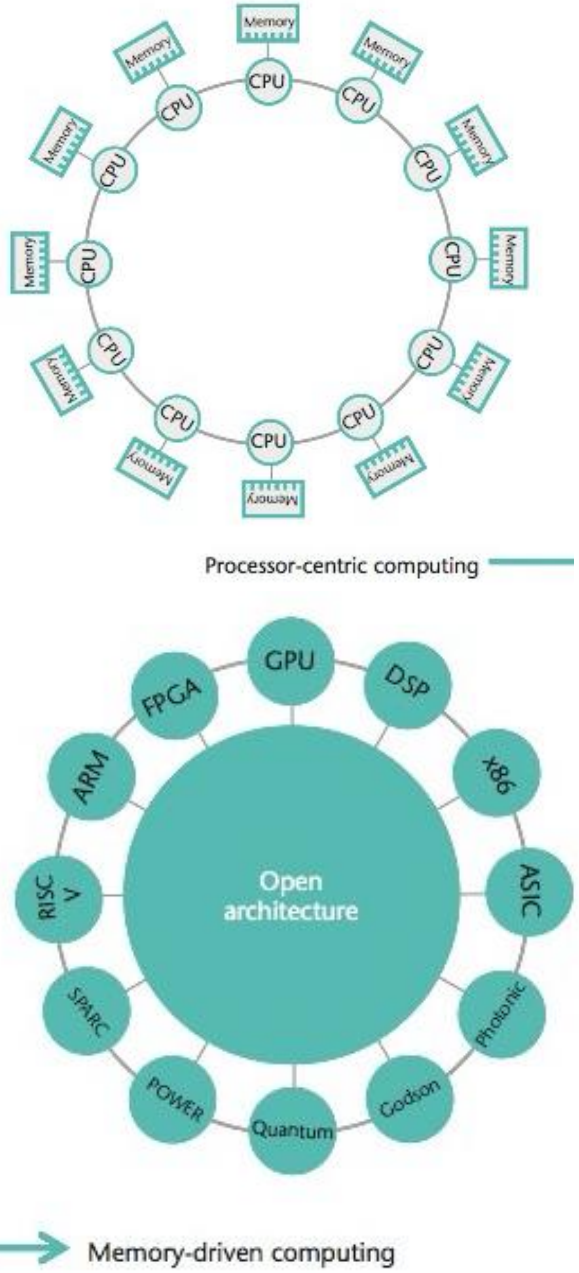
SECURE FOUNDATIONAL ZETA BYTE HPC SYSTEMS FOR 2025 AND BEYOND SV/128 - RISC-V

Steven J. Wallach (steve.wallach@gmail.com)

Presentation Outline

- Background Material (Part 1)
  - Previous efforts/research on protection
- Full Proposal (Part 2)
  - 128 bit logical address
  - 64 bit Unique Object ID
- First implementation (Part 3 & 4)
  - RISC-V SV128 ([21] Github)
    - 32 bit Object ID
  - Programmer Visible State
  - Hardware μ-State [19]
    - Multiple ISA's – heterogeneous compute
  - Contemporary security issues
    - Speculation, etc. issues – not covered



[6] R. Stanley Williams, "The End of Moore's Law", Computing in Science & Engineering, IEEE CS and AIP, March/April 2017

OBJECTIVES

- Why a 128 bit address space?
  - Security
  - Cluster wide shared virtual address
    - Heterogeneous Nodes
  - Time to do something different not just keep adding more bits
  - Begin the decade of ZETA scale computing on a scalable technology
- Software/OS oriented
  - Upward Compatible with RV32 and RV64
- Otherwise we will continue to implement and support the sins of our parents/grandparents.
- ***We can now begin to design & build SECURE PROGRAMMABLE ZETABYTE (2**70) distributed memory systems***

THE BEST BENCHMARK IS THE ONE YOUR COMPETITION CAN NOT RUN. USER CYCLES ARE IMPORTANT AS CPU CYCLE

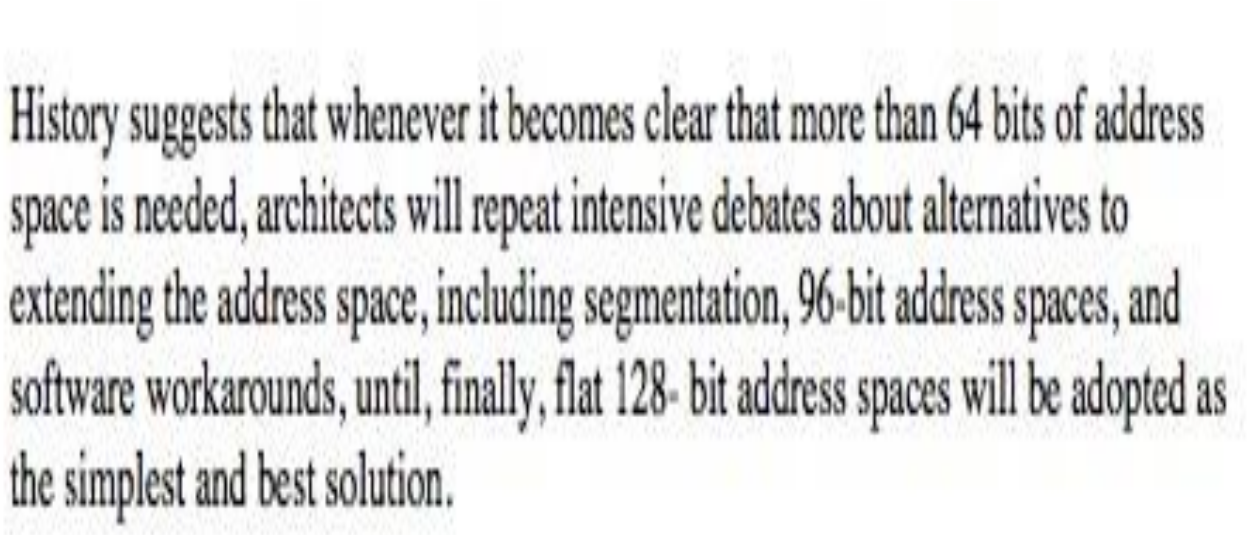
Background

- Since the late 70's, **mainstream** processors have increased the size of the virtual space by simply adding more bit

