Cross-Platform Guest Support

June 2008

Paul Knowles

TRANSITIVE™

Agenda

- Introduction
  - Who are Transitive?
  - What is QuickTransit?
  - The KVM and QuickTransit Solution
- Technical Challenges
  - What's inside a guest VM?
  - VM Initialisation
  - Controlling guest page tables from userspace
  - Shadowing foreign page tables
  - Paravirtualizing a foreign O/S
- Current Work
  - Changes to KVM
  - Possible deployment scenarios
  - Technology demonstration

TRANSITIVE™

Who are Transitive?

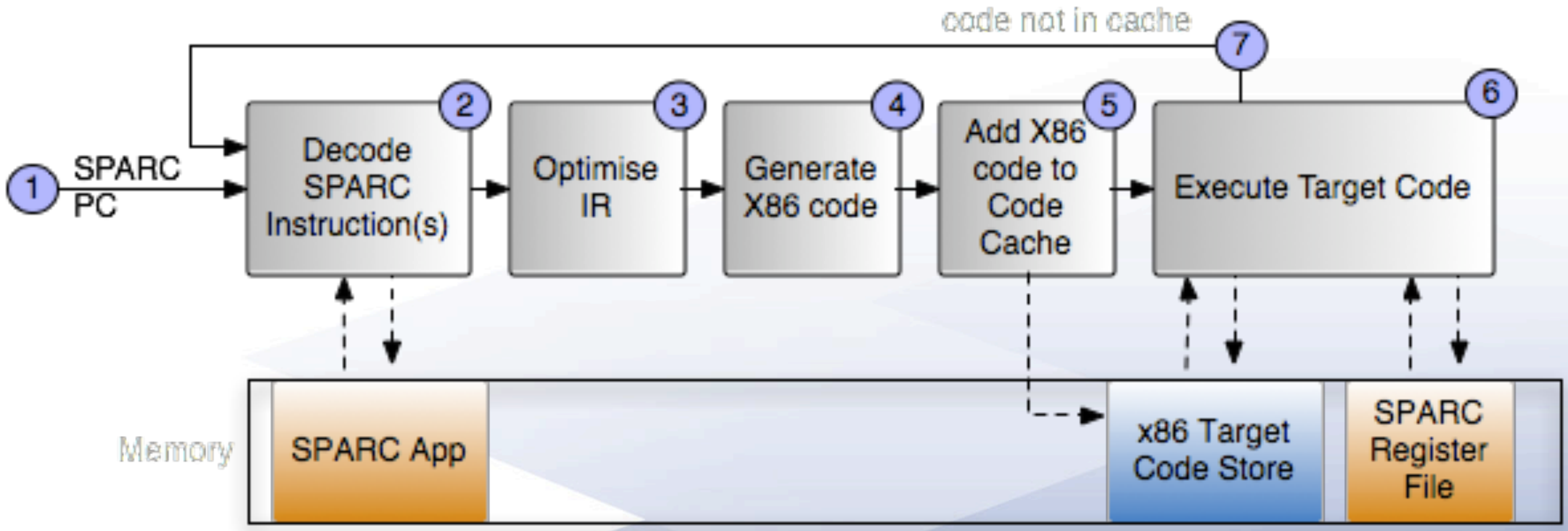
- Start-up company spun out of the University of Manchester
- Engineering in Manchester, UK (Around 90 people)
- Corporate headquarters in Los Gatos, California
- Best known for being the company behind Apple’s Rosetta
- Mission: “Every software application runs on every hardware platform”

TRANSITIVE™

What is QuickTransit?

Dynamic translation engine

- Translates from one CPU architecture to another
- Has multiple modes of translation and optimises over time

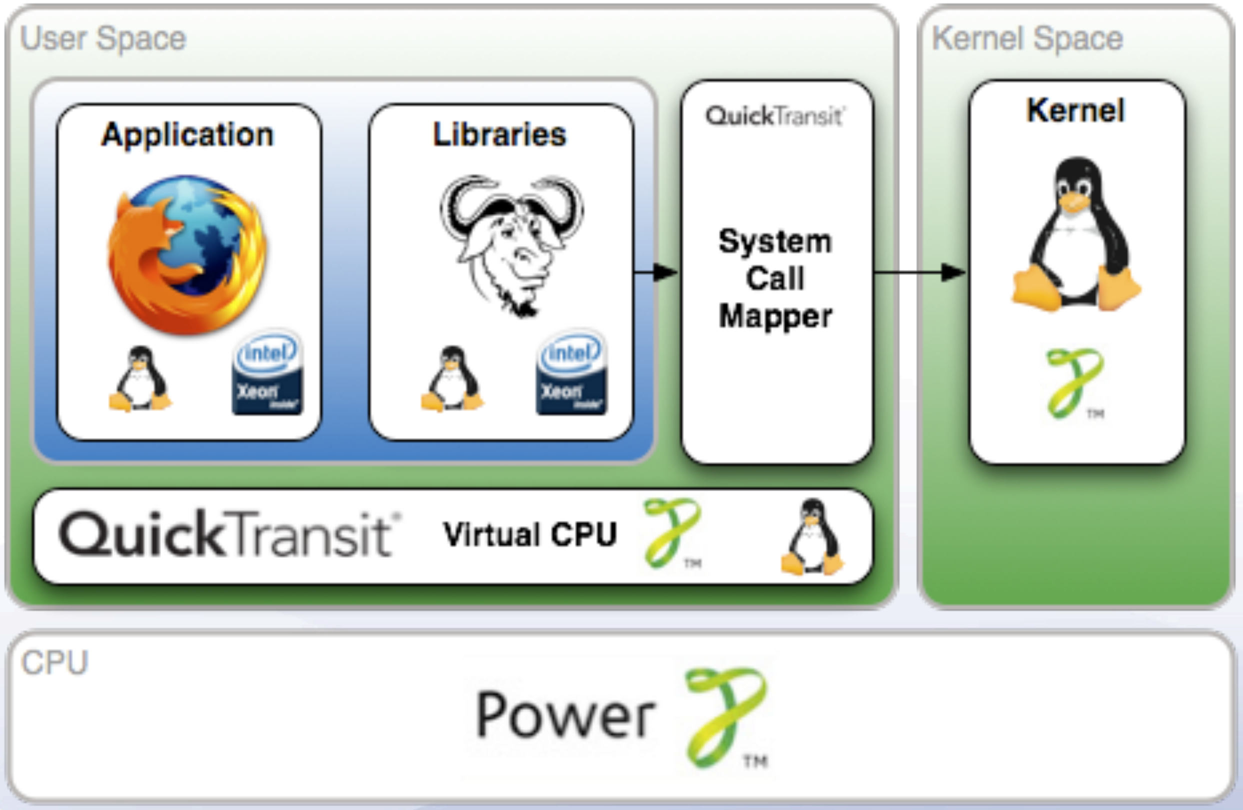


TRANSITIVE™

QuickTransit

- Normal user space application, requires no modification of the host kernel
- Allows applications compiled for one architecture to run without any modification on a different architecture

IBM PowerVM Lx86

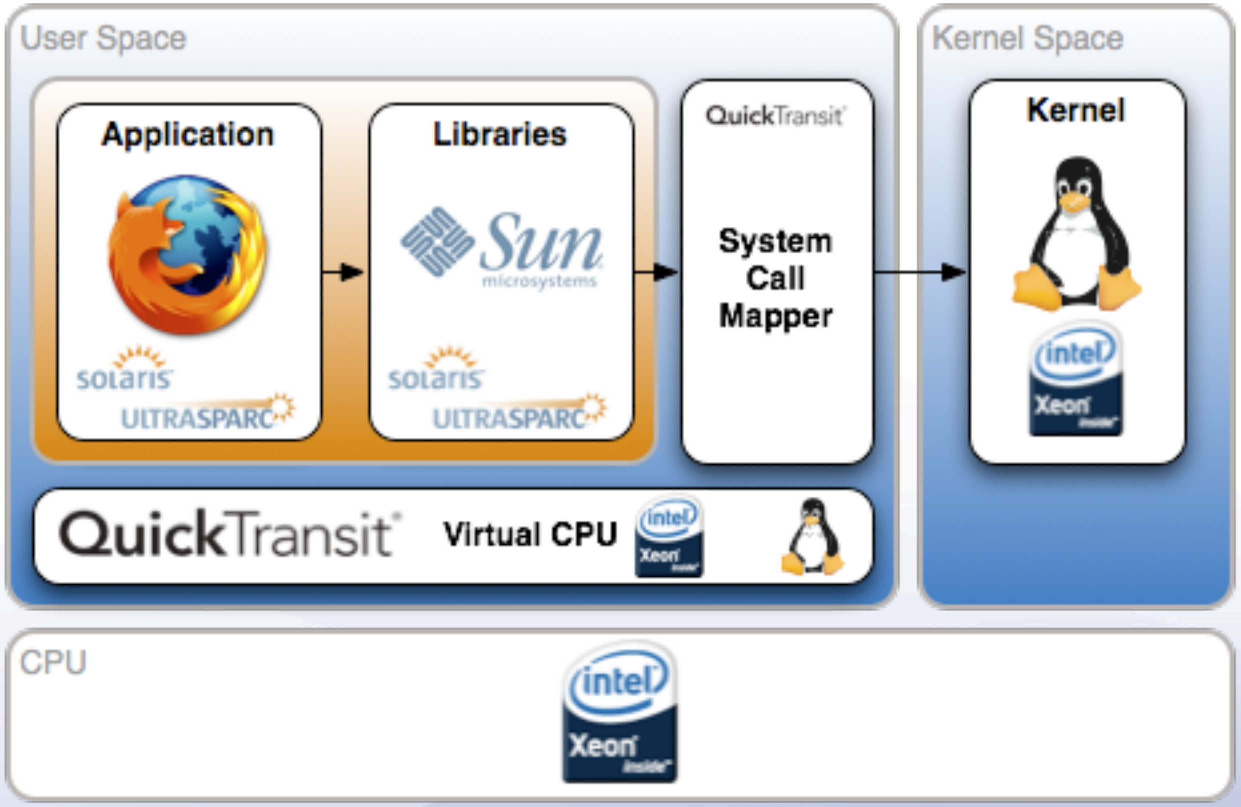


TRANSITIVE™

Can translate between OSes

- Can also translate between different operating systems
- In this example QuickTransit provides the features of Solaris on Linux without modifying the host or the guest application

