Nesting KVM on s390x

… nesting nested virtualization on IBM z Systems ®

David Hildenbrand, Software Engineer Virtualization and Linux Development
26. August 2016, KVM Forum 2016

IBM

If you're not confused, you're not paying attention.

Tom Peters, Thriving on Chaos: Handbook for a Management Revolution

Trademarks

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at **ibm.com/legal/copytrade.shtml**

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
QEMU is a trademark of Fabrice Bellard.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

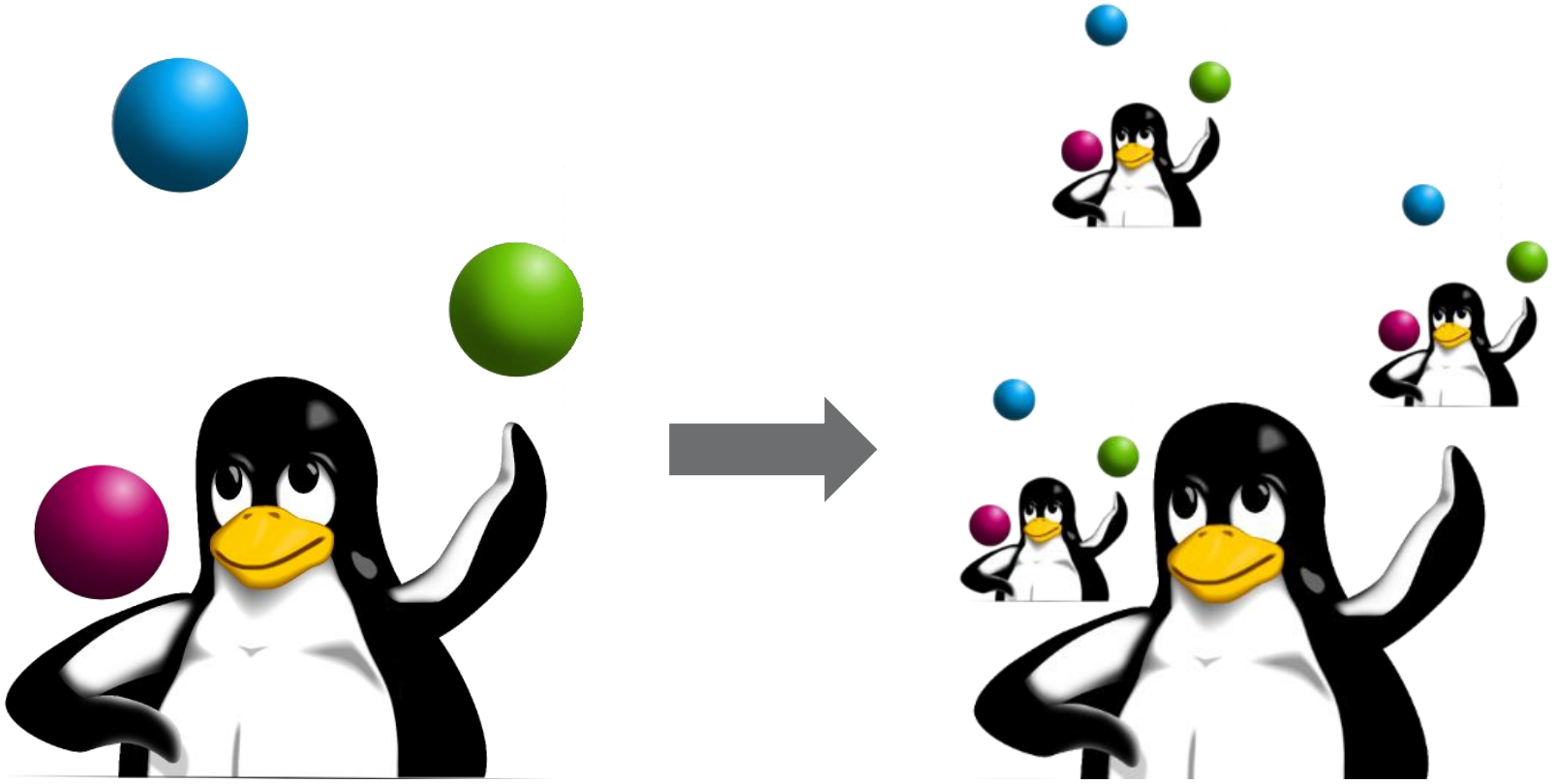
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