- Other's (pioneer's) did not simply add more bits
  - IBM – FS (Ref: 13) – 1976
    - Tagged 16 byte pointers (Capabilities)
    - System/38 is the diminutive of FS
  - Data General - FHP (Ref: 3, 4,17) - 1980
    - Ref: http://people.cs.clemson.edu/~mark/fhp.html
      - true object orientation with one-level addressing across a network (128 bit pointers)
  - Intel 432 iMAX OS – (Ref: [16]) – 1980
    - 24 bit passive address
    - 80 bit UID (16 bit checksum)

Going Forward

- Time to define a 128 bit space withc the need for 128 address arithmetic
  - What is “i” in A[i]?
  - A Virtual Address greater than 64 bits
- Time to correct and incorporate appropriate security and access mechanisms
  - Network Wide
  - Access to the web is assumed and required
    - Private Cloud



RISC V ISA SPEC (page 81 – chapter 17)

Security & Address Facts

- Computer Virtual Address's (VA) span to local disk only
  - Disk Access is now Global (In practice)
  - Remember a VA references NON_VOLATILE/DISK explicitly **NOT** main memory (CS101)
- Network Addressing (IPv4 & IPv6 span the entire network)
  - IPv6 created a 128 network address space. Unique names
- MAC and URL's addresses are unique
- EMAIL addresses are unique
- Phone numbers are global; country code, city code, local code
- Two different (web and local) address structures
  - Two different protection and addressing systems
  - Two different authentication systems
- Software needed to bridge these two domains (too much software)
- What if one unified name structure could be developed?

Foundational Basis

- The original motivation for putting protection mechanisms into computer systems was to keep one user's (program) malice or error from harming other users (program). Harm can be inflicted in several ways:
  - a) By destroying or modifying another user's (program) data.
  - b) By reading or copying another user's (program) data without permission.
  - c) By degrading the service another user (program) gets, for example, using up all the disk space or getting more than a fair share of the processing time

[1] Lampson, Protection. *Proc. 5th Princeton Conf. on Information Sciences and Systems*, Princeton, 1971

Previous Efforts

- Another solution is to address each segment with **a unique integer** which is assigned at the time the segment is created, never changed, and not reused even after the **segment has disappeared from the system. Call this the** unique integer solution. ([3,4,5] & [13]Radin's H – Handle)

[3] US Patent**,** 4,525,780, "DATA PROCESSING SYSTEM HAVING A MEMORY USING OBJECT- BASED INFORMATION AND A PROTECTION SCHEME …" , 1985

[4] US Patent, 4,821,184," UNIVERSAL ADDRESSING SYSTEM FOR A DIGITAL DATA PROCESSING SYSTEM", 1989

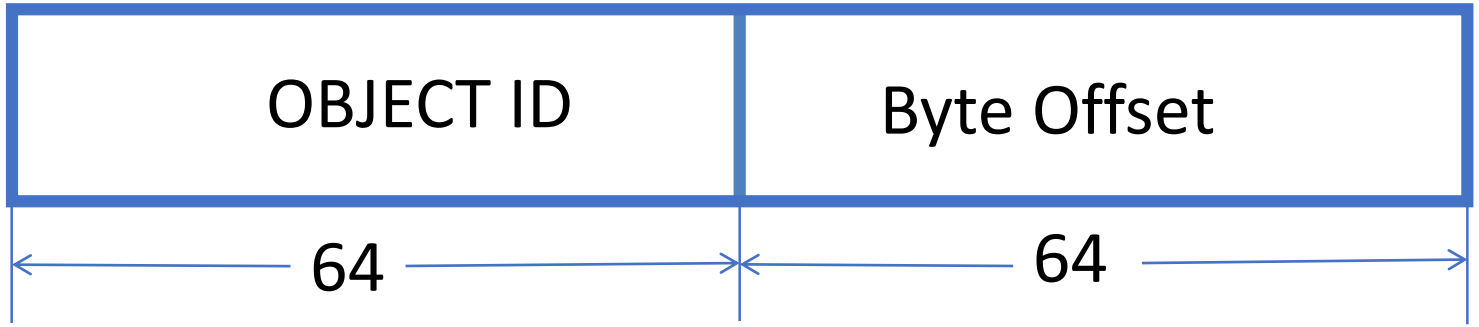
[5] Fabry, " Capability-Based Addressing", CACM, July 1974

[13] Radin, Schneider, "An Architecture for an Extended Machine With Protected Addressing", Radin, Schneider, "An Architecture for an Extended Machine With Protected Addressing", TR 00.2757, IBM May 21, 1976

PART 2 -Full Proposal

- 128 bit logical address space
- Protection structures
- For now biting off more than is feasible
- Leads to a ”RATIONAL” proposal extensible to the full proposal
  - 96 bit logical address
  - Protection system supports current Linux architecture.

Logical Address Space



- 128 Bits
- Object ID – Unique Identifier
  - a software (or hardware) structure that is considered to be worthy of a distinct name.
- Indexing is 64 bits – A[i]
  - Extended RISC-V isa (additional 64 bit registers)
- ISA independent
  - Like routing IP packets (Vendor Independent)
  - Shared Pointers/Addressing facilitates Heterogenous Processing.
  - Common Global Physical Address
- Persistent across time and space
- Protection and memory management are independent

