Software Systems at the Edge

Wei Gao

The Core of Edge Computing

- Core Problems:
  - Workload offloading
  - Efficient remote execution at the edge server
  - User mobility between edge servers

- Key Approaches:
  - VM provisioning and synthesis
  - VM migration

VM Synthesis and Provisioning

- Using Docker

- **What it is?**
  - Containerization technology
  - Application level virtualization
  - LXD, LXC, OS virtualization
- **Benefits**
  - Small footprint
  - Lightweight
  - Portability
- **Original Objective**
  - PaaS, Application deployment, Sandboxing

Fundamental requirements of Docker

1. **Service Deployment & Termination -** Basic needs
   - Compute power multiple form factor
   - Geo-distributed
   - Fast & easy to manage – low footprint.

2. **Resource & service Management**
   - Resource discovery service
   - Uncontrolled environment.
   - Heterogeneous
   - Resource come & go

3. **Fault Tolerance –** for infra & services
   - FT for infra
   - FT facilities for service

Requirement 1: Service Deployment

Requirement	Evaluation	Gap
compute power	- Support multiple form factor. - Low compute footprint. Occupies 10 – 15% of host resources. - Coexist in user devices - minimal.	Android, some initiative.
Geo-distributed	- Able to manage resources. - Clustering available e.g. Swarm.	- Hierarchical management of Edge sites
Easy to Manage e.g. - Fast Deployment - easy to install, configure, deploy.	<u>Service Deployment</u> - Service registry a.k.a. (marketplace) - Small image & no boot time - Fast deployment. <u>Resource management</u> - Docker available to most Linux distros - Highly Portable - Small image – store deltas. - Clean teardown – does not affect host. - Multi-tenant environment – ok.	<u>Service Deployment</u> - Service discovery across edges. <u>Resource management</u> - Resource discovery across edges. - Windows based Docker is not supported.

Requirement 2: Resource Management

Requirement	Evaluation	Gap
Resource discovery service	Swarm support multiple resource discovery services.	No discovery across multiple edge sites
Uncontrolled environment.		- Ownership, trust - How to enable cross-organizational resource sharing.
Resource come & go	- Swarm support discovery - Detects failing or leaving devices	- service and data should be balance. - launch new service even with minimal expected performance. - New Resource available – to increase user's QoE gradually by scaling existing service.

Requirement 3: Fault Tolerance

Requirement	Evaluation	Gap
Fault tolerance on infra	- Support multiple zones – failovers, DR .	
FT facilities for service. Support 99.999% minimal	- Loosely coupled - single-app per-container basis. - CRIU – check-pointing on containers For application that does not support state-less transactions. - Offline migration – to import export container.	- redeploy affected service (s) at different host or location e.g. another edge.

VM Migration

- Two basic strategies
- Local names: move the state physically to the new machine
  - Local memory, CPU registers, local disk (if used – typically not in data centers)
  - Not really possible for some physical devices, e.g. tape drive
- Global names: can just use the same name in the new location
  - Network attached storage provides global names for persistent state
  - Network address translation or layer 2 names allows movement of IP addresses

Process-based Migration

- Typically move the process and leave some support for it back on the original machine
  - E.g., old host handles local disk access, forwards network traffic
  - these are “residual dependencies” – old host must remain up and in use
- Hard to move exactly the right data for a process – which bits of the OS must move?
  - E.g., hard to move TCP state of an active connection for a process

VMM Migration

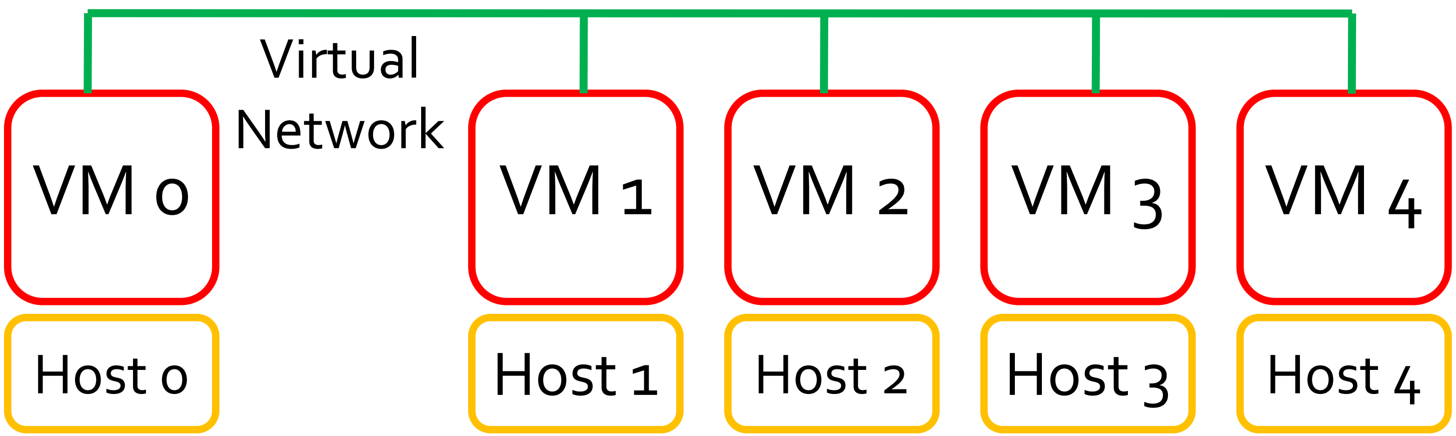
- Move the whole OS as a unit – don’t need to understand the OS or its state
- Can move apps for which you have no source code (and are not trusted by the owner)
- Can avoid residual dependencies in data center thanks to global names
- Non-live VMM migration is also useful:
  - Migrate your work environment home and back: put the suspended VMM on a USB key or send it over the network
  - Collective project, “Internet suspend and resume”