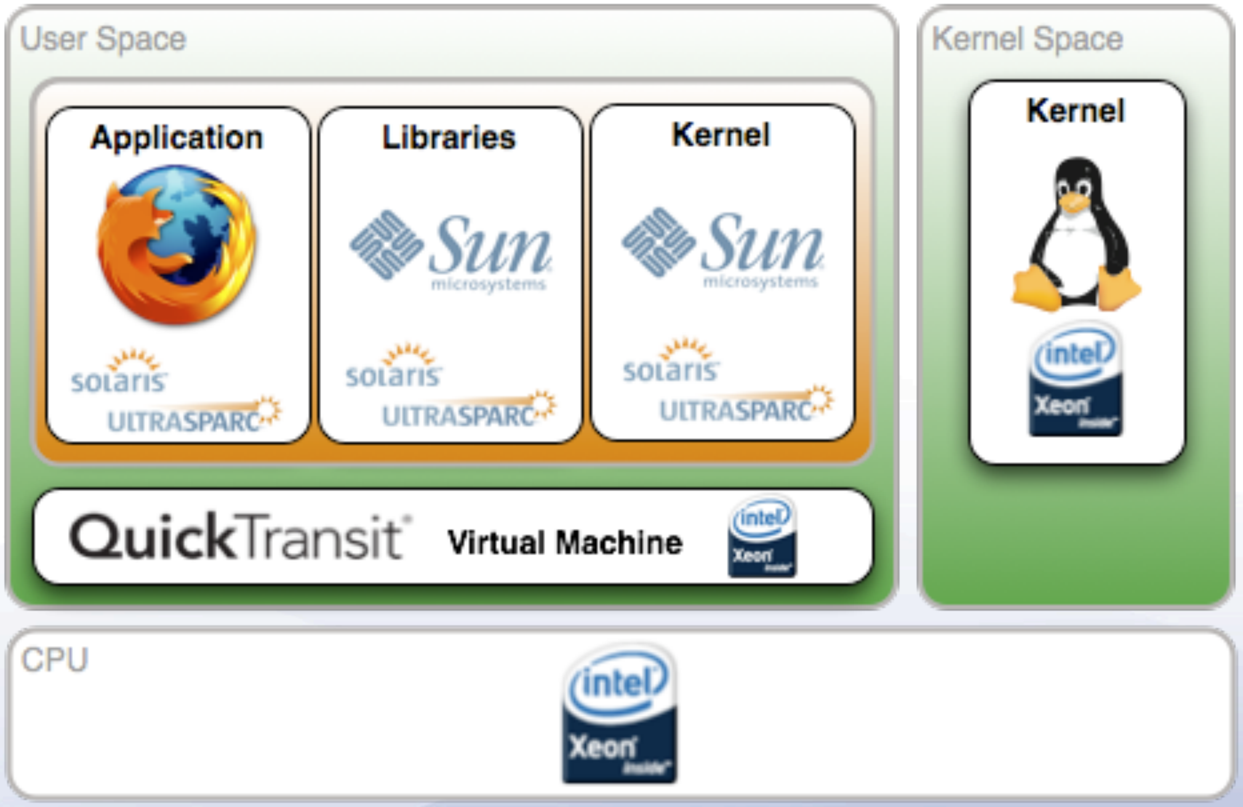
QuickTransit **Solaris/SPARC** to **Linux/x86-64**



TRANSITIVE™

How about translating the OS?

- Translate everything. Boot loader, kernel, etc..
- No longer have to map system calls
- Have to provide hardware emulation instead
- MMU emulation can be slow without hardware assistance



TRANSITIVE™

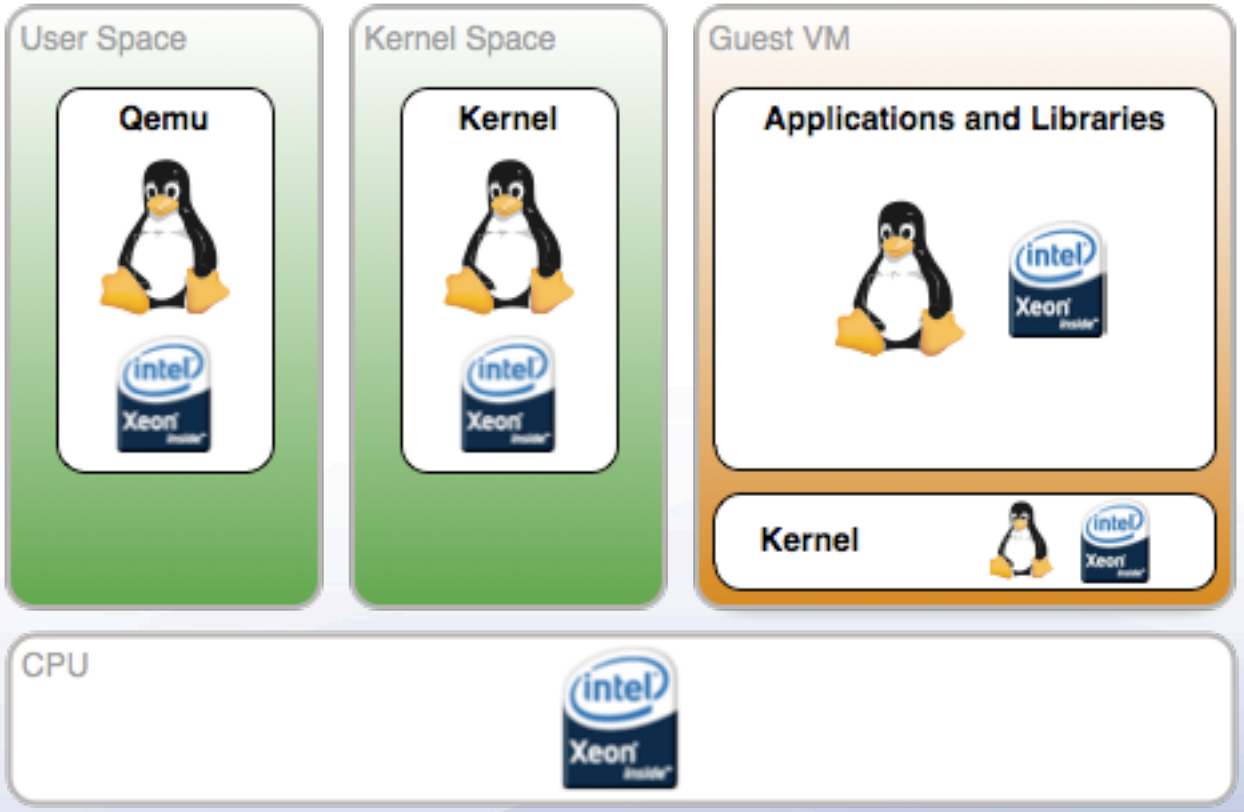
So what does this have to do with KVM?

- Use KVM to separate guest from translator address space
- Allows QuickTransit to use hardware for address mapping
- QuickTransit replaces QEMU as the userspace component
- QuickTransit provides hardware emulations