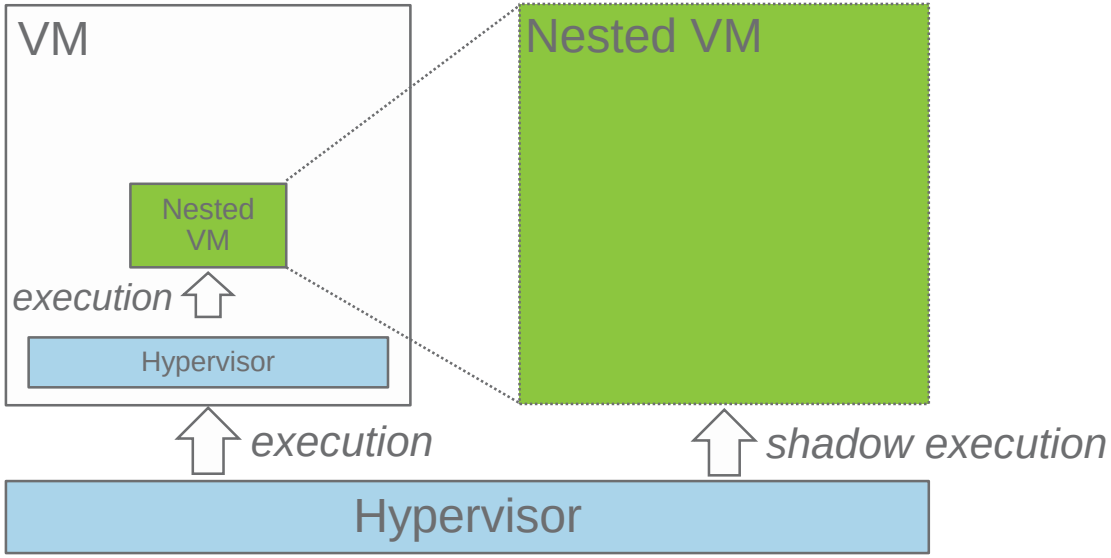
- Nested Virtualization
- Virtualization / KVM on s390x
- Nesting KVM on s390x
- Current status (Features, Migration, Security, Performance)
- Summary and Outlook
- Questions?

Nested Virtualization (1)



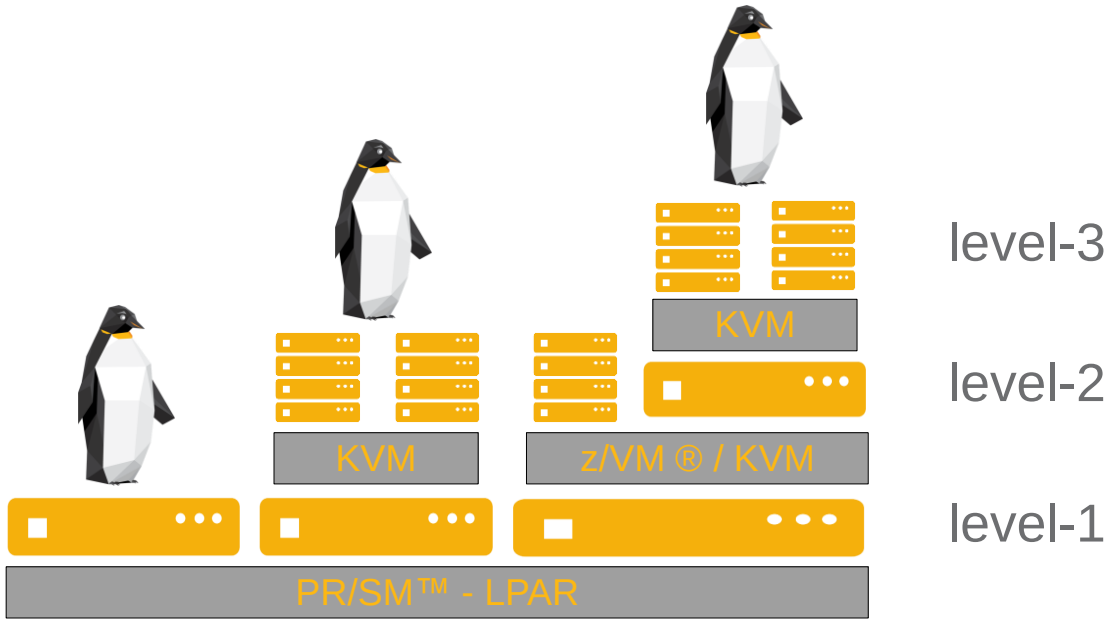
Nested Virtualization (2)