Goals and Challenges

- Minimize downtime (maximize availability)
- Keep the total migration time manageable
- Avoid disrupting active services by limiting impact of migration on both migratee and local network

A Practical Solution: SnowFlock

Stateful swift cloning of VMs



- State inherited up to the point of cloning
- Local modifications are not shared
- Clones make up an impromptu cluster

Fork has well understood semantics

partition data
fork N workers
if child:
  work on i^{th} slice of data

Parallel Computation

if more load:
  fork extra workers
if load is low:
  dealloc excess workers

Load-balancing Server

trusted code
fork
if child:
  untrusted code

Sandboxing

if cycles available:
  fork worker
  if child:
    do fraction of long computation

Opportunistic Computation

Why SnowFlock is Fast

- Start only with the basics
- Send only what you **really need**
- Leverage IP Multicast
  - Network hardware parallelism
  - Shared prefetching: exploit **locality patterns**
- Heuristics
  - Don’t send if it will be overwritten
  - Malloc: exploit **clones generating new state**

The Industrial State of the Art

- ETSI MEC:
  - Work on architecture progressing, target Feb 2016, then focus on APIs to 3rd parties by summer 2016. Lack of engagement by a number of companies, incl. Ericsson.
- ETSI MEC PoCs: 2 proposals available so far.
- Other publicly announced trials/PoCs:
  - DTAG with Nokia, Continental, Fraunhofer for V2X, motorway A9 in Germany, see PR 11 Nov 2015
  - EE with Nokia on video orchestration, Wembley stadium London, see MEC Congress 29-30 Sept 2015
- Big Cloud players:
  - IBM: quiet
  - HP: quiet
  - Google: most recent move for latency reduction is introducing bridges to CDNs (see PR) http://www.lightreading.com/data-center/cloud-strategies/google-builds-cloud-bridges-to-cdns/d/d-id/718092
  - Microsoft:
    - Edge cloud a non-feature at Future Decoded, 11 Nov 2015.
    - However, MS Azure has joined Open Fog on 19 Nov 2015.
    - Cloudlets still promoted by V. Bahl, Sept 2015. http://www.networkworld.com/article/2979570/cloud-computing/microsoft-researcher-why-micro-datacenters-really-matter-to-mobiles-future.html

Open Edge Computing

The good news

- Cloudlets are a well-known concept. The term has been adopted by various companies (incl. Nokia, Akamai)
- Lots of publications exists about Cloudlets as enabler.
- Many parties know about the cloudlet concept (though don’t take action in OEC)
- Cloudlets predate ETSI MEC and Open Fog for a number of years, has a long research history
- Working code for Cloudlet is available.
- Level of interest in Cloudles appears to be rising: Queries e.g. from
  Crowncastle, NTT Labs, Elisa

Open Edge Computing

Challenges

- Level of interest:
  - OEC is a (too) small group today. Initial interest from various players hasn't translated into action or funding: e.g. Red Hat, Nokia
- Overlap with others:
  - In particular with ETSI MEC IEG: regarding goal to reach out to verticals and IT industry
- Level of usability/maturity of Cloudlet software platform:
  - Difficult to install from scratch for non-experts, in particular difficult to install both OpenStack + Cloudlets on any HW/SW platform → could we create an installer for both?
  - Possibly current SW too tightly linked to a particular HW configuration
  - Cloudlet per se is not the E2E solution yet: What about orchestration, app deployment, edge discovery etc.?
    Recommended: Get System Integrators/Consulting companies on board to do that job.

StakeHolder Analysis

Active in OEC initiative
- Intel
- Huawei
- Vodafone
- CMU
- Vedams

Key players in Verticals
- Consulting companies
- Members of vertical industries

System integrators
- CGI
- Accenture
- …

Key players in Openstack
- Red Hat
- HP

Key players in Linux
- Red Hat
- Canonical/Ubuntu

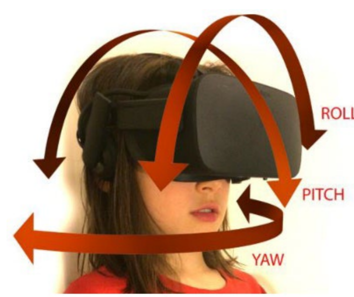
Key players in Cloud
- Microsoft Azure
- IBM Bluemix
- AWS
- Google
- Salesforce.com
- ….

Interested in Cloudlets
- NTT labs
- CrownCastle
- Vantrix
- Daqri

Position unclear regarding OEC
- Nokia
- Ericsson
- ALU
- Juniper
- Brocade

Virtual Reality

- Immersive senses of virtual world
  - Stereo images in 3D simulated environment
  - Equipped with a Head Mounted Display (HMD)