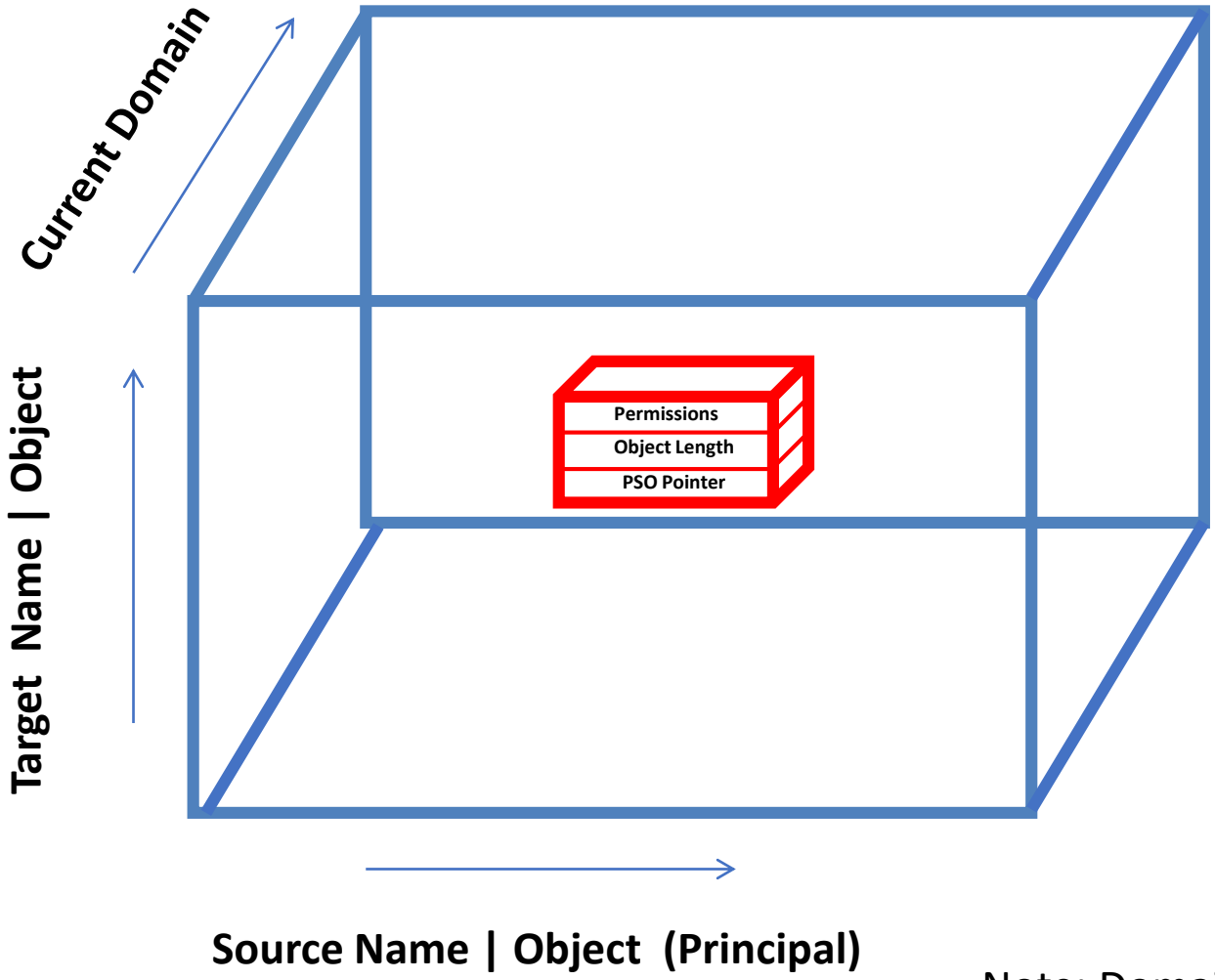
Motivations

- We need better security
- We need computer virtual addressing to reflect the contemporary uses
  - Network Wide
- We do ***NOT*** want 128 bit flat addressing
- We need pointer interoperability between computer systems.
  - Memory Centric - Heterogeneous ISA
- We need a simplified sharing mechanism
- We need authentication, revocation and protection again malware/virus's

Unified Name Server Operation

- The Domain Name System (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network devices worldwide.
- Unified Name Server Operation
  - Manage Creation and Revocation of Objects (UID's)
  - Authenticate and Validate Name Request
  - Does NOT store data or applications
  - Translate UID's (memorized domain names) into IPv6 address's when accessing EXTERNAL to HOST system
    - Just like OS's translate Virtual Address's into disk/ssd address's
    - Are Object's IPv6?

Protection - ACL (matrix) – uses OBJECT names -maintained by Name Server-



Note: Domain and Process
Are NOT unique

Authentication

- The address space is unique over time and space. Any computer supporting this address space is addressable by the name server.
- Accessing an object for the first time requires
  - Permission to access (i.e., download A.OUT or .EXE file, entry within an ACL)
  - Access privileges for the object
    - Local and Network read/write/execute
    - Access only thru a protected sub-object
    - Execution Domain
- Domain of execution
  - Level of user (e.g., gold, platinum, executive platinum)
  - Admin (Level 1, 2, or n)
- In essence we have a global access control list
  - We have that today, but don't realize it
  - It is distributed (e.g., ADOBE maintains it 2D slice of the matrix)
  - Each Vendor has their own access control list

Example using An Application

Steve Wallach – OBJECT

Adobe PDF Reader - OBJECT

- Can Execute in My Domain
  - Can read and write my file system
- Can execute in different domain
  - Determine level of trust
  - Can read my data, but not write
  - Can’t send data back to ADOBE (network permissions)
- **Adobe FLASH - NO ACCESS**

ACL Entry

- PSO – Protected Sub-Object (Sandbox – [11])
  - Virtual Machine
  - Requires software interpretation
    - Address of Sandbox
  - Mediated access
  - Part of Object creation
    - The meta data of the object
  - … the concept of confining a helper application to a restricted environment, within which it has free reign