- Turn guest into hypervisor: *run virtual machines*
  - *Test / debug environment* (e.g. for new KVM releases)
  - *Simulate different hardware variants*
- Without HW support for nested virtualization
  - *Trap and emulate* (like „KVM-PR“ e.g. for PowerPC ®) *in guest*
  - *Emulate* HW virtualization (using HW virtualization) *in host*



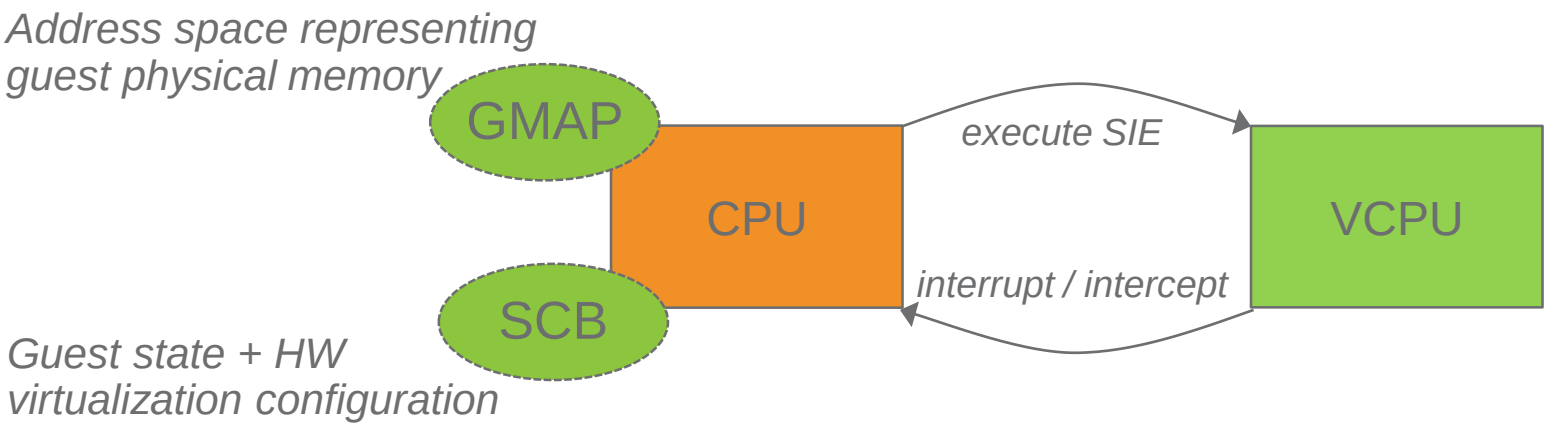
- Nested guest can also run nested guests ... *it usually simply cascades*
- Until now *only x86* emulates HW virtualization in KVM for its guest