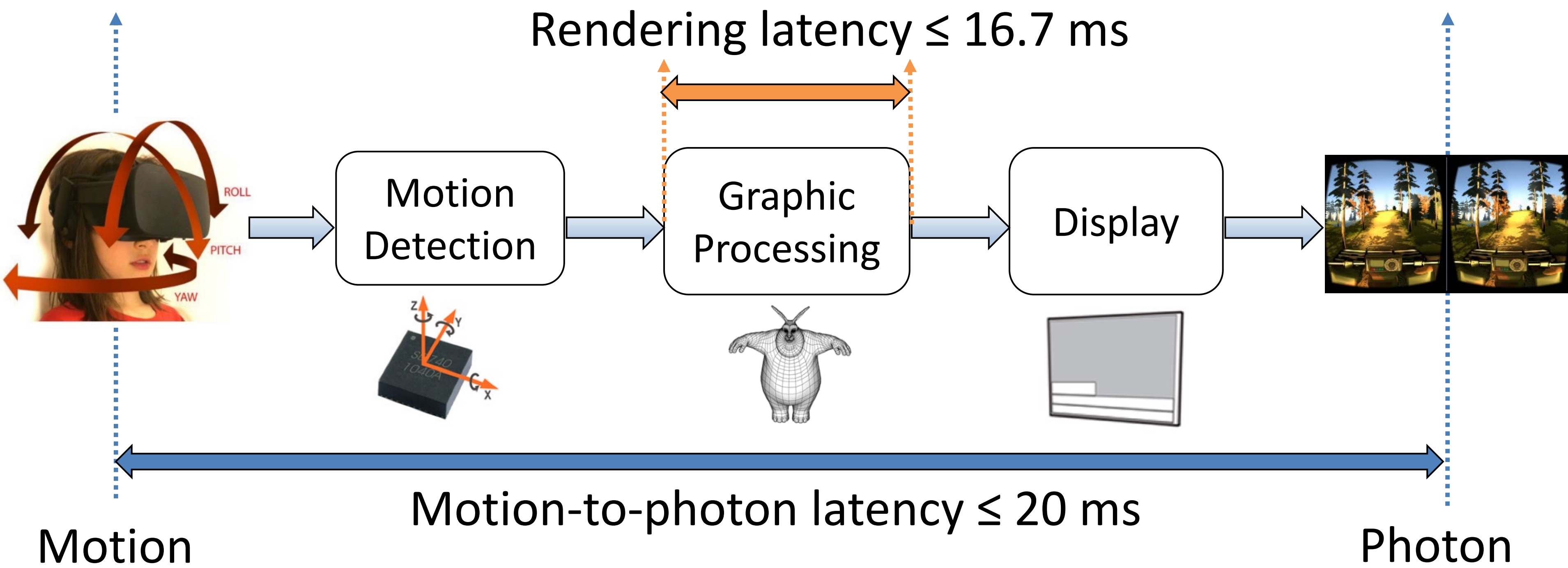


Virtual Reality

- Immersive senses of virtual world
- Applications
  - Gaming
  - Healthcare
  - Education
  - Training

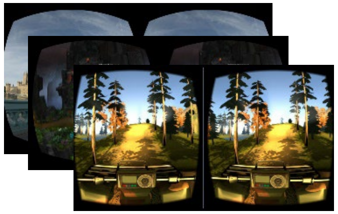
Virtual Reality

- Stringent performance requirements
  - High frame rates (≥ 60 fps, trending 90+ fps)
  - Low motion-to-photon latency (≤ 20 ms)
    - To avoid motion sickness



Virtual Reality

- Tethered virtual reality
  - Powerful desktop PC (e.g., Oculus Ready)
  - Head mounted display
  - HDMI connection



Oculus Ready PC:
- GPU: NVIDIA GTX 1050 Ti
- CPU: Intel i3-6100
- Memory: 8GB+ RAM

Virtual Reality

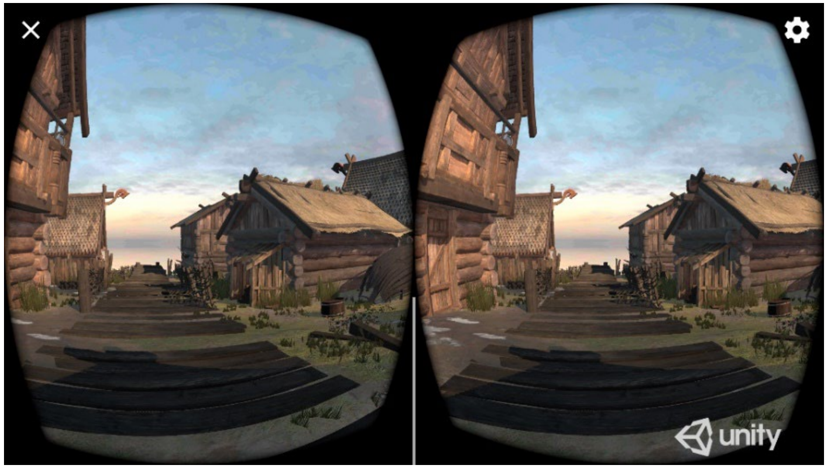
- Tethered virtual reality
  - Powerful desktop PC (e.g., Oculus Ready)
  - Head mounted display
  - HDMI connection

Restricted Movement
&
Tripping Hazards

A Better Option: Mobile Virtual Reality

- Untethered VR via smartphones
- Usable anytime and anywhere
- However…
  - Limited computational capacity
  - Limited battery life