TRANSITIVE™

KVM

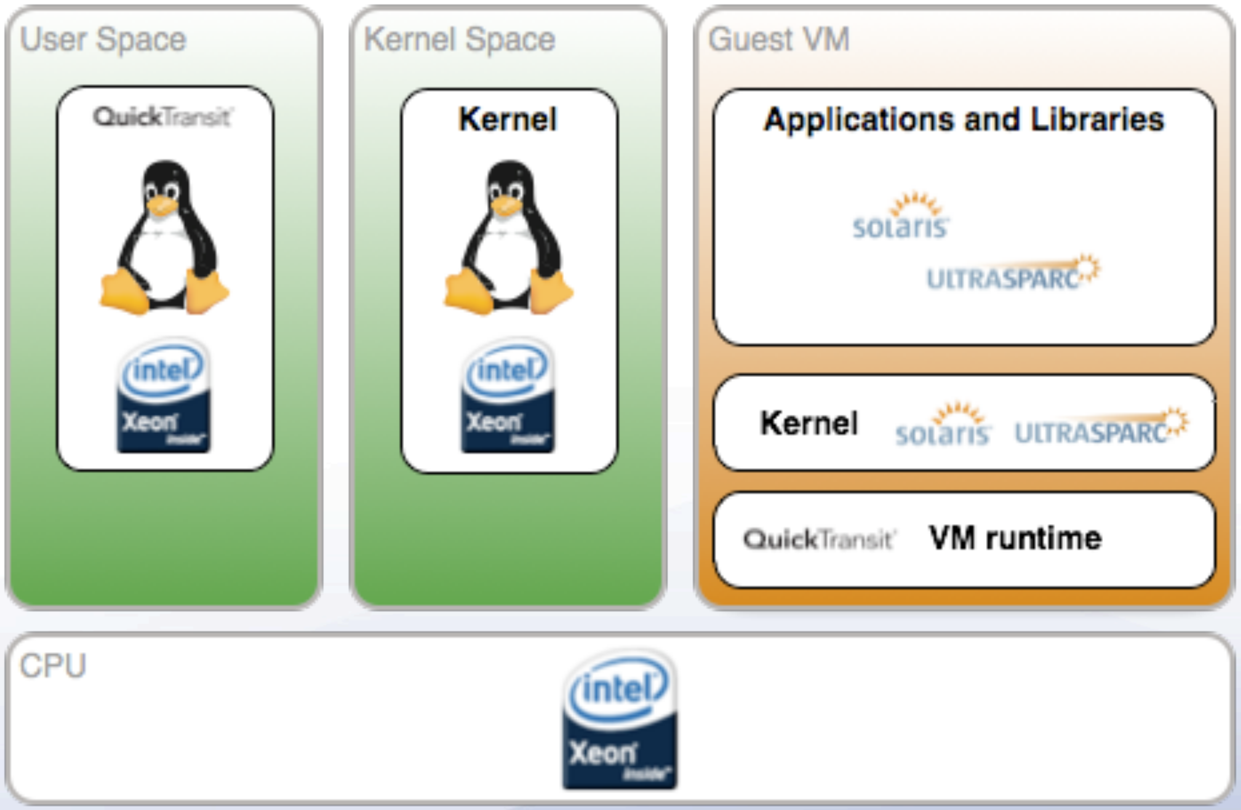
- QEMU operates as the userspace
- Provides the hardware emulation
- The guest VM is very similar to the machine it is actually running on



TRANSITIVE™

KVM + QuickTransit

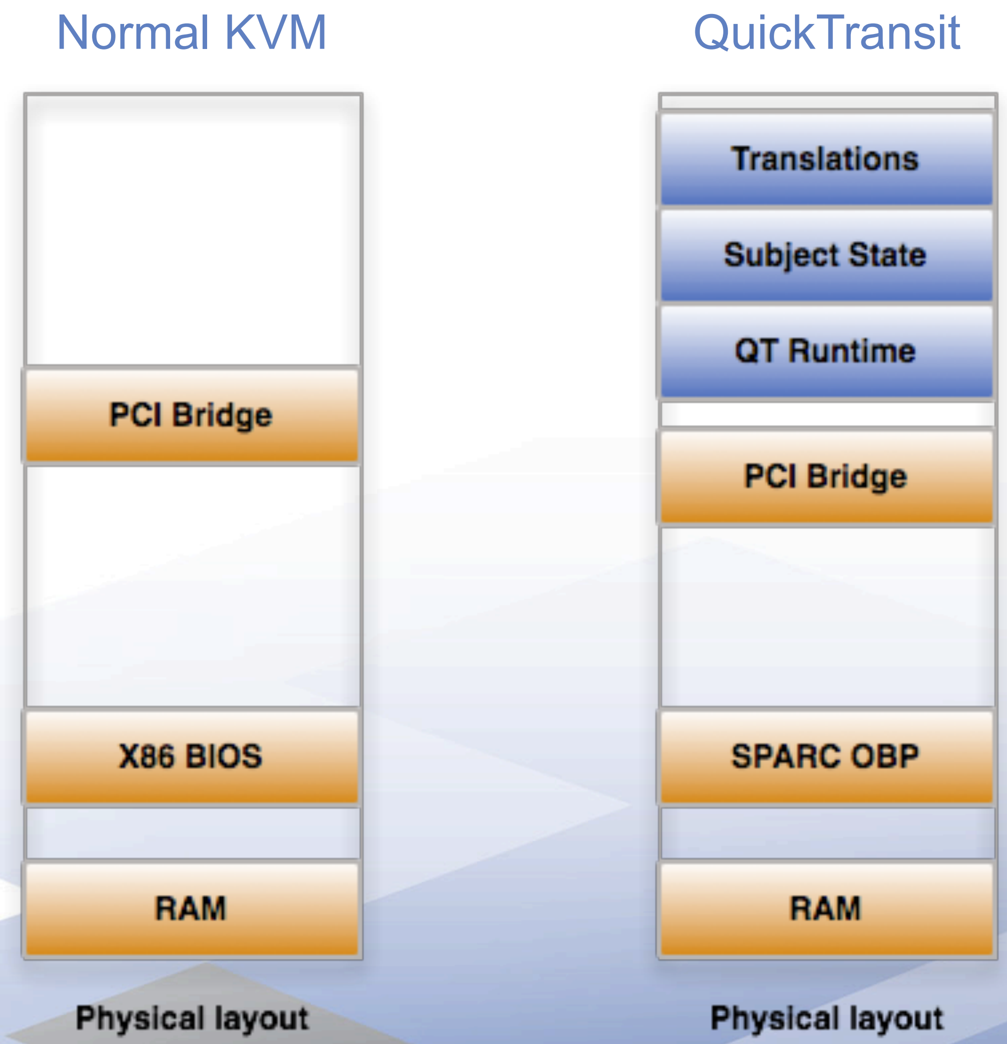
- QuickTransit is the userspace component
- QuickTransit provides hardware emulation
- The guest VM is nothing like the target machine



TRANSITIVE™

What's inside the VM?

- The guest VM has the physical layout of the machine under translation
- Along with additional mappings for the translator machinery
- Need to map in translator code and page tables, subject machine state (register banks), and the actual translations of the code



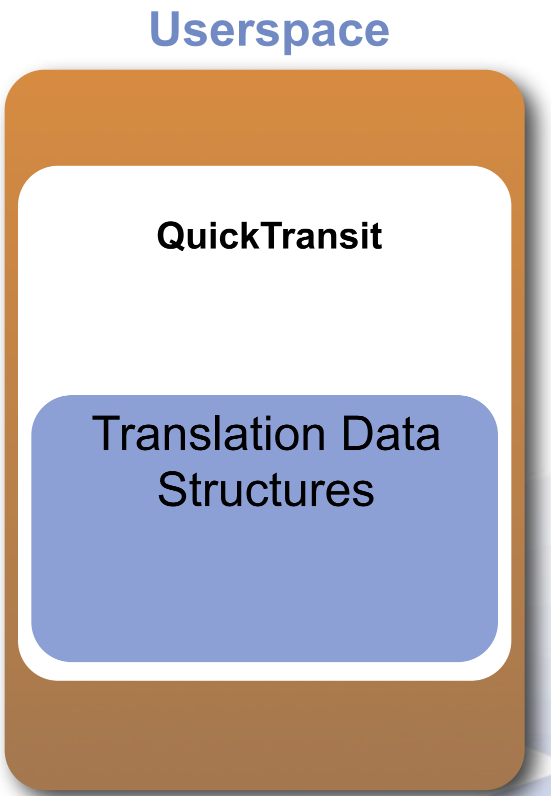
TRANSITIVE™

VM Initialization

- Different to how the guest VM is normally initialized
- Describe the physical layout of the machine under translation
- Also need to map in extra bits for the translator
- VM started in 64-bit mode with paging enabled
- No need to boot through the normal X86 boot sequence