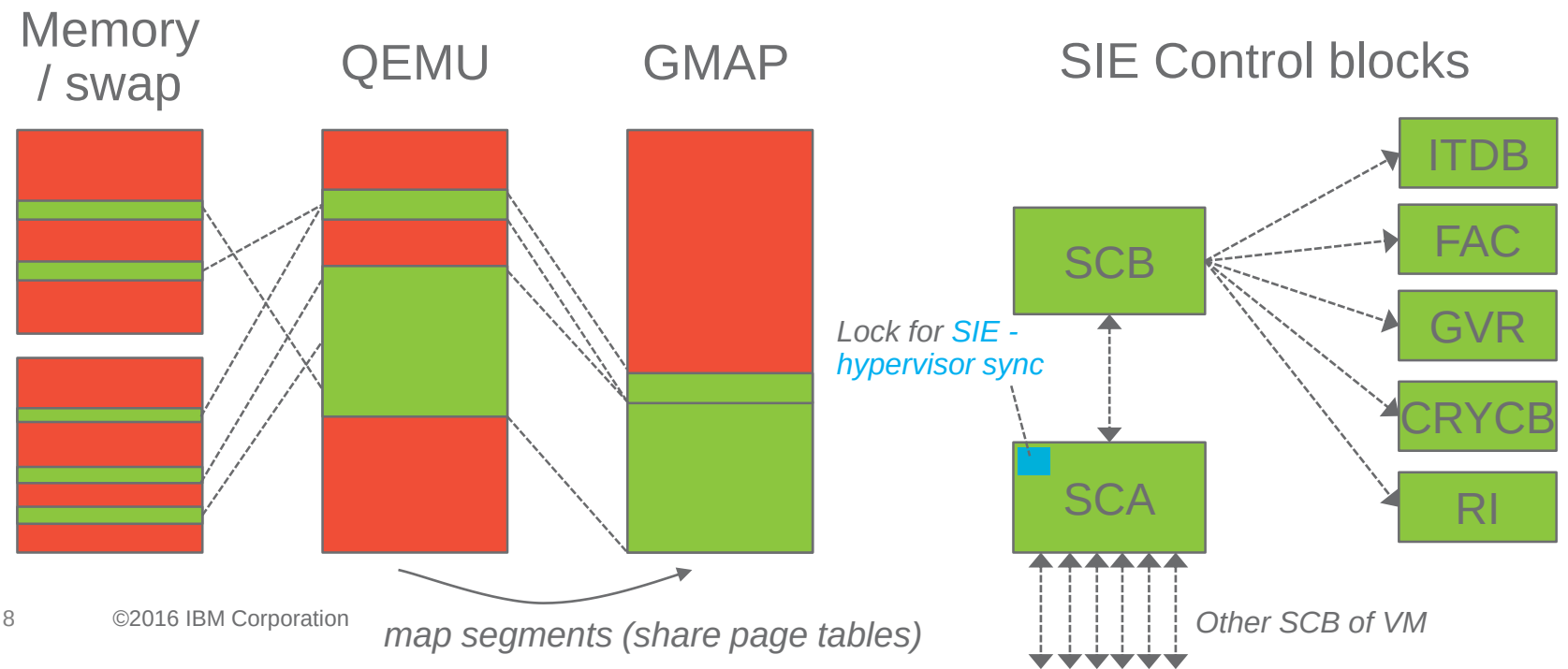
Virtualization / KVM on s390x (1)



