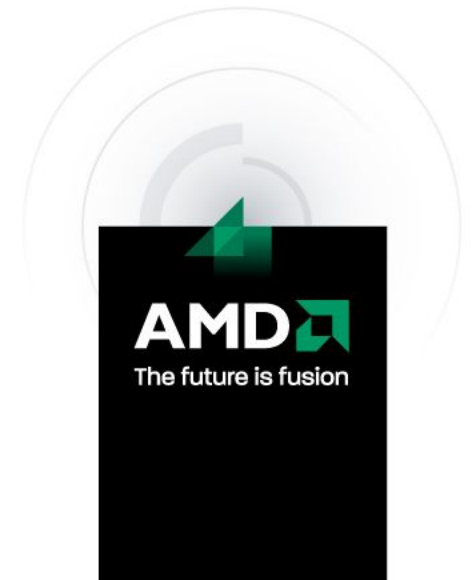


The Challenges of Guest Migration

Chris Schlaeger

Director Operating System Research Center



What is Guest Migration?

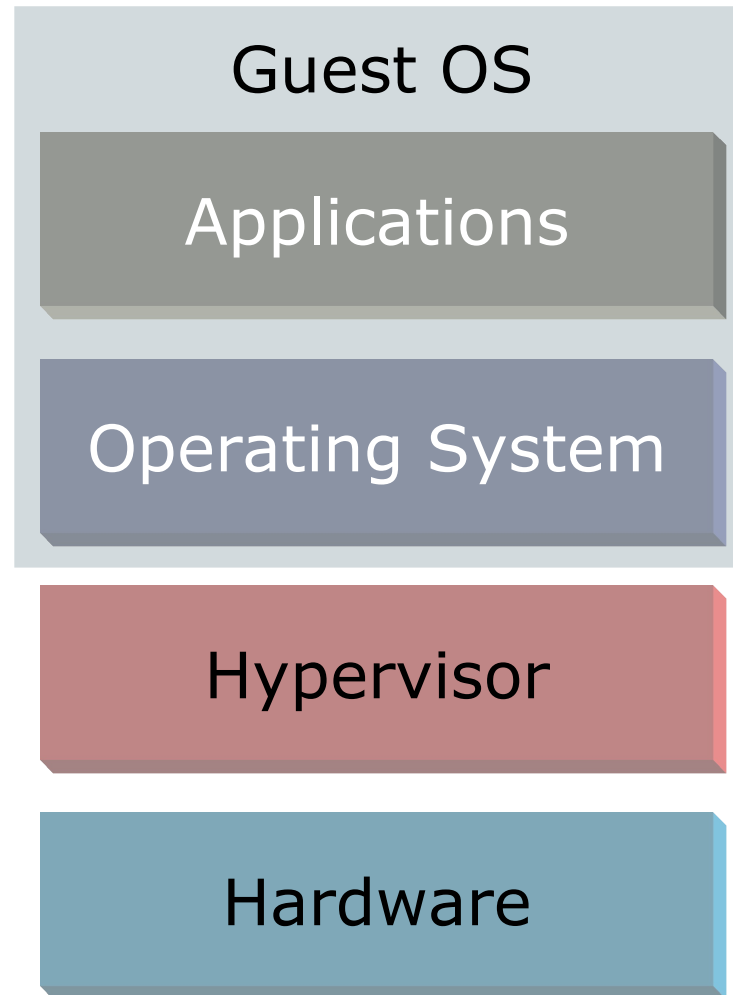
Applications

Operating System

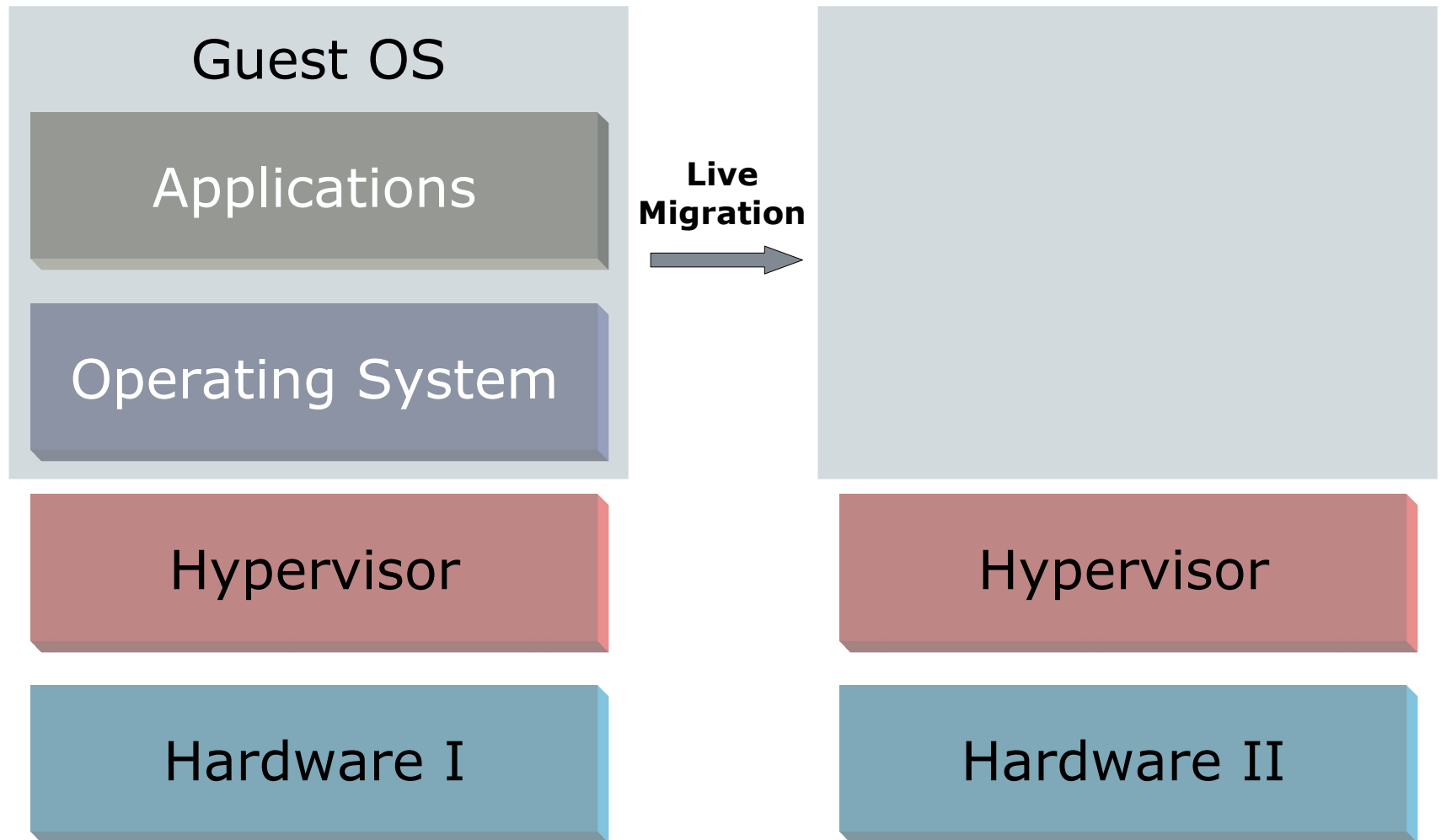
Hardware



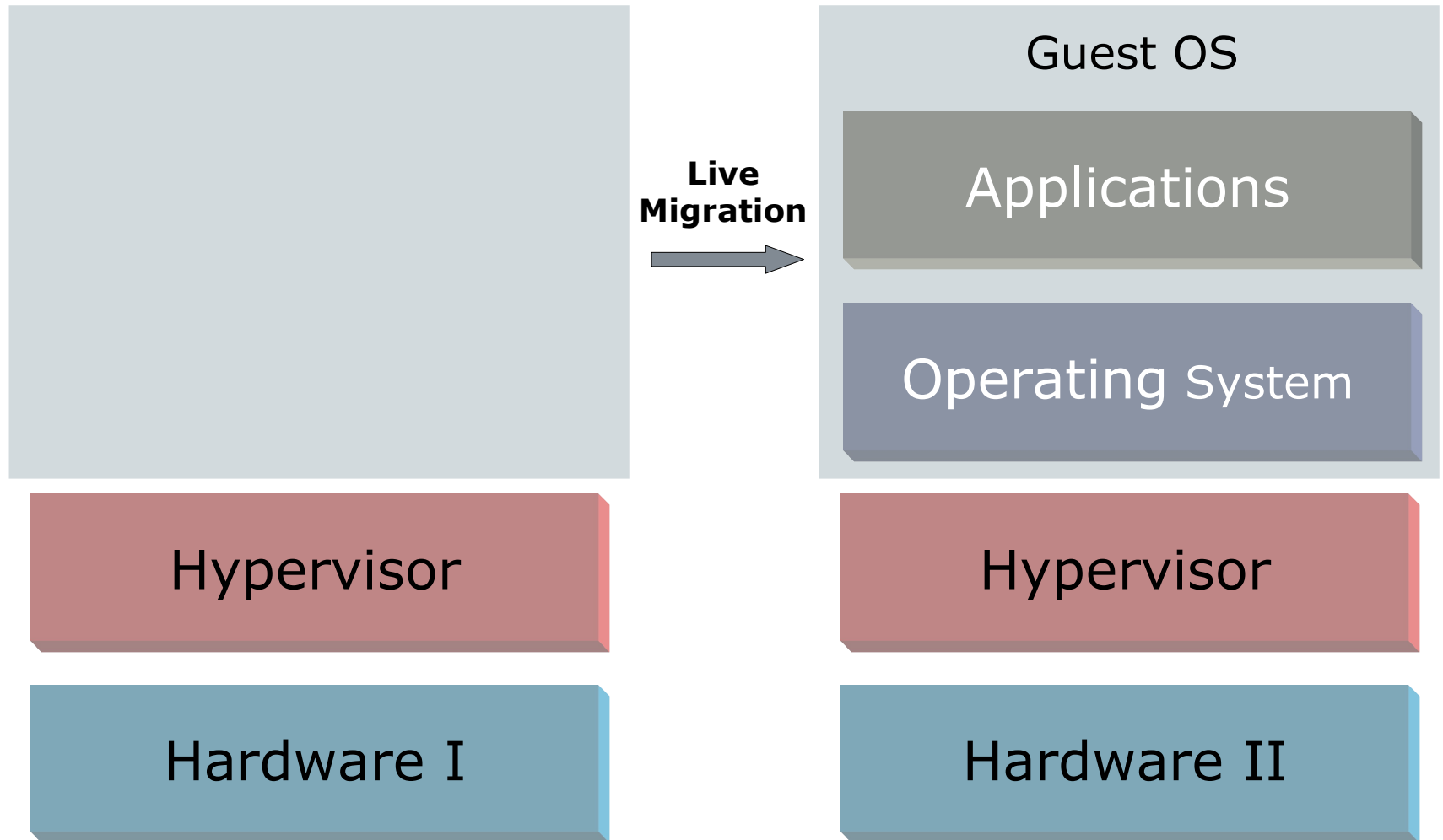