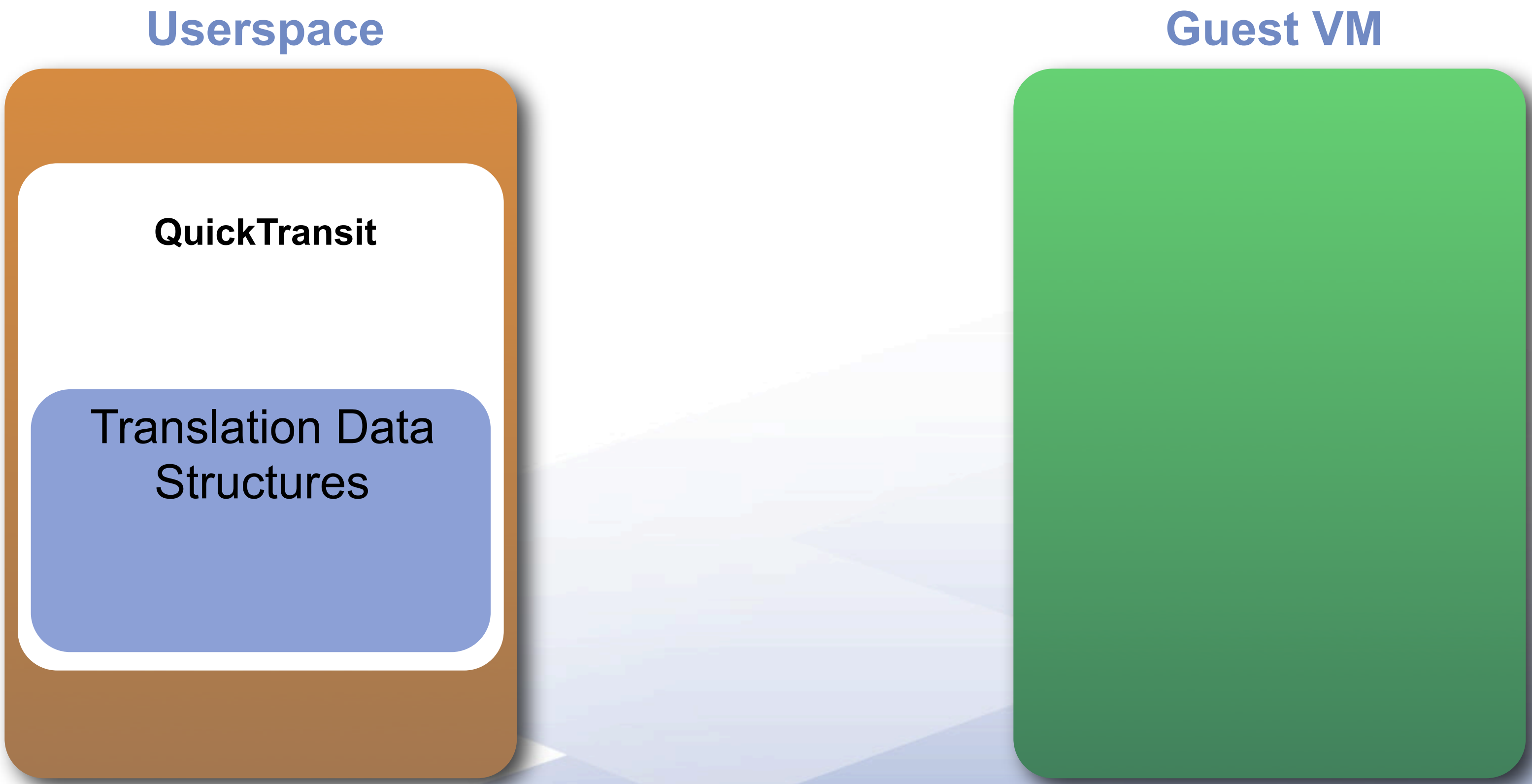
TRANSITIVE™

How does it work?

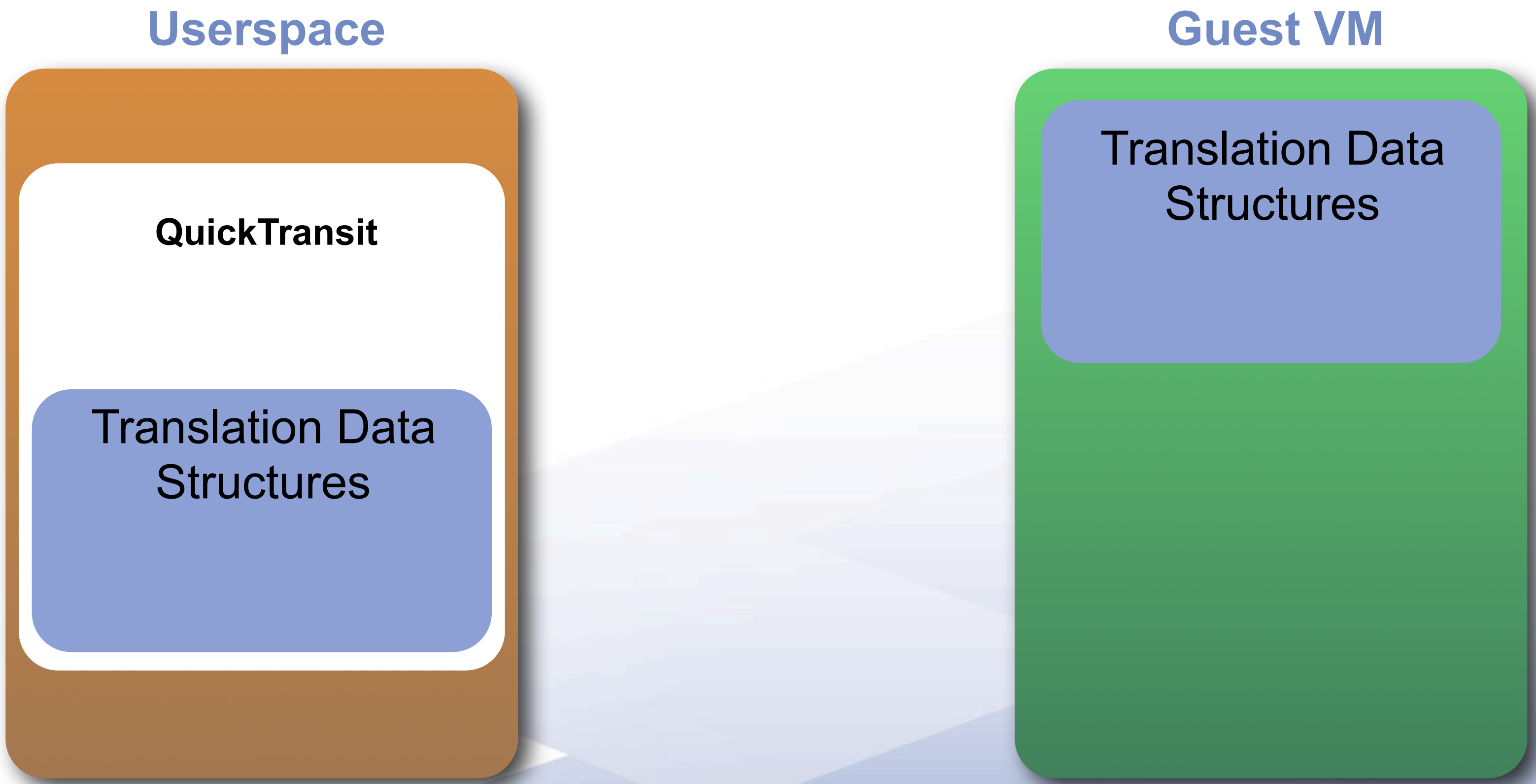


TRANSITIVE™

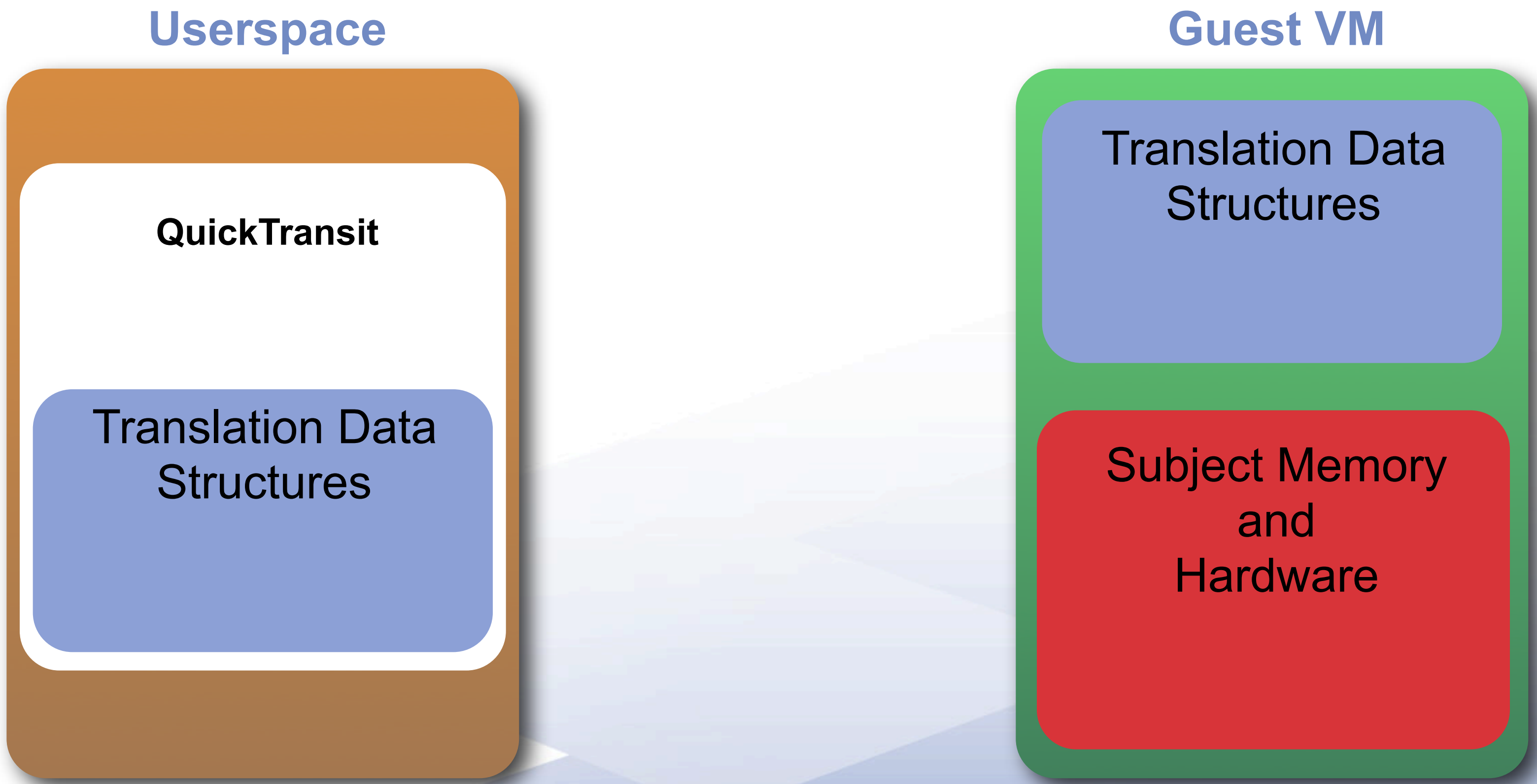
How does it work?



How does it work?