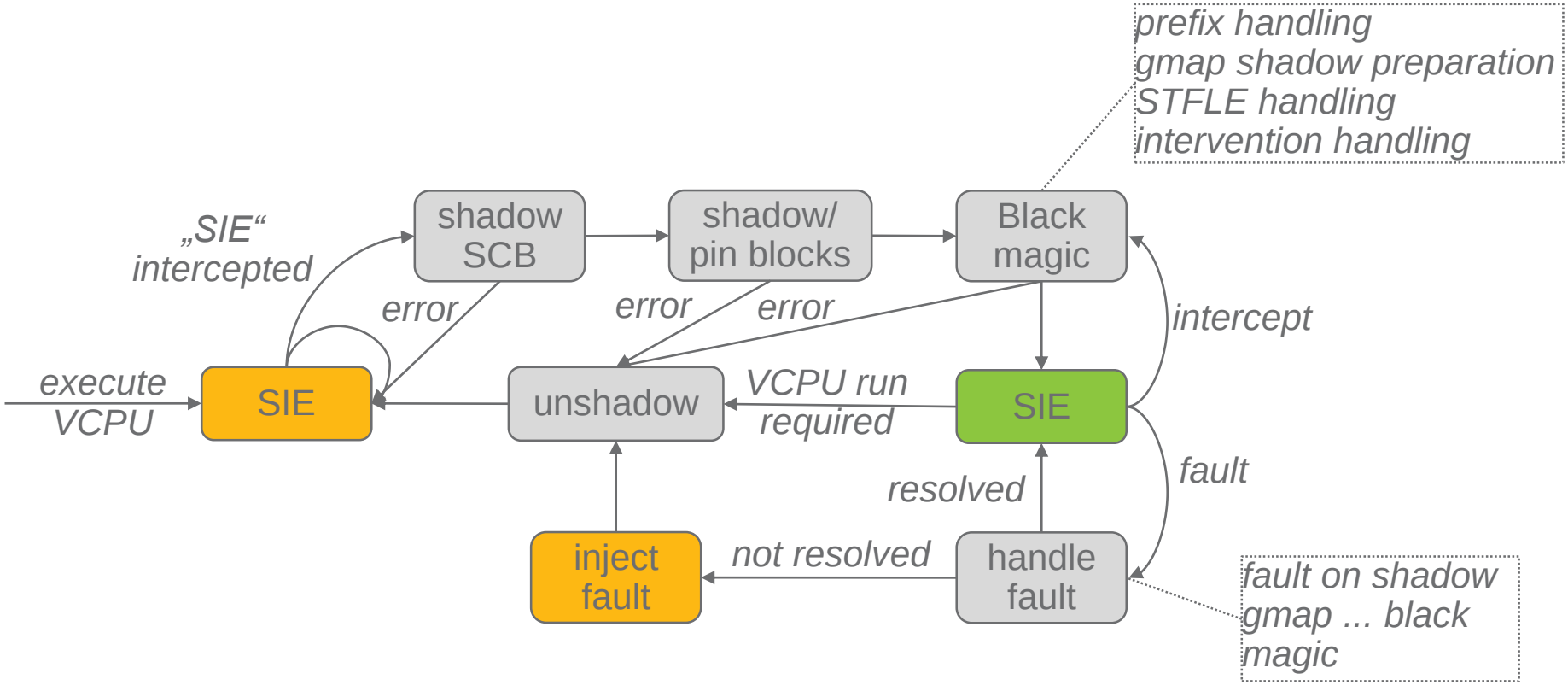
- *At least one level* of virtualization (logical partitioning)
- Hardware provides *support for two levels*
- *SIE (Start Interpretive Execution)* instruction is the entry point to HW virtualization
  - *Interpretes most instructions + guest interrupts*
- *SIE facilities* add aditional interpretation mechanisms (performance / features)

Virtualization / KVM on s390x (2)





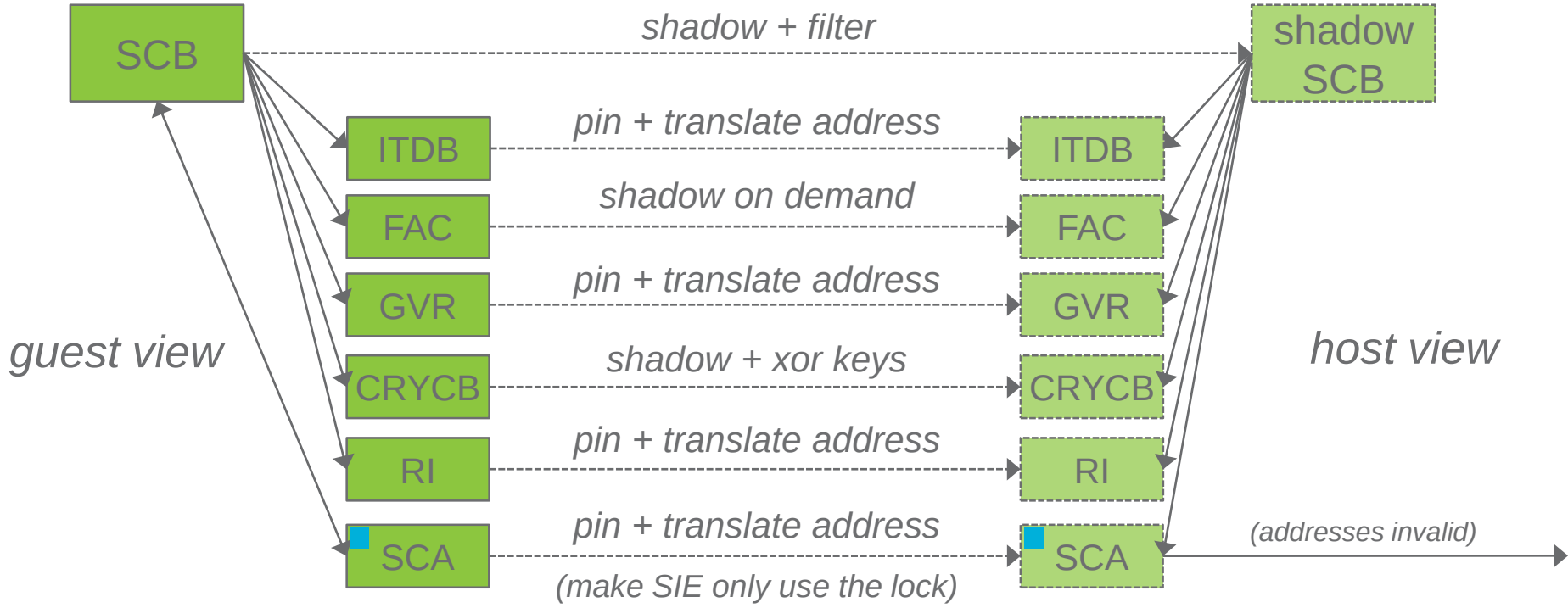
Nesting KVM on s390x (1)