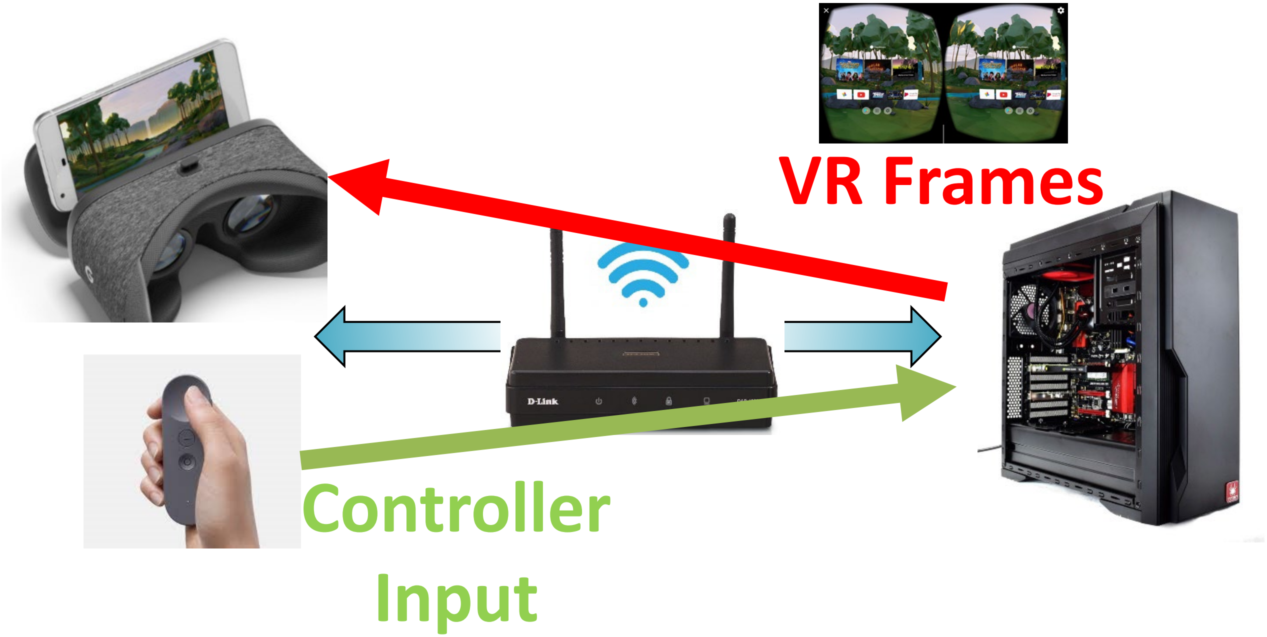
Poor Graphic Details



Low Frame Rates (<13 FPS)

Solution: Remote Frame Rendering

- Render VR frames at the nearby edge cloud
  - Based on controller input
  - Wirelessly transmit VR frames back



Challenges

- Computation overhead

VR:

- **Panoramic** (360^{o})
- **4K** (3840x2160)

Traditional:

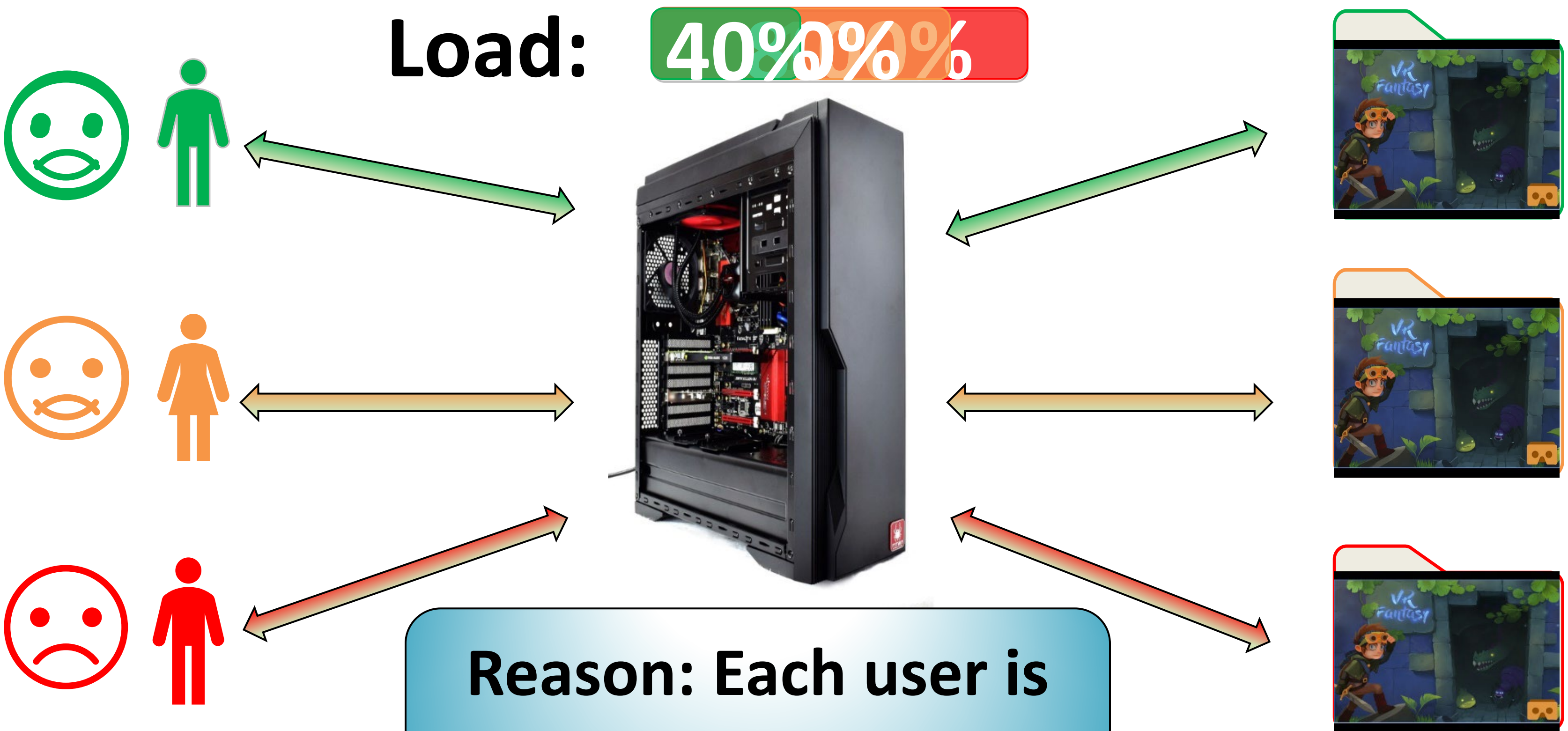
- **Partial** (90^{o})
- **720p** (1280×720)