What is Guest Migration?



What is Guest Migration?



What is Guest Migration?



How does Live Migration work?

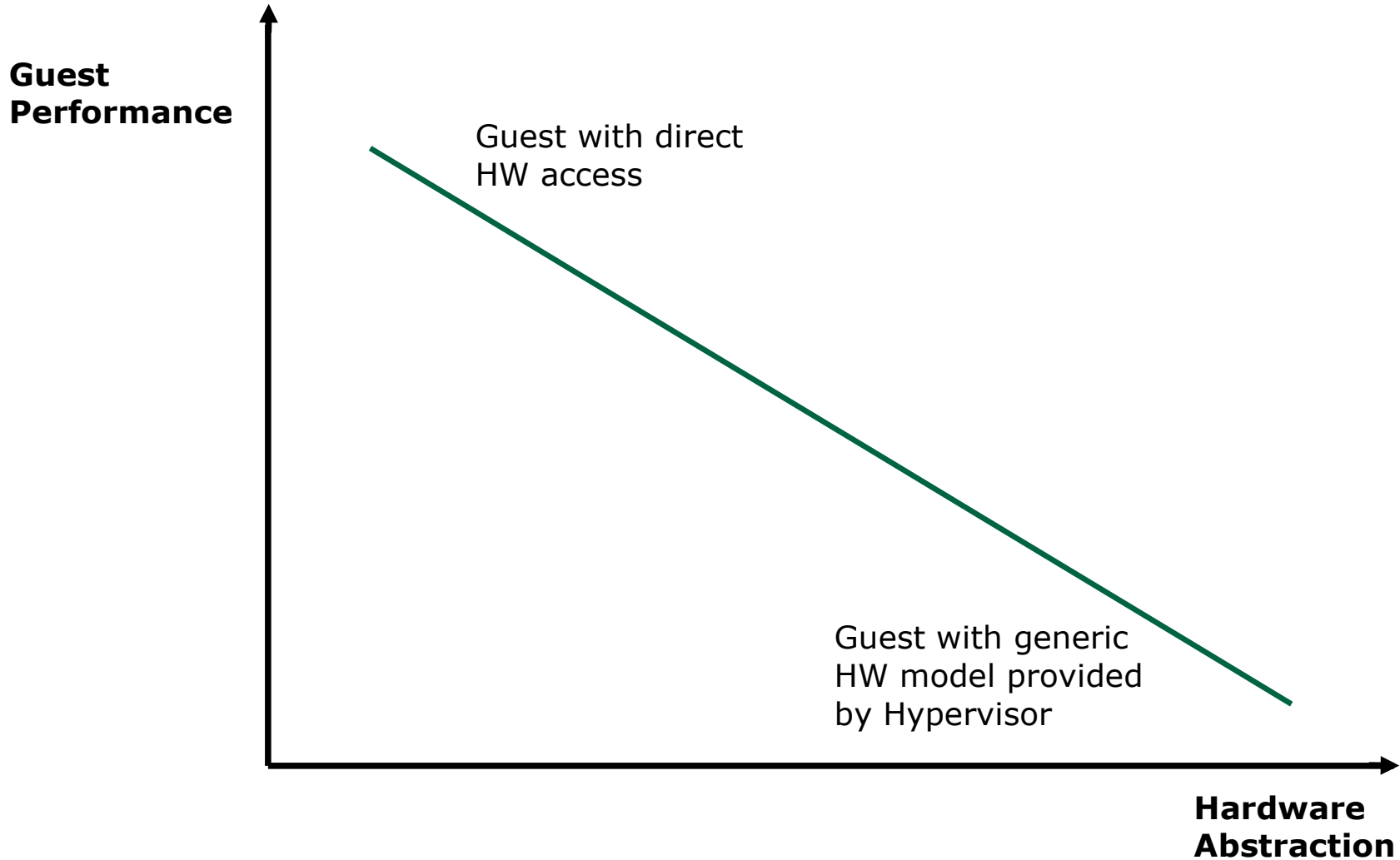
Generally:

1. Freeze the guest
2. Copy guest state to new host
3. Migrate guest network connection
4. Resume the guest

This process should be transparent to the guest (and any clients communicating with the guest).



Trade-off between HW-Abstraction and Performance



Least Common Denominator Approach

- Emulate as few features as possible
- Only expose features to guests that are present on all systems in the migration pool
- Once a feature has been detected it must stay present during the lifetime of the guest OS instance



Establishing your Baseline with CPUID Bits

```
# cat /proc/cpuinfo
processor      : 0
vendor_id     : AuthenticAMD
cpu family    : 17
model         : 3
model name    : AMD Turion(tm)X2 Ultra DualCore Mobile ZM-82
stepping      : 1
cpu MHz       : 600.000
cache size    : 1024 KB
physical id   : 0
siblings      : 2
core id       : 0
cpu cores     : 2
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level   : 1
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse
sse2 ht syscall nx mmxext fxsr_opt rdtscp lm 3dnowext 3dnow constant_tsc rep_good nopl pni cx16 lahf_lm
cmp_legacy svm extapic cr8_legacy 3dnowprefetch osvw skinit
bogomips      : 4400.08
TLB size      : 1024 4K pages
clflush size  : 64
cache_alignment : 64
address sizes  : 40 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate
```



Vendor String

- OS'es and hypervisors use the "vendor string".
- It should only be used to interpret the meaning of the other leaf pages!
- It should not be used to make assumptions about the underlying architecture!
- Currently generic strings don't work. Some OS'es have white lists and refuse to boot.
- The hypervisor exposes a fake real system to the guest.



Allowing multiple CPU Generations in the Migration Pool

All migration among heterogeneous CPUs depends on software's strict adherence to CPUID. If software doesn't find a feature in CPUID, it must not use the feature.

- Exceptions are grandfathered in: SSE1 storage detection, Monitor/Mwait in user mode.
- All SVM-capable and VT-x capable CPUs support at least SSE2, so SSE2 (or SSE3) is a good minimum that avoids the annoying legacy exceptions.
- Monitor/MWait support in user mode lacks a CPUID bit, but the instructions can be intercepted: no functional problem here.



Let's push it further: Migrating between AMD and Intel Systems

CPUID Method works for most features, but not all!

The Good:

- SSE generations, Monitor/MWait, RDTSC

The Bad:

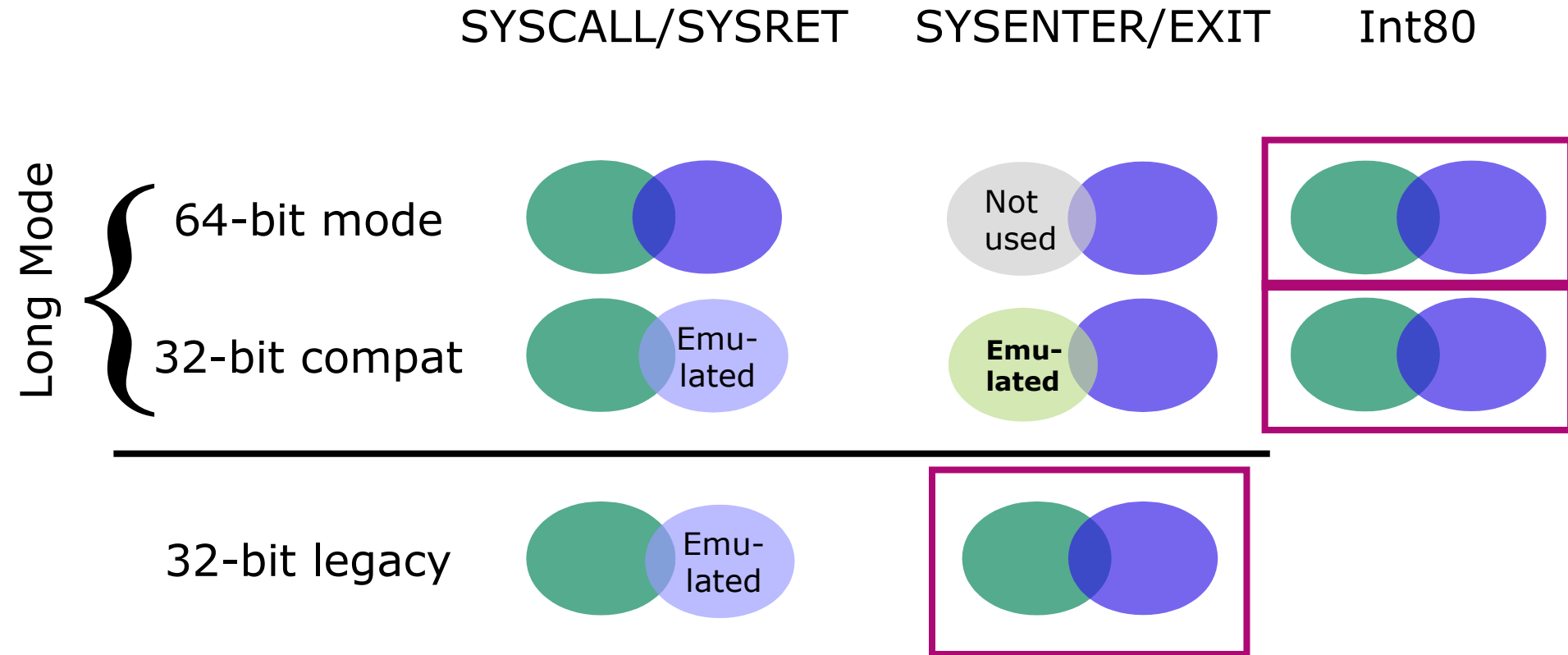
- SYSCALL/SYSENTER in various modes is more of a headache
- Windows does a FAR JMP into 64-bit mode, which then does SYSCALL: not a problem
- Linux is a problem. It picks SYSCALL or SYSENTER at boot, and this causes problems in migration of 64-bit mode guests. See next slide.

And the Ugly?

- FP ULP precision



SYSCALL and SYSENTER non-overlap in various subsets of Long Mode: which instruction do we want Linux to use for cross-vendor migration?



Performance Implications

- Emulation cost for syscalls in 32 bit OS'es are around 5%
- No emulation penalty on 64 bit
- Potential performance loss due to lowest common denominator feature set (very application dependent)



Floating Point

Nearly every FP instruction provides identical results on Intel and AMD. Anything that follows IEEE rounding rules is covered, but there are always exceptions!

- SSE reciprocal and reciprocal-square root approximation instructions: AMD and Intel provide different numbers of significant bits
 - RCPPS, RCPSS, RSQRTPS, RSQRTSS
- x87 instructions with infinite series implementations can return answers whose results differ in the units in the last place (ULP).
 - Trigonometric functions
 - Transcendental functions
 - Reciprocals
- Everything else (SSE1-SSE4 and most x87) gives identical answers obeying the IEEE754 rounding rules.



Software should be OK with the FPU differences!

Sane numeric algorithms should be OK:

- Good numeric analysis should account for variance in ULPs in considering numeric convergence
- Many libraries use SSE rather than x87, so are cross-vendor migration-safe by construction.

Are there insane algorithms that would fail?

- Study still in progress to find any surprises.
- How to write an intentionally migration-unsafe, legitimate algorithm is left as an exercise.



Conclusion

- By following a few simple steps you can maximize the size of your migration cloud
- Mixing AMD and Intel systems in the same cloud is as simple as having multiple CPU generations
- Performance impact is usually negligible
- Management tools should be used to deal with administrative aspects of guest migration



Trademark Attribution

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2009 Advanced Micro Devices, Inc. All rights reserved.