- Emulation is called *Virtual SIE a.k.a vSIE*
- Control blocks / page tables contain *addresses only valid in the guest*
  - *Create shadows in the host*, containging valid host addresses
  - SCB vs. Shadow SCB
  - GMAP vs. Shadow GMAP

Nesting KVM on s390x (2)

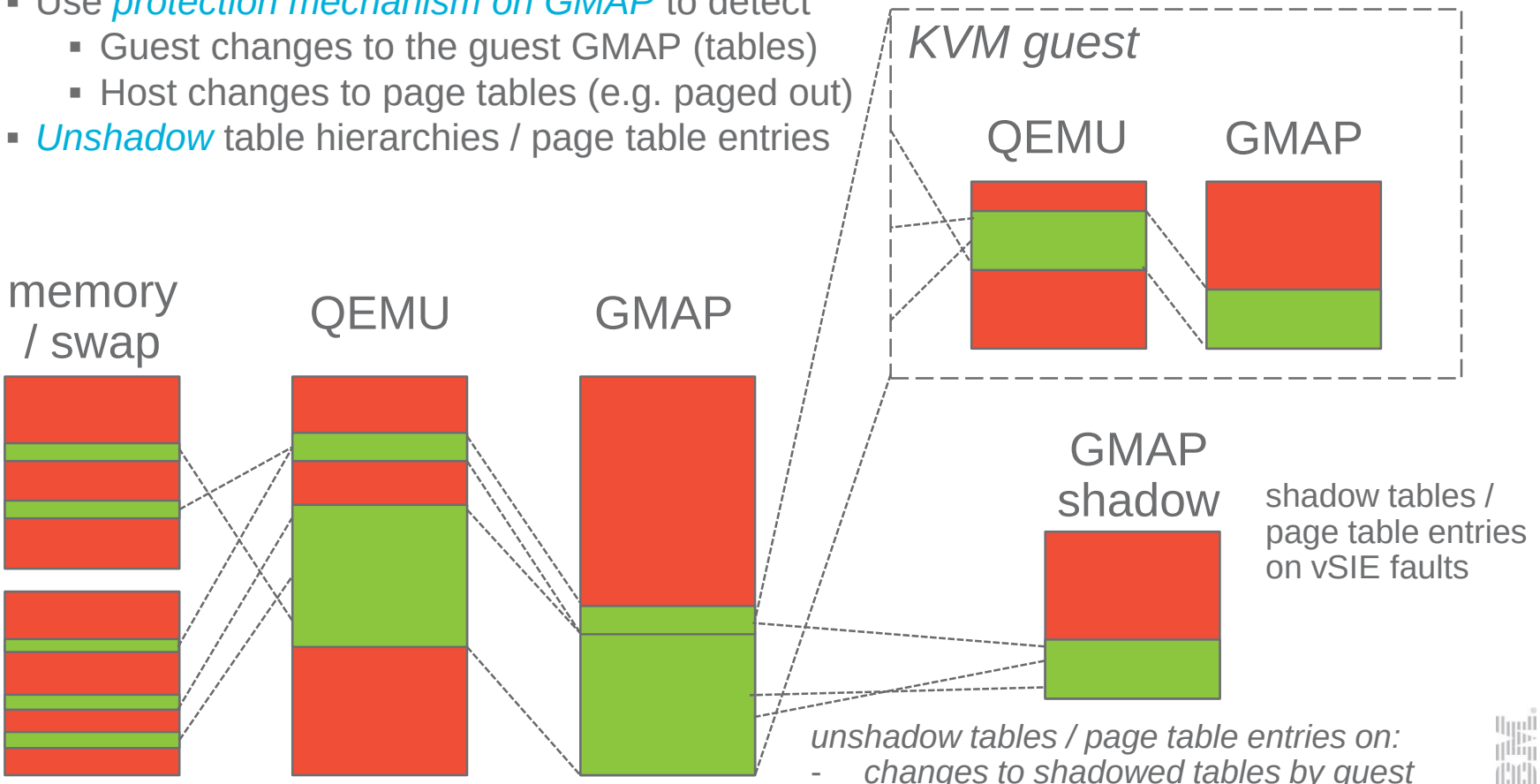
1. *Intercept SIE instruction* executed by KVM guest VCPU
2. *Validate/Copy/Filter* provided SCB, creating a *shadow SCB*