6X More Computation

- Communication overhead
  - ~33MB raw pixels per frame
  - ~2GB per second

Challenges

- Resource constraints at the edge cloud
  - Poor scalability to serve multiple users



Reason: Each user is served <u>independently</u>

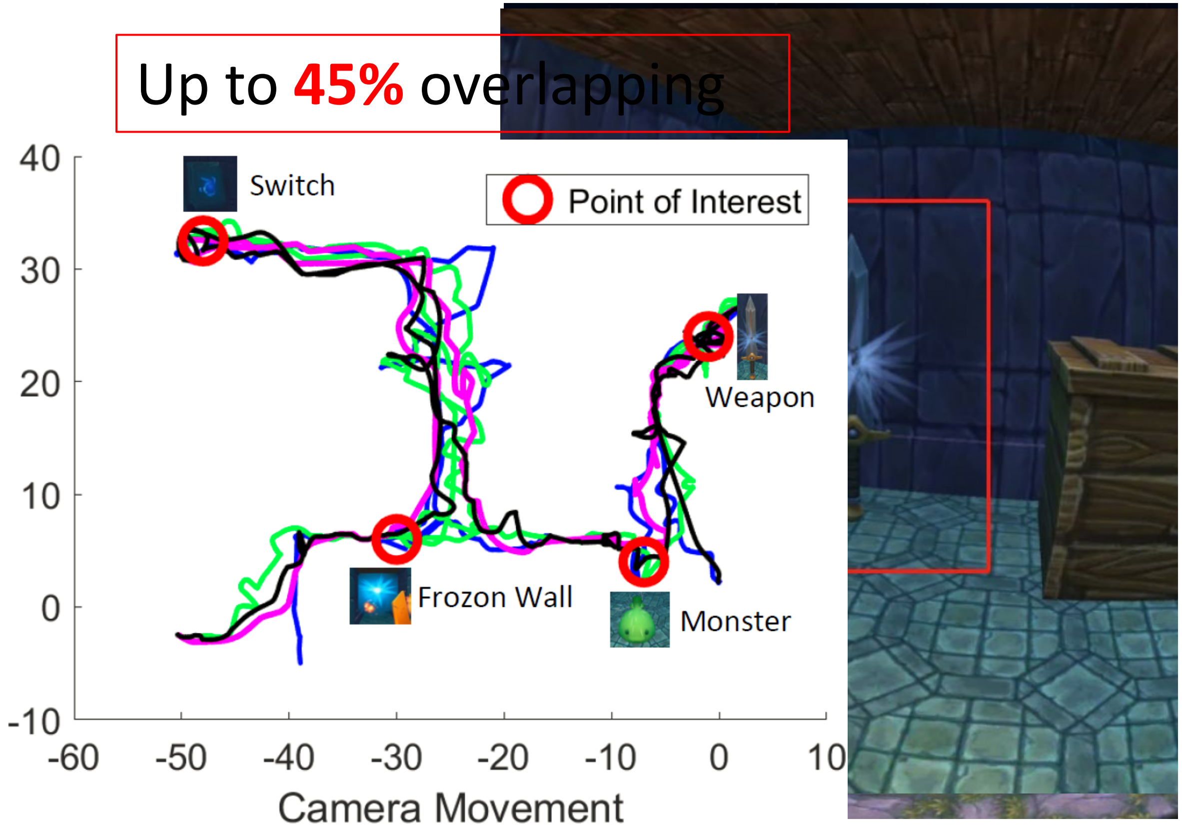
Motivation: Redundancy in VR Frames

- Frame redundancy
  - Locality of user movements in the VR world
  - Identical or similar background views among different users
- Pixel redundancy
  - Majority of pixels retained across consecutive frames

Frame Redundancy

- Spatial locality of user movement
  - Overlapped movement trajectories centered at Points of Interests (PoIs)