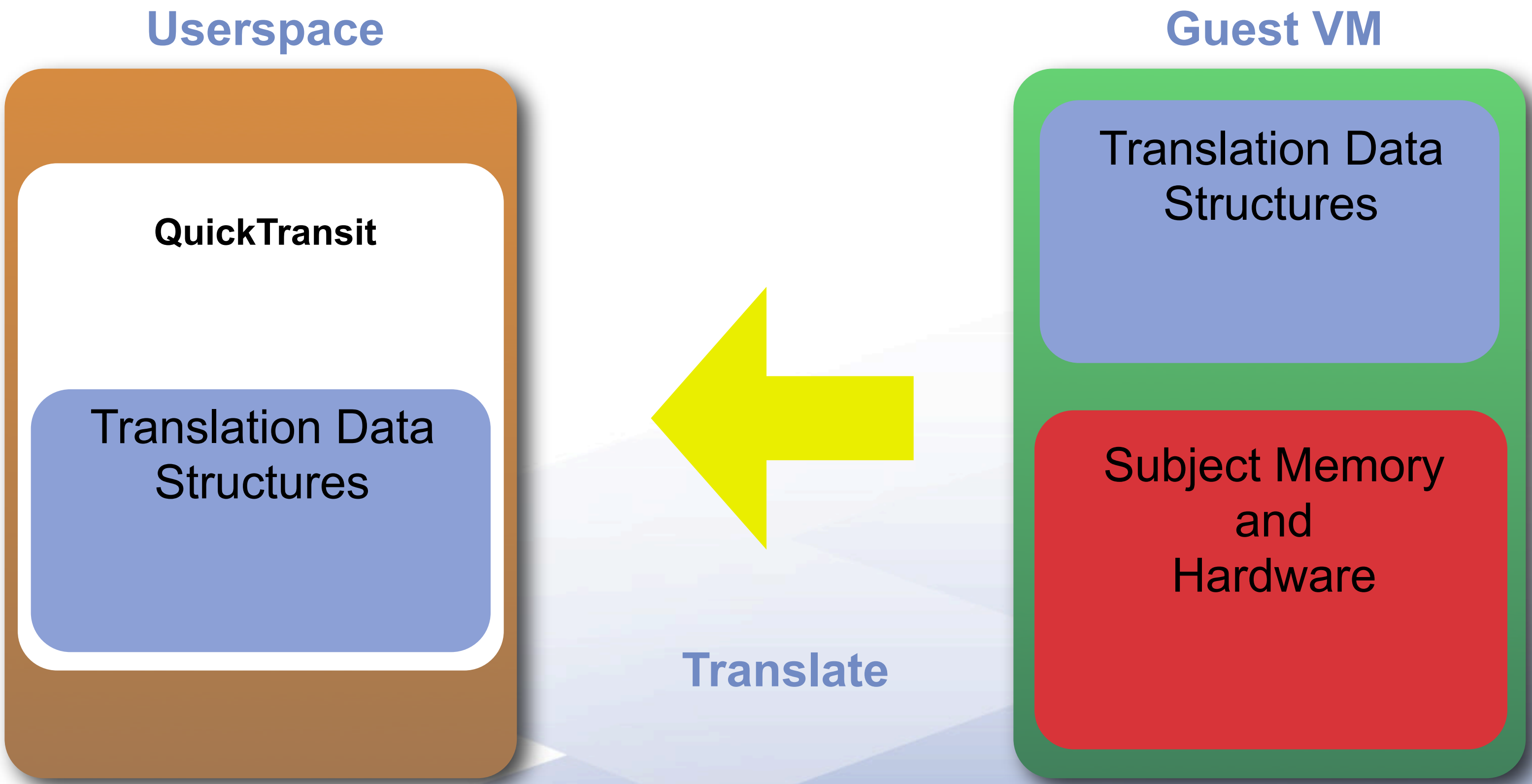


How does it work?

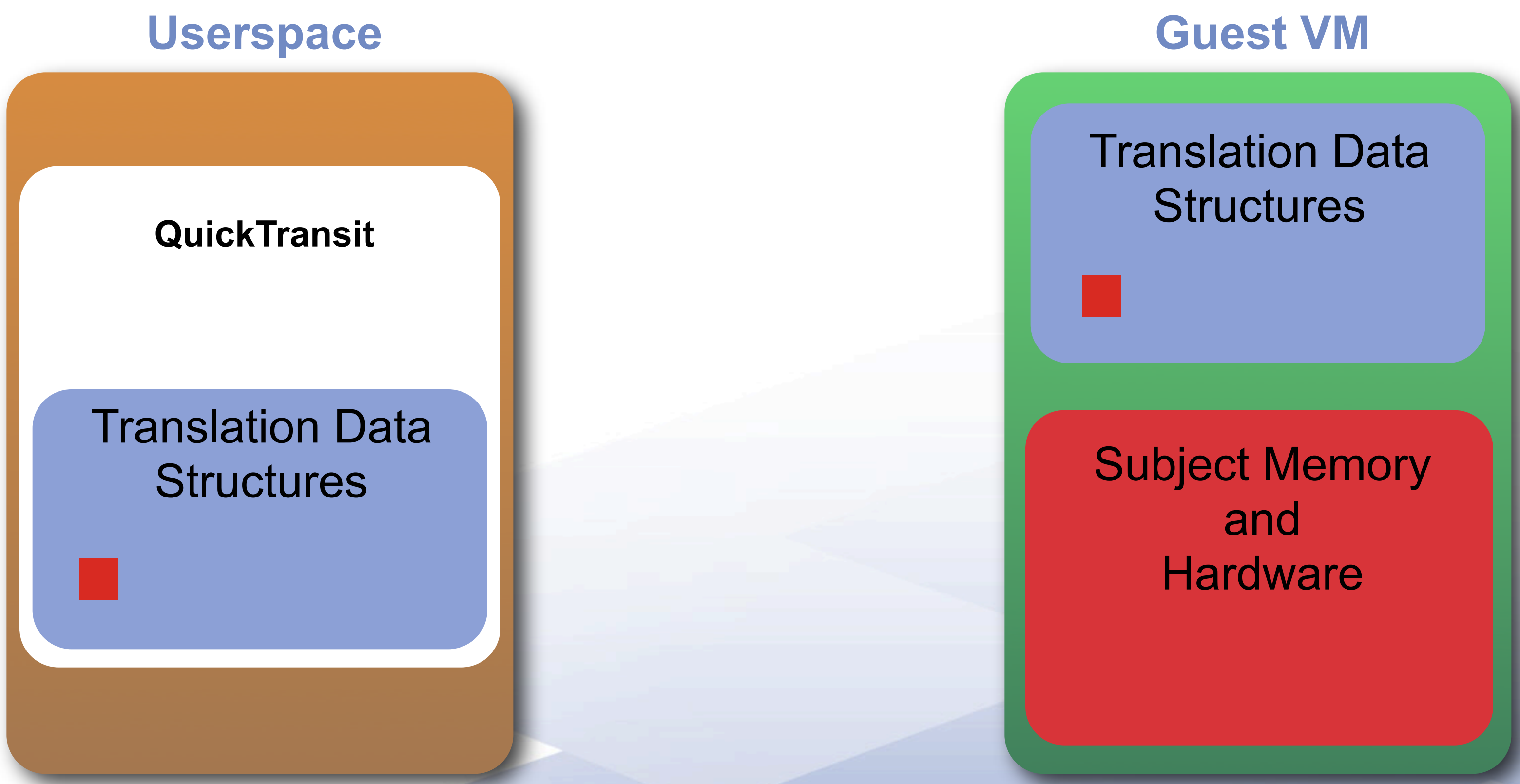

Userspace
QuickTransit
Translation Data Structures
Guest VM
Translation Data Structures
Subject Memory and Hardware

TRANSITIVE™

How does it work?