3. *Pin/Shadow/Filter satellite control blocks* referenced in the SCB
   - 31bit addresses in the SCB: shadow on DMA page ☹
4. *Execute the SIE* using the shadow SCB and shadow gmap

Nesting KVM on s390x (3)

4. *Fill/manage GMAP shadow* on *vSIE faults*
   - Create *shadow table hierarchy*
     - Walk guest provided tables by *reading in guest memory*
     - All tables are *initially empty* and *filled on demand (shadowing a lower level table)*
   - Lowest level (page tables) reference real host pages
   - Use *protection mechanism on GMAP* to detect
     - Guest changes to the guest GMAP (tables)
     - Host changes to page tables (e.g. paged out)
   - *Unshadow* table hierarchies / page table entries



Nested KVM on s390x (4)

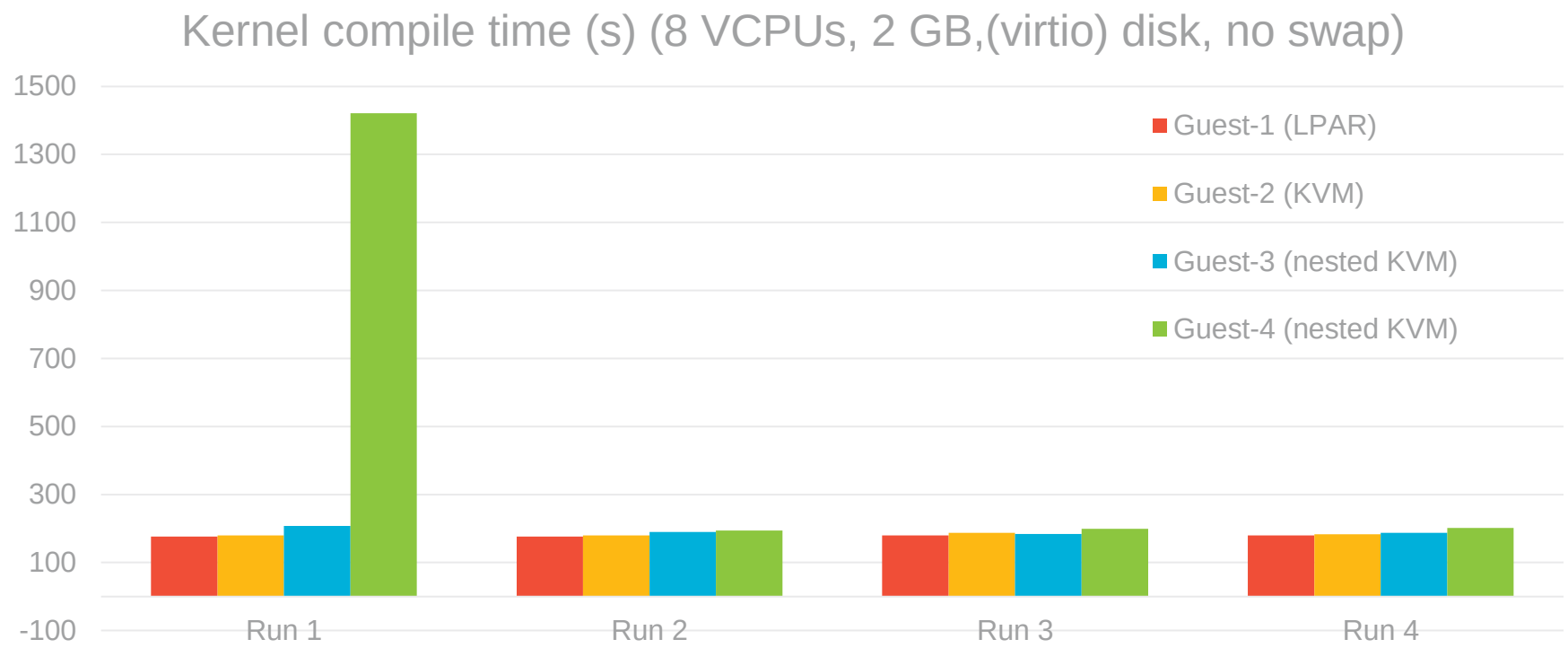
5. *Re-execute the SIE* as long as possible (VCPU run not required)
6. Inject interrupts *into the KVM VCPU guest only* if required (due to vSIE faults)
   - We never inject anything into the nested KVM guest VCPU
7. *Unshadow/unpin* control blocks
   - Unpin sattelites only – *no other blocks have to be unshadowed*