Protected Domains

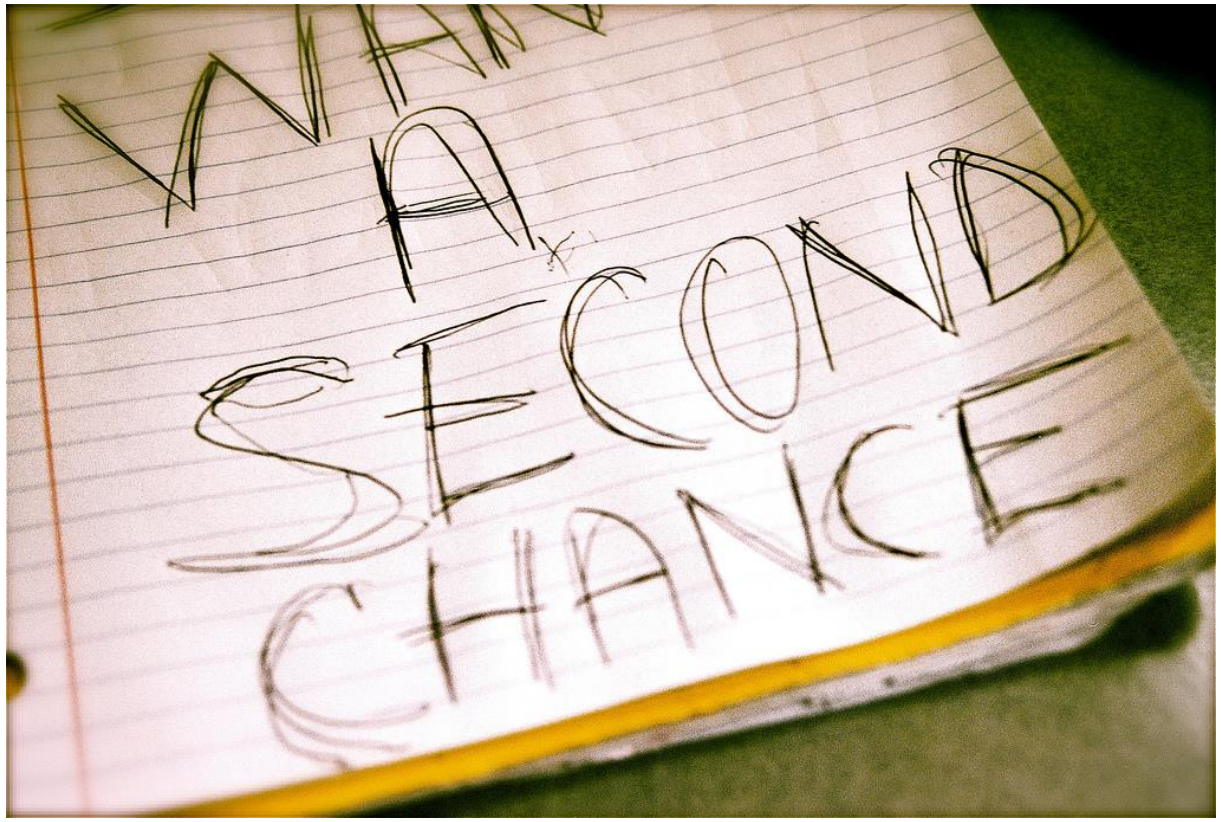
- Definition
  - An object in a domain with mediated access. An application MUST always use mediated access's to reference data external to the object and/or make system calls.
- Sand boxing an app (SHADOW STACK)
  - Handling Malware/Virus
- Sand boxing email
  - Restrict and Terminate (with malice – force quit)
- **The Principle of Least Privilege ([2] & [11])**

Revocation

- Every client has a “KILL SWITCH”
- The central name server is accessed and each object has a kill capability (only initiated by the owner)
- What if the unified name server is NEVER accessed again??
  - Watch Dog Timer?
- Compare to a capability based system ([5] Fabry & [12] Watson, et. al, CHERI)

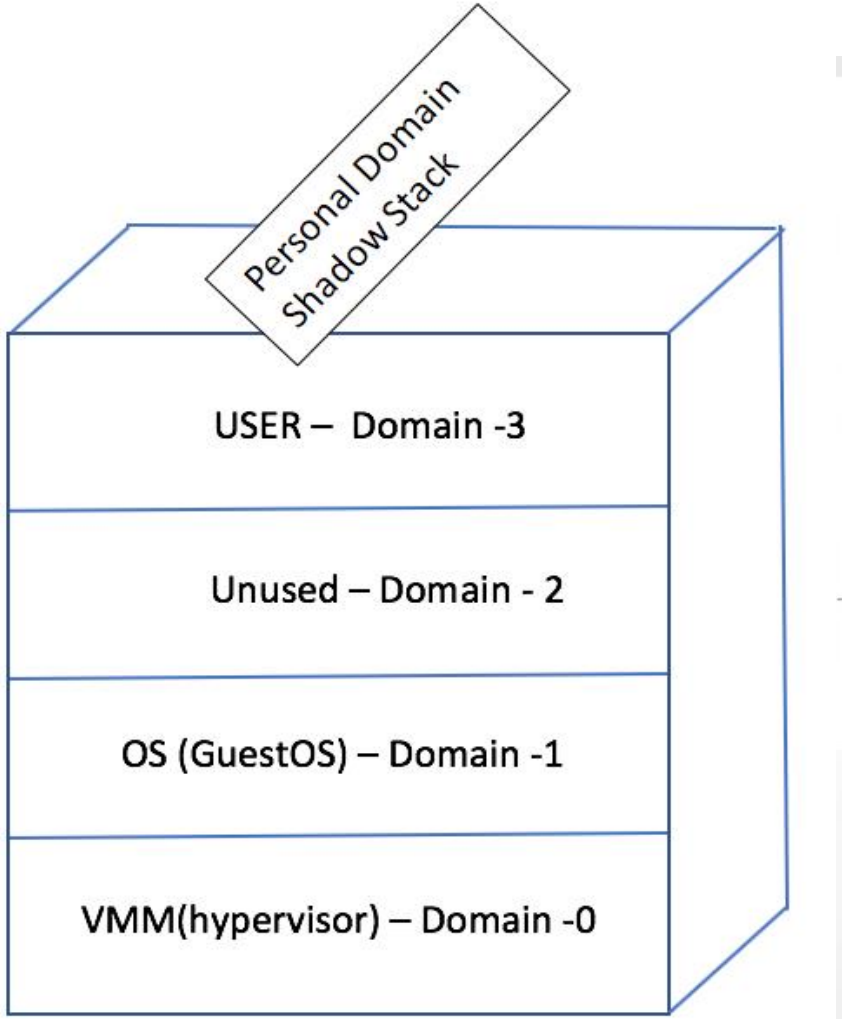
PART 3 -NOW WHAT?