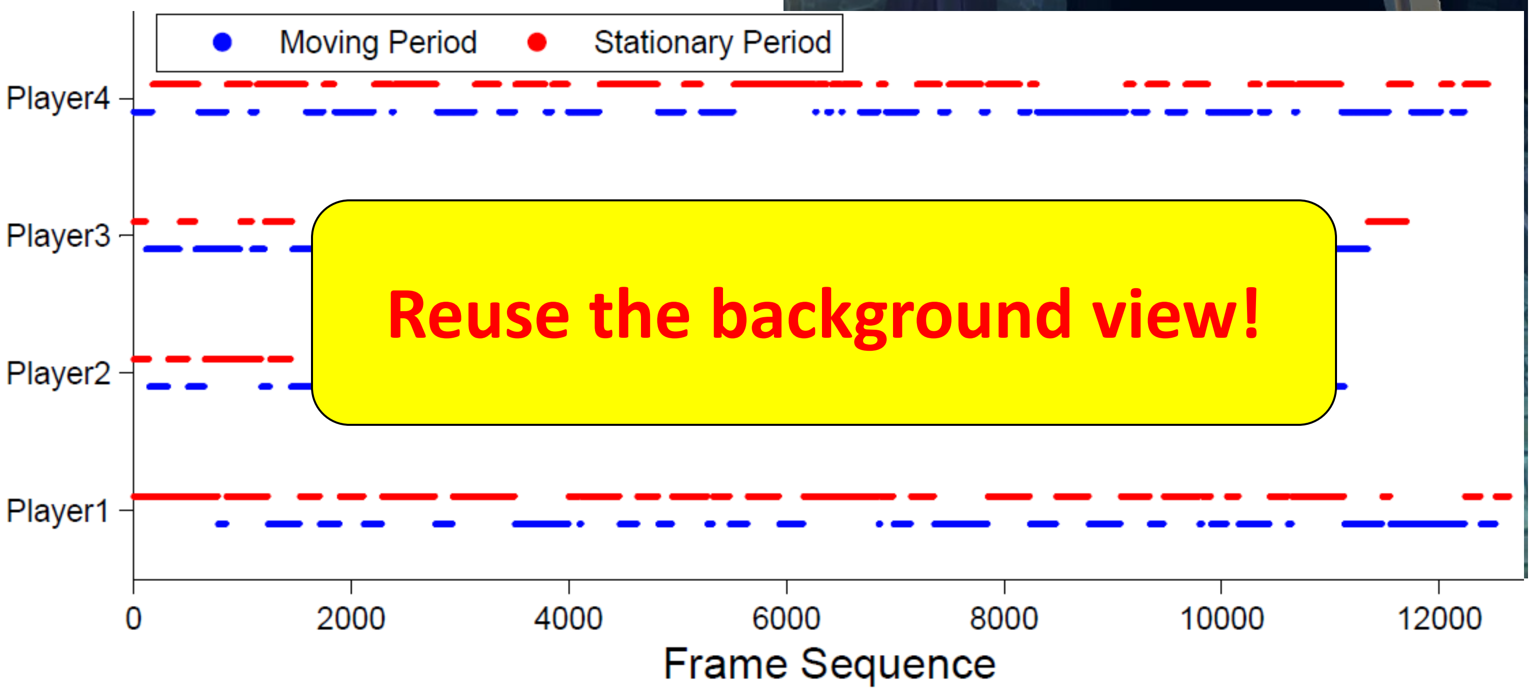
Up to **45%** overlapping



Frame Redundancy

- Temporal locality of user movement
  - Intermittent movement: stop and interact with VR objects