8. *Re-execute KVM guest VCPU*

Current state

- >= 248 CPUs, transactional execution, vector registers, huge pages (1M, 2G) ...
  - Basically everything a KVM guest has, except CMM because ...
- vSIE doesn‘t support all SIE facilities (e.g. CMMA or SIGP interpretation like z/VM)
  - shadow *table entries could be empty* although they are not in the shadowed one
  - SCA contains pointers to invalid SCBs – shadowing is not an option
- No, known bugs – still *„kvm.nested=1“* required for now
- *No know „incompatibilities“* with the SIE specification (okay, there is a small one ...)
- *CPU model support* required to turn it on (*„-cpu host“*)
- *Migration simply works*: guest memory contains all nested guest state
  - Shadow structures (GMAP, SCB ...) are silently recreated on the new host
  - GMAP shadow code should be able to deal with *user fault just fine*
- *Really hard to break out of vSIE, even into its hypervisor:*
  - We don‘t emulate any vSIE instructions – *SIE handles everything for us*
  - We don‘t inject any vSIE interrupts – *SIE handles everything for us*
  - When shadowing, we *heavily filter the SCB*, to not allow e.g. strange addressing modes
  - GMAP shadow is *based on GMAP only*

Performance



- *LPAR: 8 CPUs (not dedicated), 8 GB, SCSI disks, no swap*
- *First memory access is expensive*
  - The GMAP shadow has to be filled on first memory access
  - Building a GMAP shadow on a GMAP shadow is horribly expensive (Guest-4)
- Once memory is faulted into the gmap shadow, *overhead is quite small*
  - *Lockless lookup/reuse* of shadow SCB (to avoid TLB flush) + shadow gmap
- Kernel src on *multi-paravirtualized disk* via virtio-blk
  - Rebooting the compiling guest (clear caches) didn‘t affect compile times

How deep can we go?

Source: ***http://paisleymagic.storenvy.com****/collections/236287-nesting-dolls/products/1447594-penguins-nesting-eco-friendly-doll-russian-dolls-matryoshka*

Summary and Outlook

- I was able to *start a kernel in guest-6* ... while having lunch
  - Can we improve the gmap shadow/unshadow + pagefault pingpong somehow?
- KVM is now able to run with a *minimum amount of SIE facilities*
- We found *one random memory overwrite* + minor bugs in KVM code
- Can we *reduce the amount of DMA pages*?
  - This would allow us to keep more shadow SCBs in the cache
- Can we *reuse data in the shadow SCB*, not shadow/check everything again?
- *„kvm.nested=1“*, can it ever be dropped completely?
- *CPU model support* in QEMU to finally turn it on
- Support all *new HW features as KVM support is added*


IBM
Thank you!
ibm.com/linux
IBM