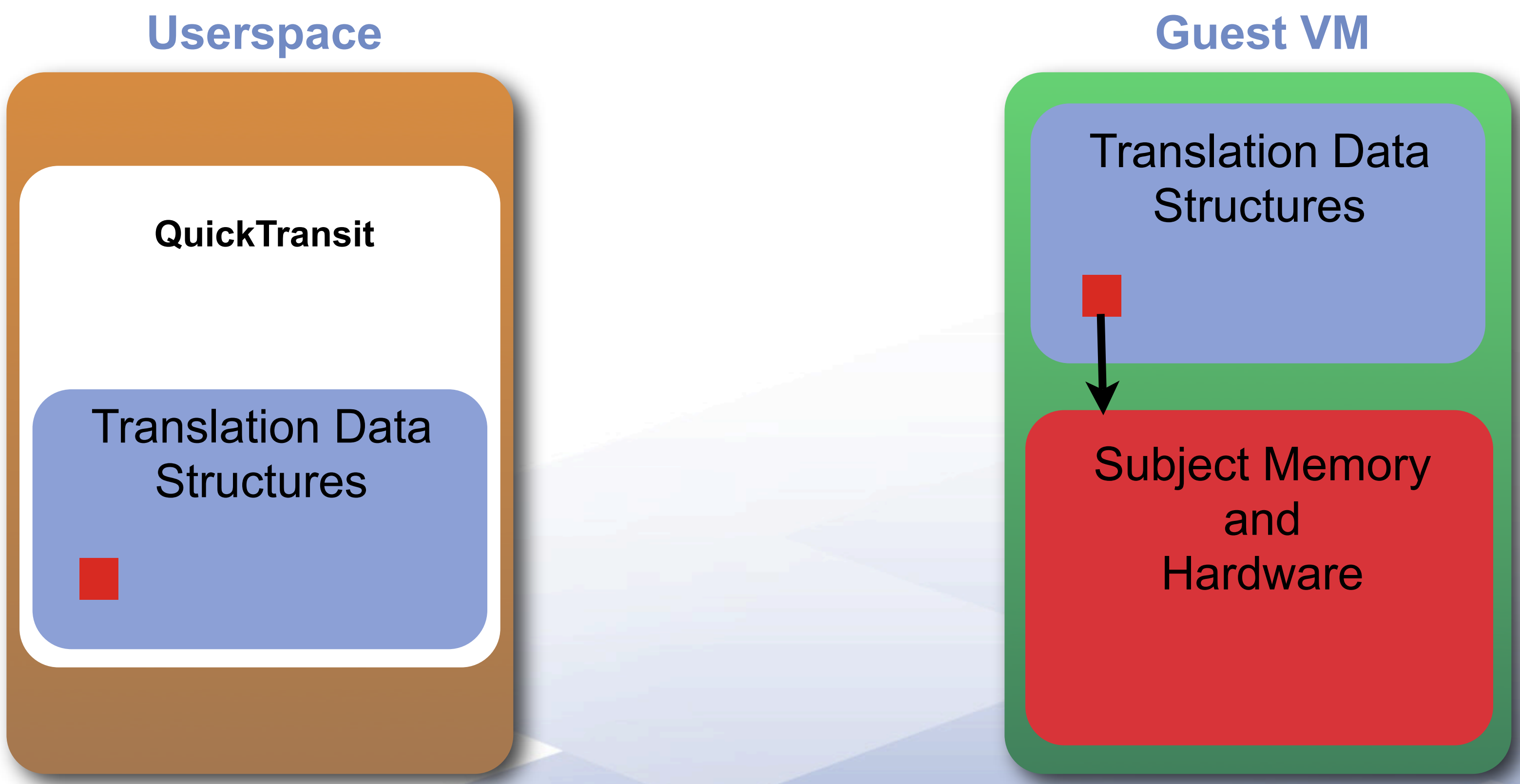

Userspace
QuickTransit
Translation Data Structures
Translate
Guest VM
Translation Data Structures
Subject Memory and Hardware

How does it work?


Userspace
QuickTransit
Translation Data Structures
Guest VM
Translation Data Structures
Subject Memory and Hardware

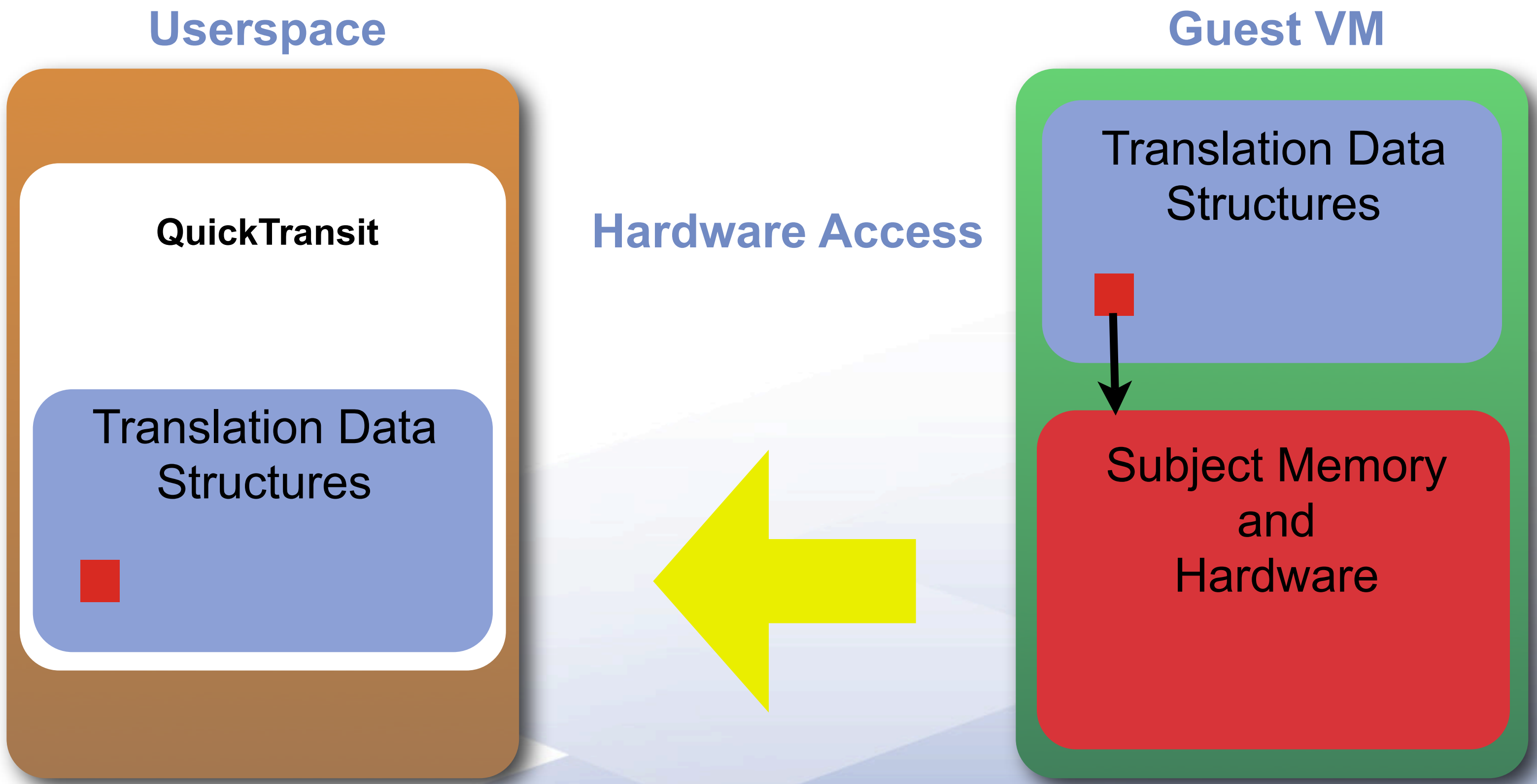
TRANSITIVE™

How does it work?

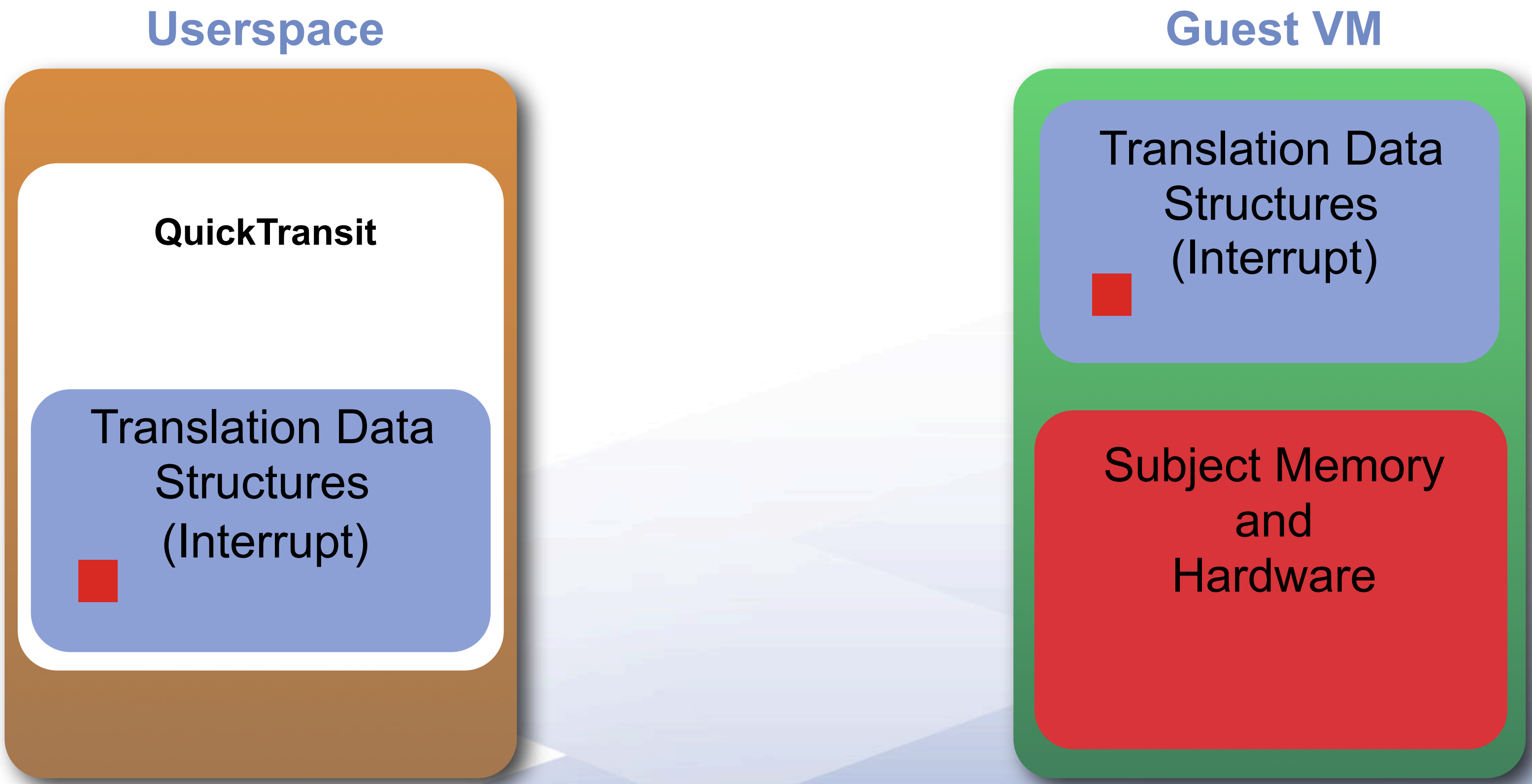

Userspace
QuickTransit
Translation Data Structures
Guest VM
Translation Data Structures
Subject Memory and Hardware

TRANSITIVE™

How does it work?