- We have a foundational basis
- We have a need
- We now need to be pragmatic
- We now need to be able to implement the hardware and software.
- We gratefully thank our parents and grandparents



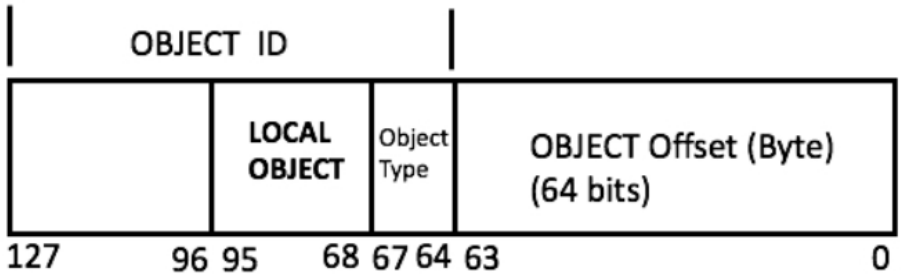
Current Proposal for RISC-V SV128 [21]

- 4 Domains (non hierarchical)
  - Direct support for VMM
- Personal Domain for sandboxing
  - On a domain basis
- 96 bit logical address space
  - 32 bit object. 64 bit object offset
- Upward Compatibility
  - RV32 and RV64
- Logical to physical address translation
  - Uses RV64 (and its options) translation per object
  - Hash Table – entire 64 bit object/offset used



LOGICAL ADDRESS SPACE (SV128)

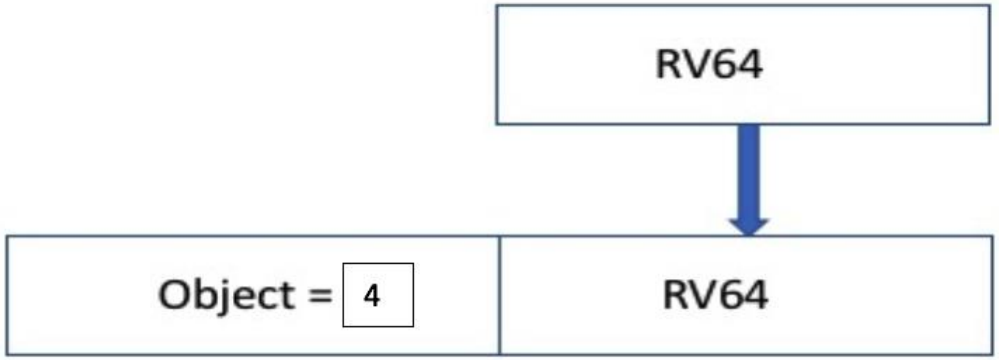
- 128 bit logical address
- 96 bits now defined.
- Object Types
  - PGAS
  - Kernel (protected)
  - Expansion to 128 bits
  - Local Object indexed into table to determine other object types (e.g., encrypted)
    - Late Binding

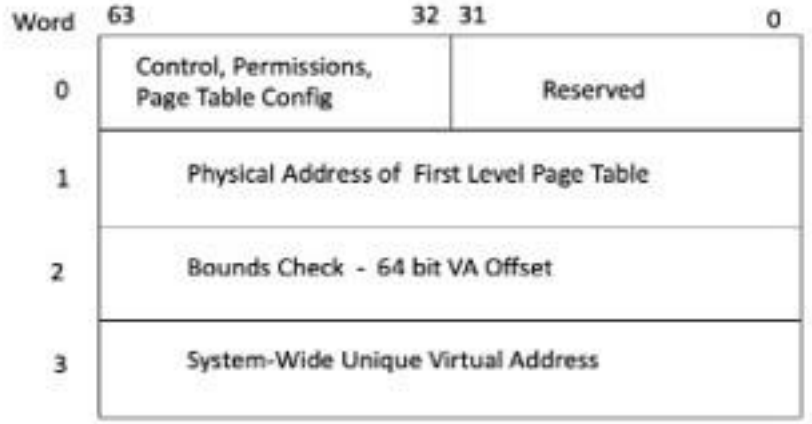


- Object Type= 0,1,2,3 - Kernel Object.
- Object Type = 4 – Byte offset is RV64
- Object Type= 5 - Byte offset is RV32
- Object Type= 6 - Interpret Object Bits 68 thru 95 (28 bits) as a PGAS Object
- Object Type = 7 - Interpret Object Bits 68 thru 95 (28 bits) as a local OBJECT ID
- Object Type = 8 thru 14 reserved
- Object Type- 15 – Interpret Bits 68 thru 127 (60 bits) as an Object ID.