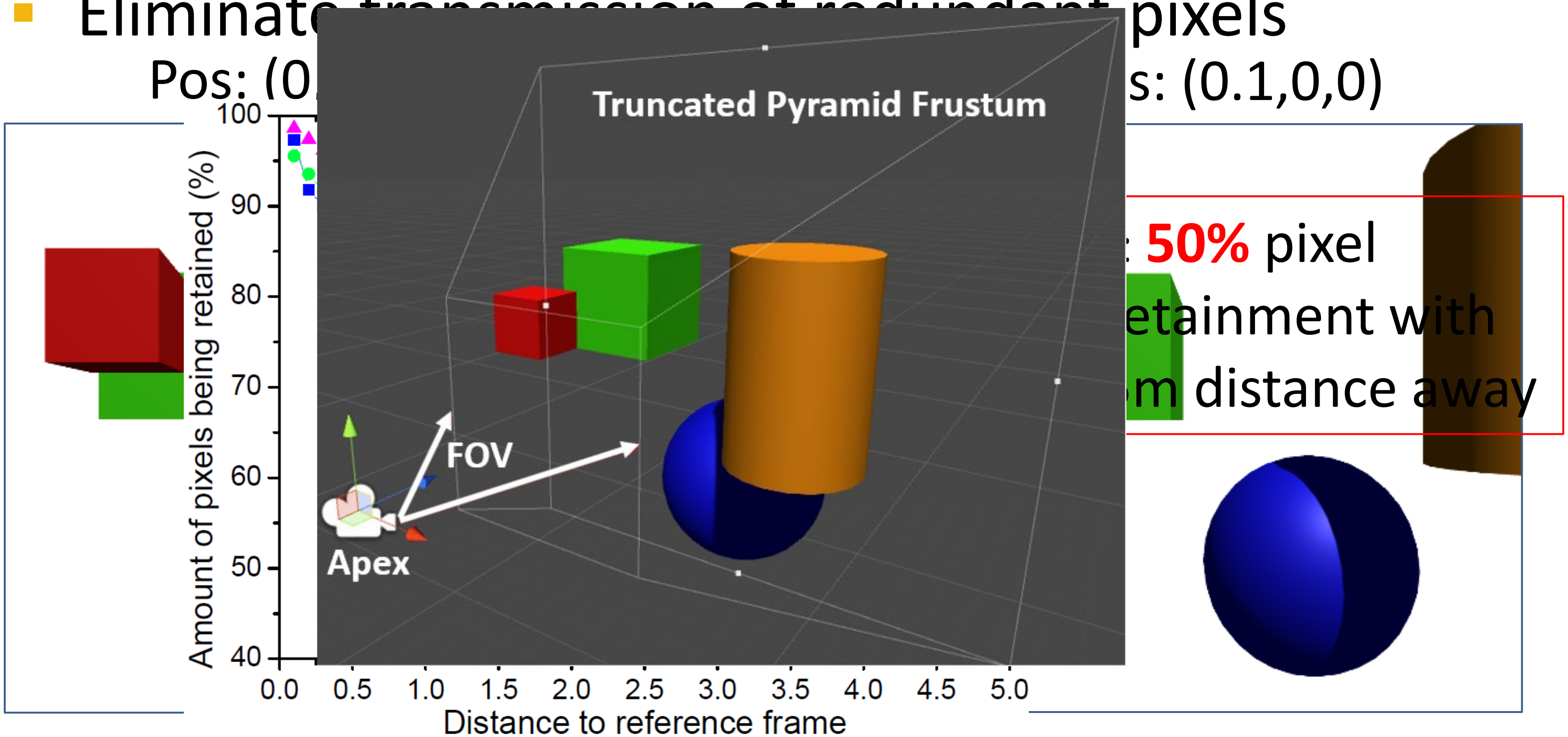
≥ **53%** of stationary time



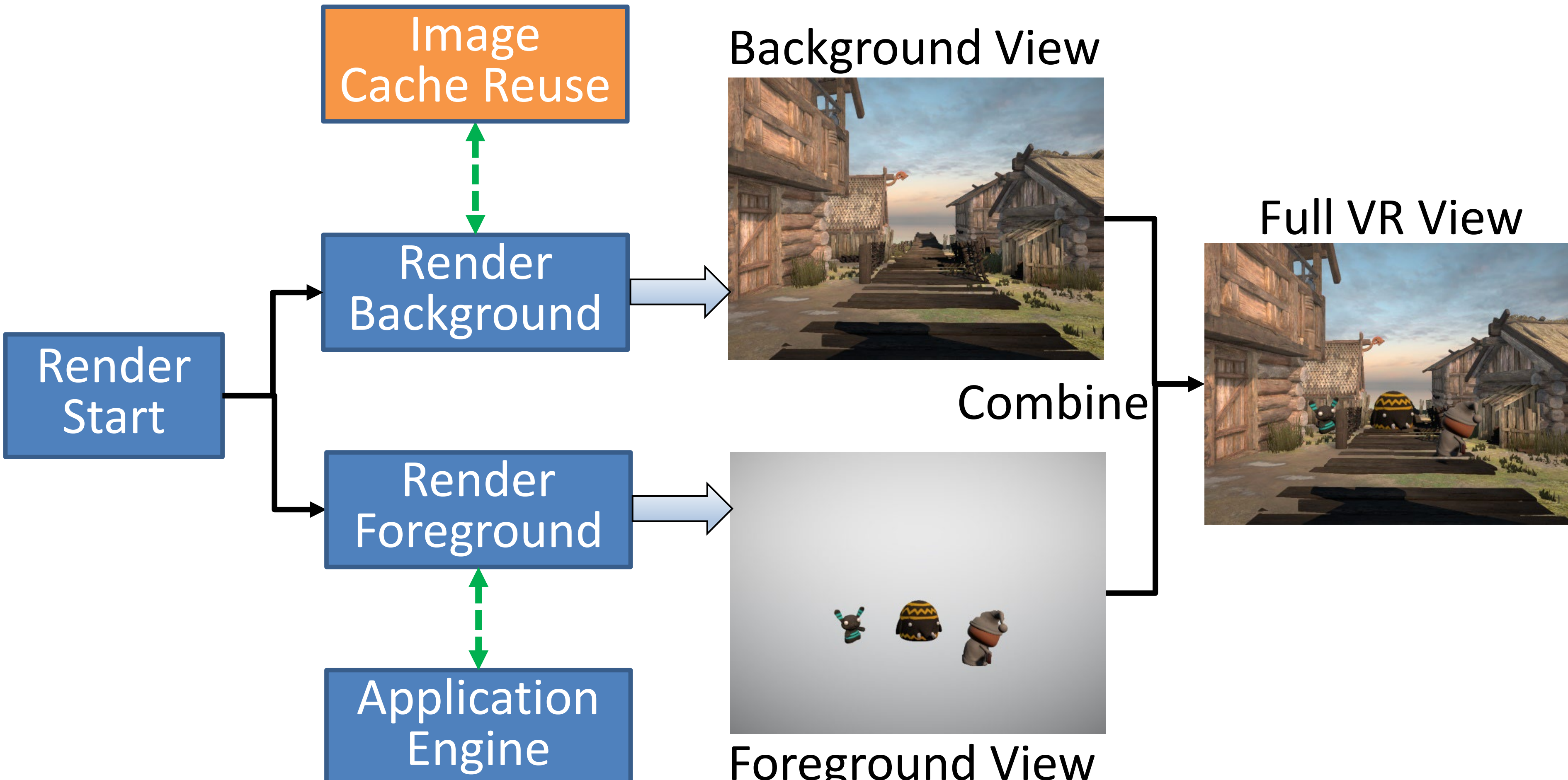
Reuse the background view!

Pixel Redundancy

- Perspective projection in VR applications
  - Project 3D object to camera surface
  - Reduce impact of user movement
- Eliminate … pixels