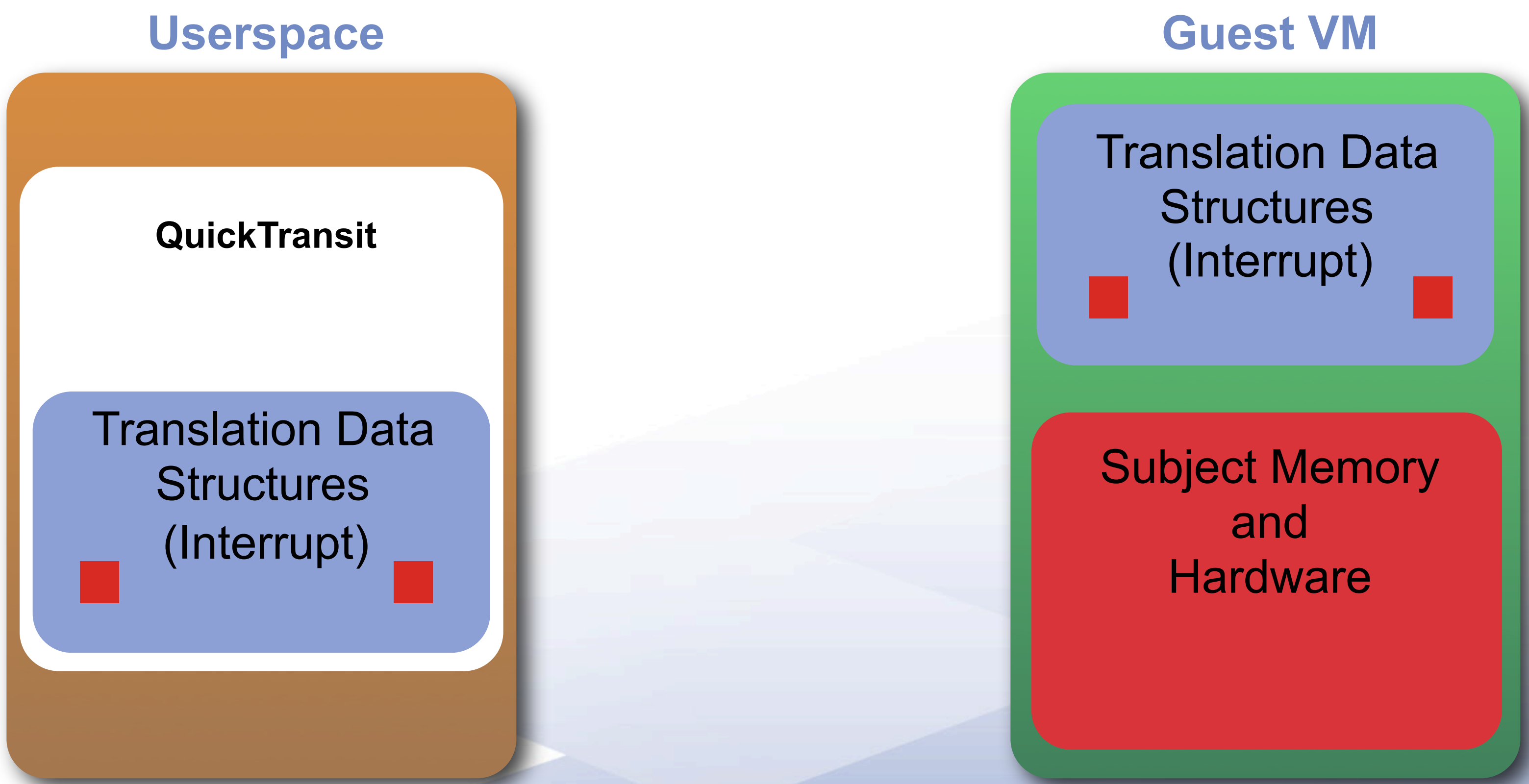

Userspace
QuickTransit
Translation Data Structures
Hardware Access
Guest VM
Translation Data Structures
Subject Memory and Hardware

How does it work?

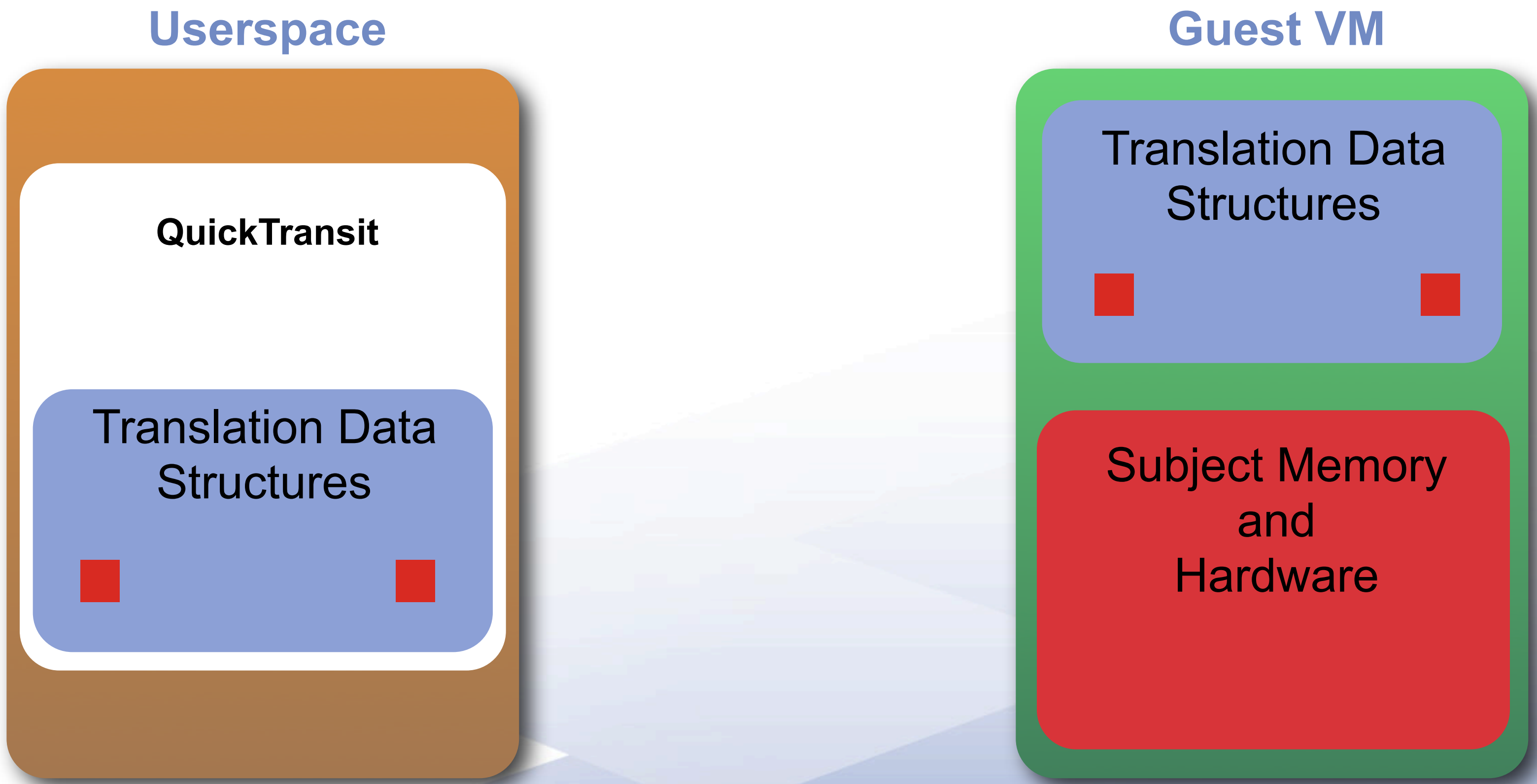

Userspace
QuickTransit
Translation Data Structures (Interrupt)
Guest VM
Translation Data Structures (Interrupt)
Subject Memory and Hardware

TRANSITIVE™

How does it work?