Mapping RV64 into SV128 Protection & Bounds Check

- RV64 mapped to SV128





Object Table Entry

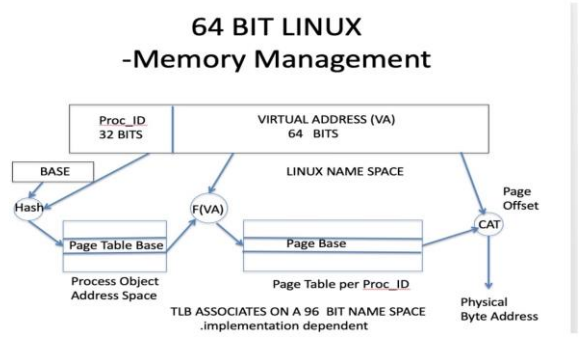


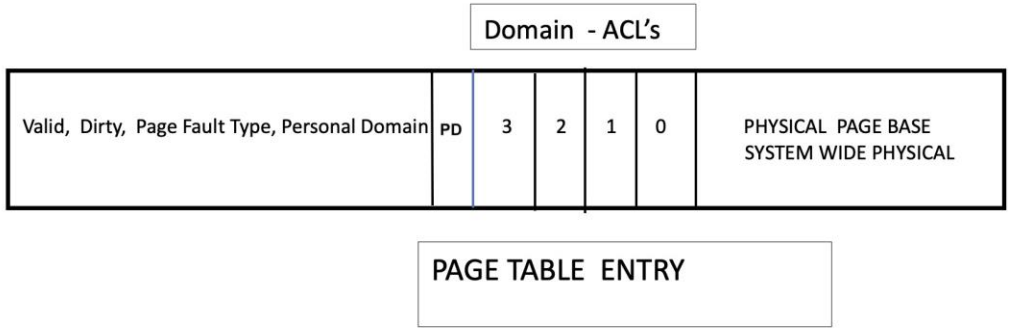
Figure 12: CURRENT 64 Bit Linux Kernel structures

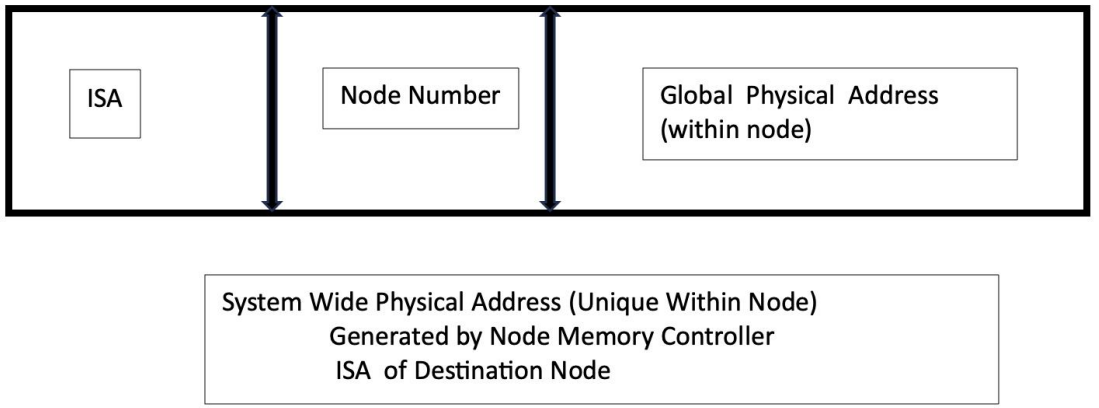
Entry specifies:

.bounds check, levels of page table, page_size, etc.
.replaces one CSR register with indexed table entry
.each domain/process has its own table (principal)
.first level of data protection (is reference permitted?)
.Object Type
.Unique TLB Naming - no flush between context switching
.ISA (heterogeneous compute nodes)

Data Reference Permission Bits

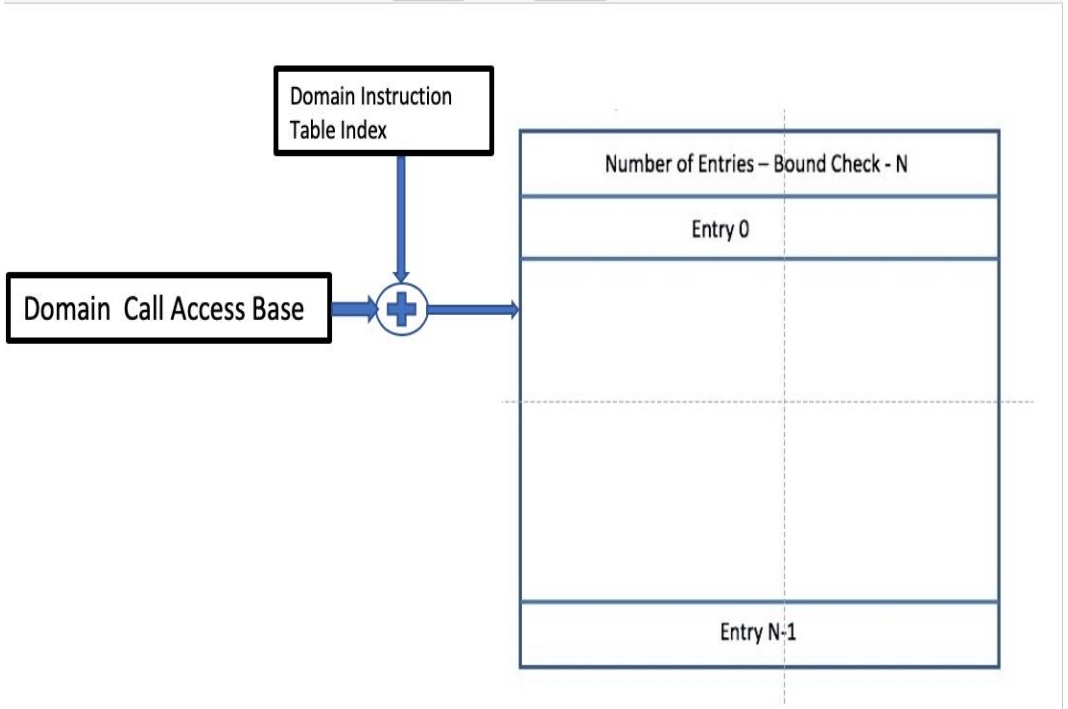
- Read, Write, Execute per domain
  - Non-hierarchical
- Shadow stack permission bits (personal domain)
- Same format for permission bits and controls as word 0 of Object Definition Table
- Page Fault Type
  - SSD, etc. (replace or cache?)
    - HPC – 1 byte per peak flop