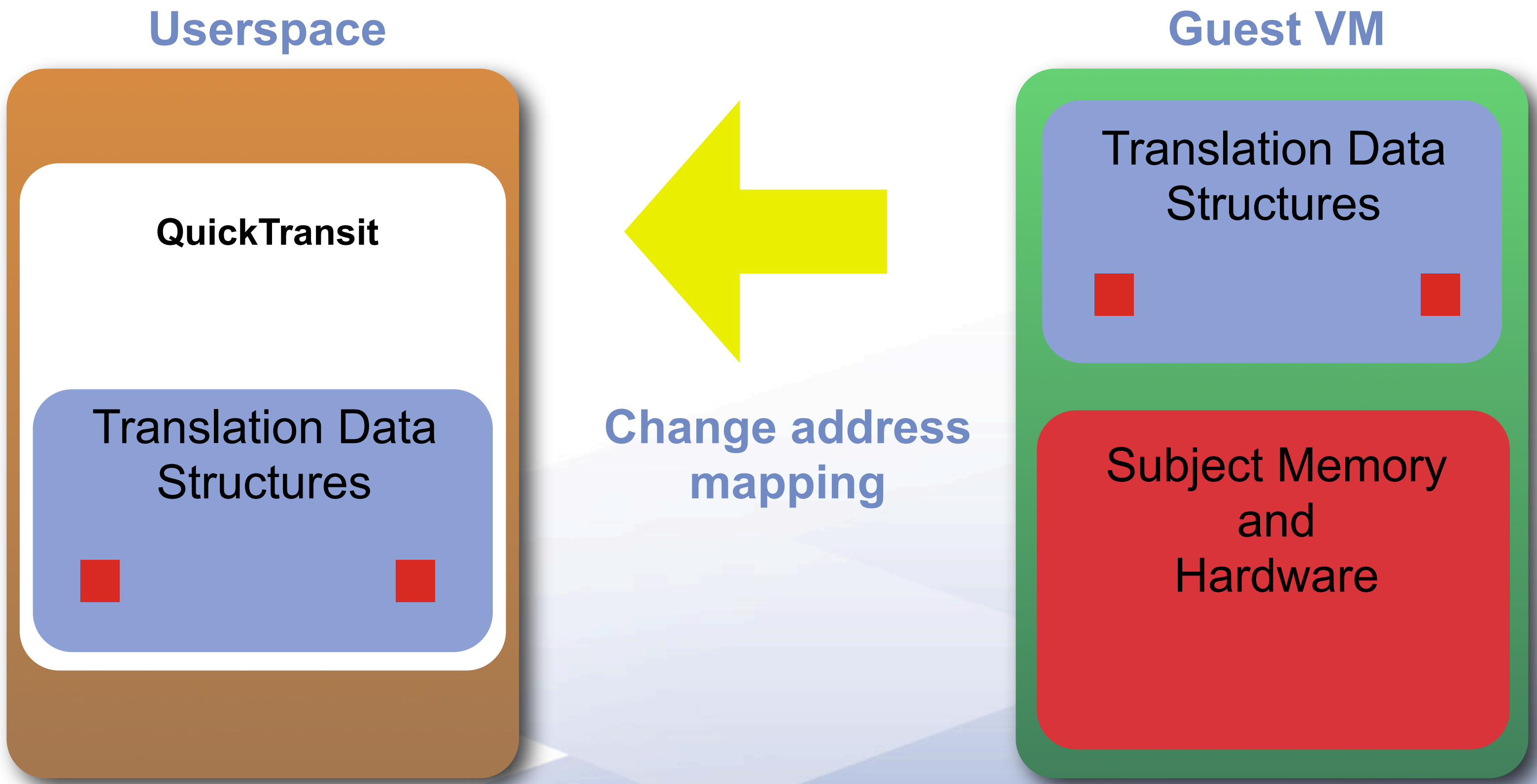

Userspace
QuickTransit
Translation Data Structures (Interrupt)
Guest VM
Translation Data Structures (Interrupt)
Subject Memory and Hardware

How does it work?


Userspace
QuickTransit
Translation Data Structures
Guest VM
Translation Data Structures
Subject Memory and Hardware

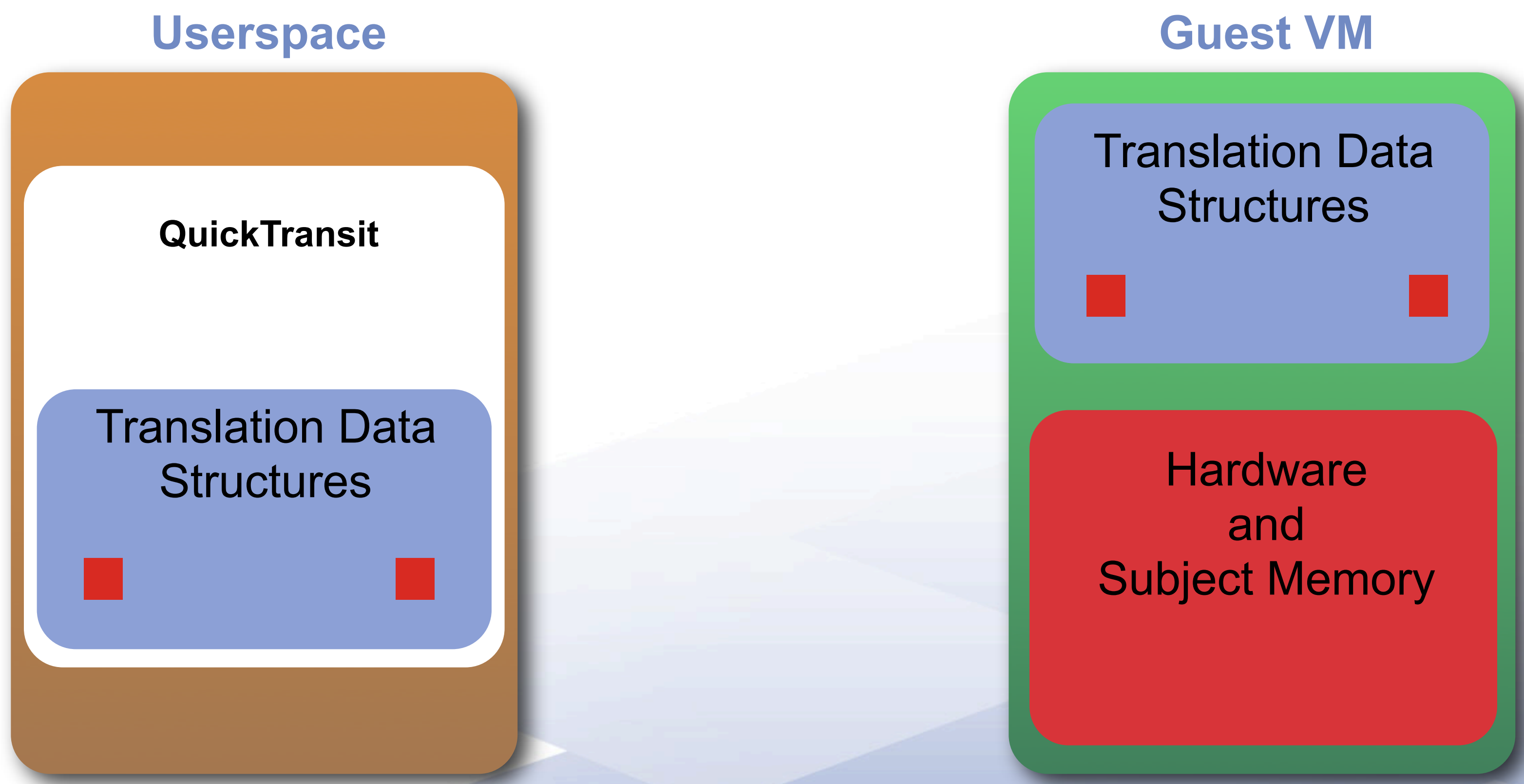
TRANSITIVE™

How does it work?


Userspace
QuickTransit
Translation Data Structures
Change address mapping
Guest VM
Translation Data Structures
Subject Memory and Hardware

TRANSITIVE™

How does it work?


Userspace
QuickTransit
Translation Data Structures
Guest VM
Translation Data Structures
Hardware and Subject Memory

How does it work?

- Minimal translator runtime mapped inside the VM
- Simply jumps from one block of translation to the next
- Exits back to translator when we need new translations
- Exits for hardware accesses as normal
- Also exits back to translator when we need to update page table mappings

TRANSITIVE™

Page table management

- Page tables managed by the translator
- Translator constructs tables that map in translations
- Also constructs tables to reflect the mappings of subject MMU
- Translator has to map semantics of subject MMU on to X86 page tables

TRANSITIVE™

Shadow guest page tables

- Currently 2 levels of shadowing
- QuickTransit shadows subject MMU
- KVM then shadows QuickTransit page tables
- Possibly inefficient, potential for improved performance

Paravirtualizing the guest OS

- Subject can be paravirtualized like any other guest
- QuickTransit captures subject hypervisor traps and maps them through
- We have paravirtualized block devices for our guests to provide disk images

TRANSITIVE™

Changes to KVM?

- So far we have only had to make 1 change to KVM
- Added the ability for the userspace component to invalidate shadow page table entries
- KVM has met all our needs very well

TRANSITIVE™

How do we see this being used?

- Virtualization in the data centre
  - Run any VM on any hardware you have capacity on
  - Live migrate across architectures
- Desktop virtualization
  - Developers can test their software on many platforms
  - VMs cost far less than real machines for infrequently used software
  - Demos can be taken on the road on standard laptops

TRANSITIVE™

About the demo

- AMD Alchemy Pb1500 MIPS development board
- MIPS Linux running SDL Doom as the init process
- Runs unmodified on the real hardware
- QuickTransit maps MIPS TLB on to X86 page tables
- QuickTransit provides emulation of:
    - AU1500 Serial device
    - AU1500 Network device
    - Epson graphics chip
- Translates everything, including the boot monitor

TRANSITIVE™

Technology Demonstration

AMD Alchemy Pb1500 MIPS Linux

TRANSITIVE™

Questions?

TRANSITIVE™