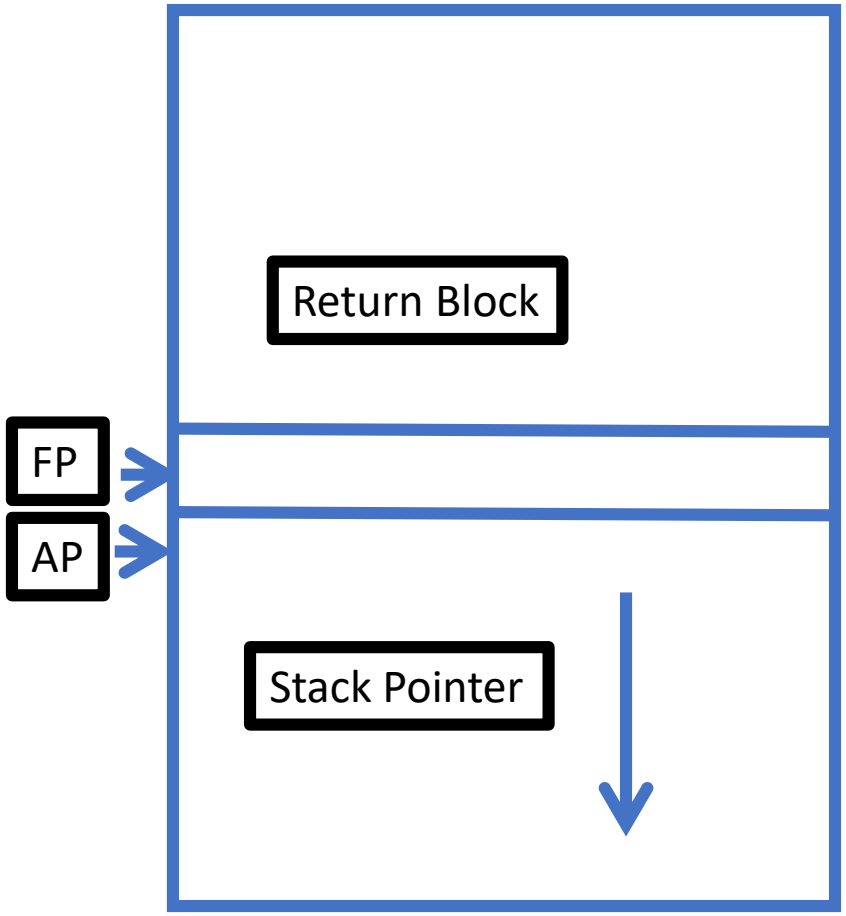


Domain Switching

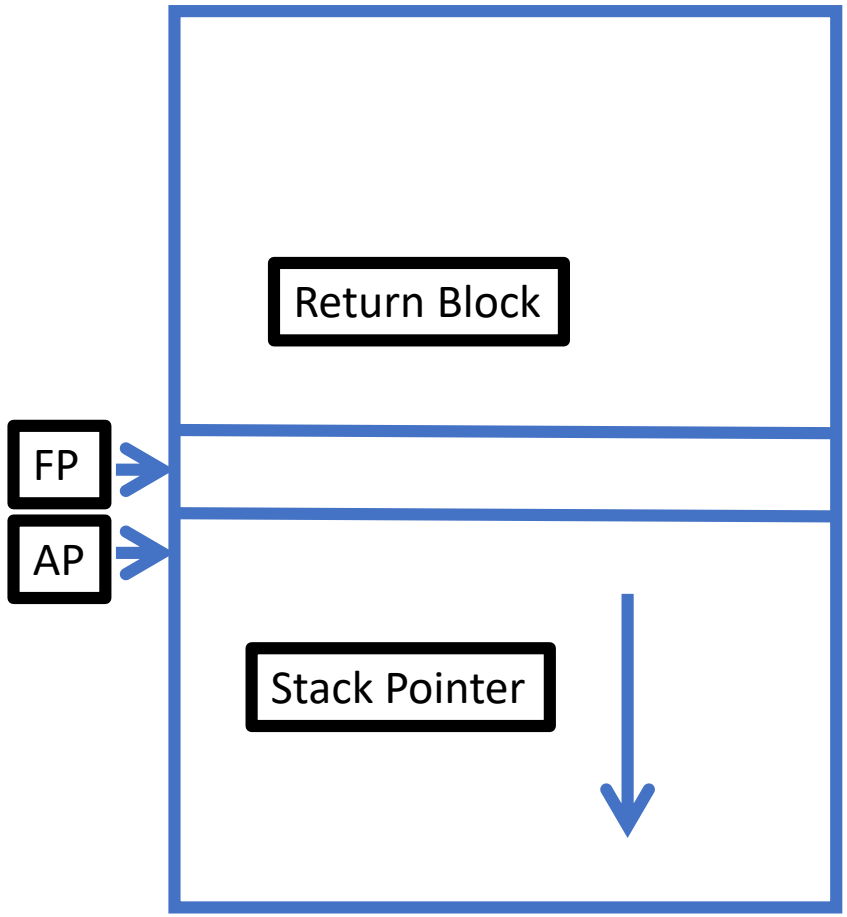
- One Table per domain
- Domain Call Instruction specifies Domain/Index. Table does:
  - Bounds Check
  - ENTRY: Permission bits for each caller domain per entry (e.g., D2 call D3 for entry X).
  - ENTRY: Valid (Sparse Table)
- Shadow stack makes calls using table in current domain.
  - Mediates calls (PD)
- Stack saving and switching
- Validate Instructions to check for Melt Down Attacks



Shadow Stack - Architecture



Shadow Stack (Hardware Maintained)
Different Domain, Same virtual address
Return via THIS STACK



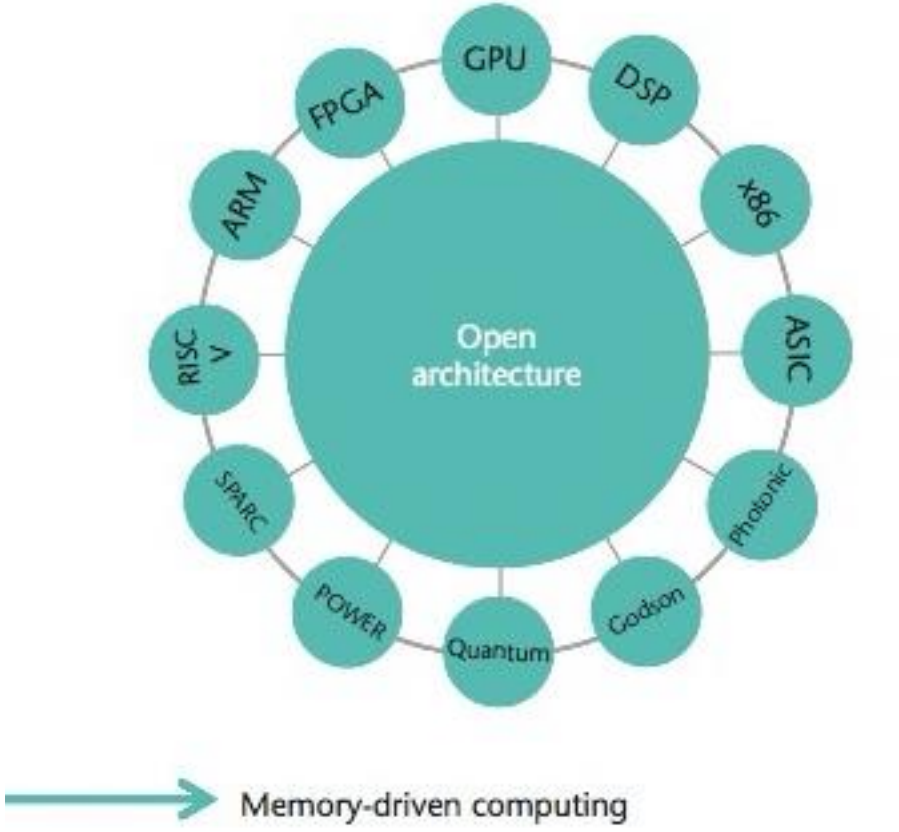
User Visible Stack
Can not write/read Shadow Stack

Compatibility with RV64

- Execute RV64 a.out.
- Map 64 bit virtual into 128 virtual
- System calls to intermediate server
  - 64 bit pointers to 128 and then kernel call
- Use RV64 utilities (e.g. Editor) to create RV128 files

Benefits/Characteristics

- Logical Address Space is 96 bits
- Bounds Check on address
  - Google Analysis of Android Kernel [18]
    - 44% kernel bugs (missing/incorrect bounds check)
- System Wide TLB
  - No flush between Process or Hypervisor multiplexing
- Direct Support for PGAS like addressing
- Common Virtual Address of different ISA's
  - Or, just common physical



Summary

- Build an emulation/prototype: FPGA and/or software

Acknowledgement

- Dave Clark (MIT) – reviewed presentation, made many suggestions
- Daniel Wallach (RICE) – challenges, suggestions
- Terrel Magee – Micron
- John Leidel – tactcomplabs
- Steve Poole - LANL

References

- [1] Lampson, Protection. *Proc. 5th Princeton Conf. on Information Sciences and Systems*, Princeton, 1971
- [2] Schroder & Saltzer, "The protection of information in computer systems", PROCEEDINGS OF THE IEEE, VOL. 63**,** NO. 9, SEPTEMBER 1975
- [3] US Paten**t,** 4,525,780 ,"DATA PROCESSING SYSTEM HAVING A MEMORY USING OBJECT-BASED INFORMATION AND A PROTECTION SCHEME FOR DETERMINING ACCESS RIGHTS TO SUCH INFORMATION"
- [4] US Patent, 4,821,184," UNIVERSAL ADDRESSING SYSTEM FOR A DIGITAL DATA PROCESSING SYSTEM"
- [5] Fabry, " Capability-Based Addressing", Communications of the ACM, July 1974, Volume 17, Number 7
- [6] R. Stanley Williams, "The End of Moore's Law", Computing in Science & Engineering, IEEE CS and AIP, March/April 2017
- [7] Mark Hill, https://www.sigarch.org/a-primer-on-the-meltdown-spectre-hardware-security-design-flaws-and-their-important-implications/
- [8] https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html
- [9] http://en.wikipedia.org/wiki/Heap_feng_shui
- [10] **https://gruss.cc/files/kaiser.pdf,** KASLR is Dead: Long Live KASLR, D. Gruss, et. al.
- [11] Ian Goldberg, David Wagner, Randi Thomas, and Eric Brewer (1996). "A Secure Environment for Untrusted Helper Applications (Confining the Wily Hacker)". *Proceedings of the Sixth USENIX UNIX Security Symposium*.

References

- [12] Robert N.M. Watson, Peter G. Neumann Jonathan Woodruff, Jonathan Anderson, Ross Anderson, Nirav Dave, Ben Laurie, Simon W. Moore, Steven J. Murdoch, Philip Paeps, Michael Roe, and Hassen Saidi. CHERI: a research platform deconflating hardware virtualization and protection. Workshop paper, Runtime Environments, Systems, Layering and Virtualized Environments (RESoLVE 2012), March, 2012.
- [13] Radin, Schneider, "An Architecture for an Extended Machine With Protected Addressing", TR 00.2757, IBM May 21, 1976.
- [14] J. Chase, H. Levy, et. al, "**Sharing and protection in a single address space operating system**" " Journal ACM Transactions on Computer Systems (TOCS) - Special issue on computer architecture, Volume 12 Issue 4, Nov. 1994, Pages 271-307
- [15] Zhang, Estrin, et. al, "Named Data Networking (NDN) Project, NDN -0001, October, 31, 2010**.**
- [16] F. Pollack, et. al., " **The iMAX-432 object filing system"** Proceeding SOSP '81 Proceedings of the eighth ACM symposium on Operating systems principles Pages 137-147
- [17] **http://people.cs.clemson.edu/~mark/fhp.html**
- [18] What's New in Android Security", Google I/O 2017. https://www.youtube.com/watch?v=C9_ytg6MUP0
- [19] Mcilroy, et. al, " Spectre is here to Stay. An analysis of side-channel and speculative execution" https://arxiv.org/pdf/1902.05178.pdf
- [20] http://compcert.inria.fr/
- [21] https://github.com/tactcomplabs/sv128-archspec