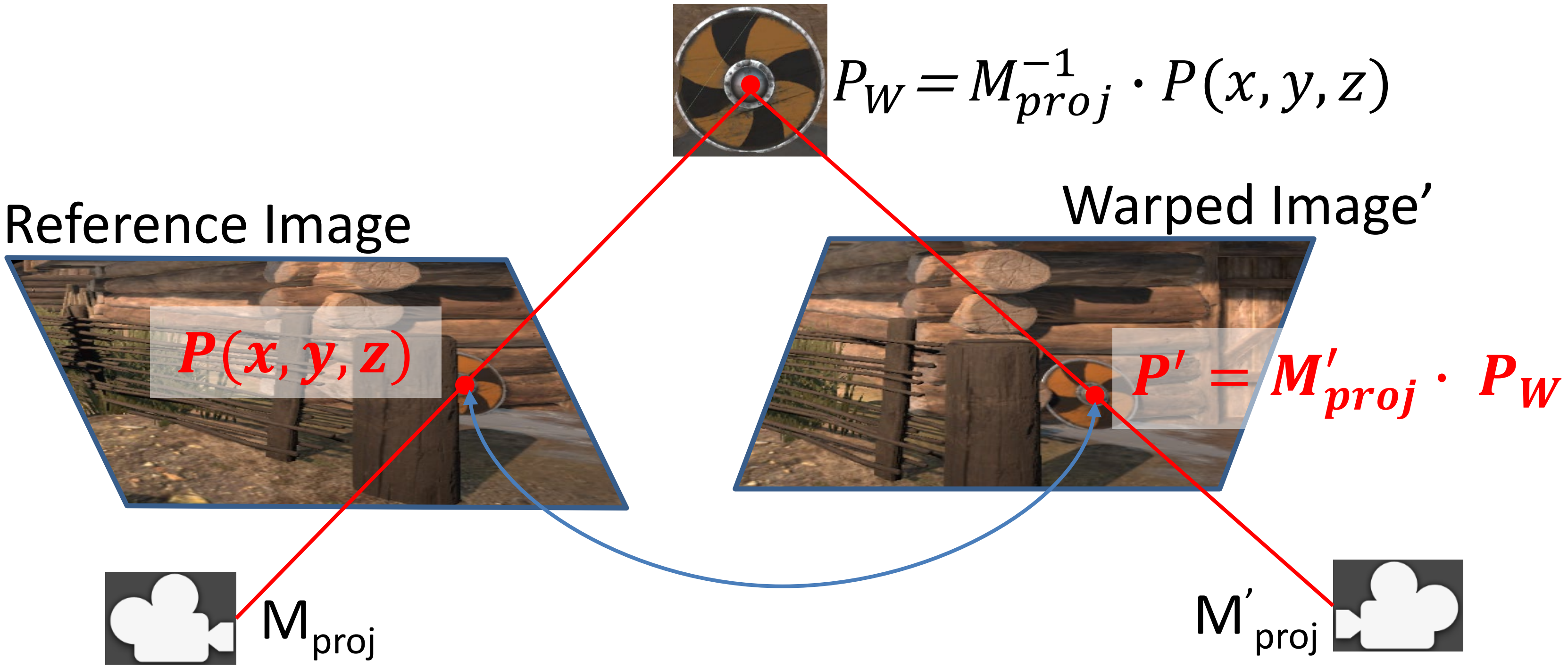


VR Frame Rendering



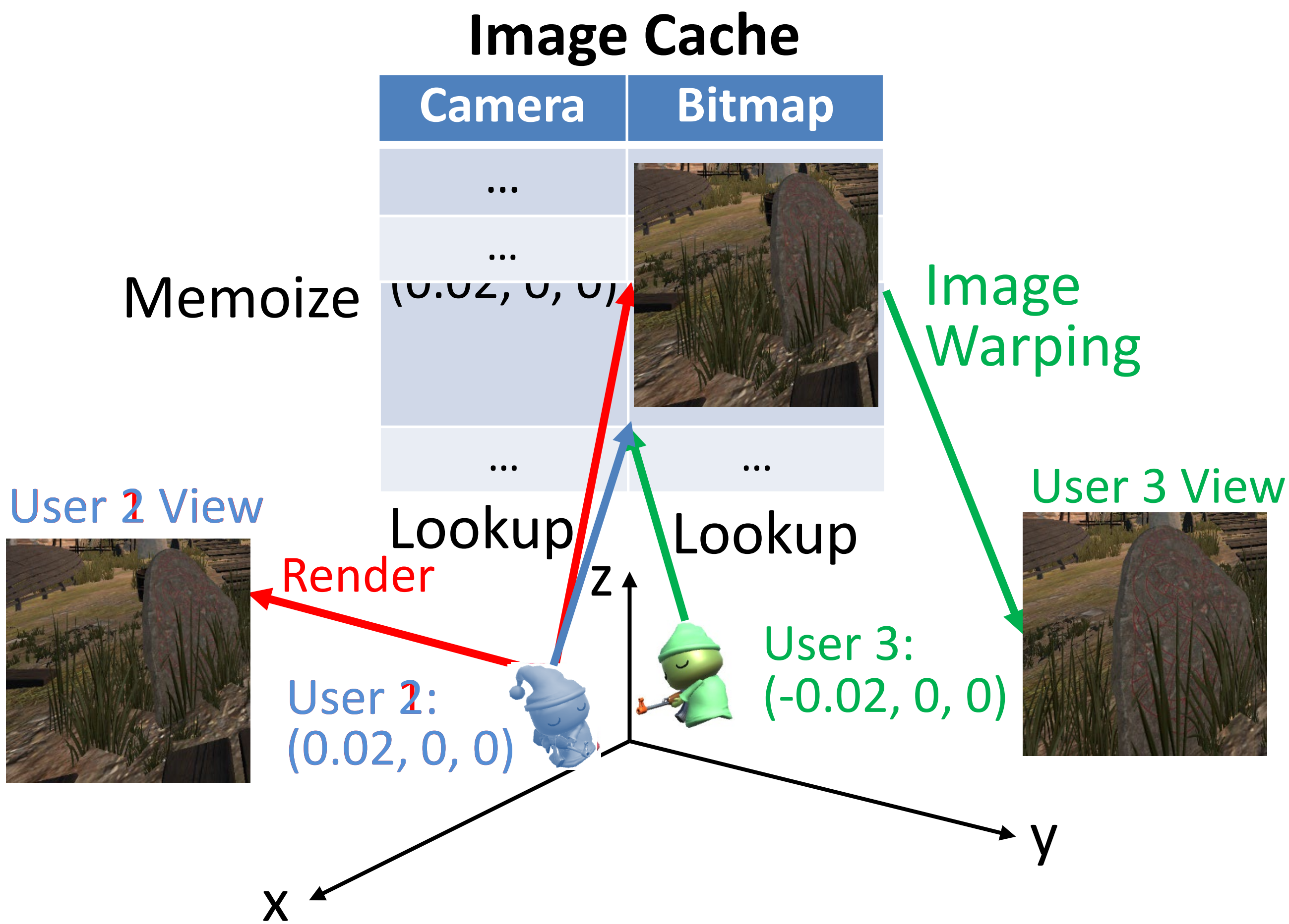
Starting Point: Image Warping

- Reproject a frame to a new camera view
  - Compensate sudden user movement
  - Avoid motion sickness



Reducing Computation Overhead

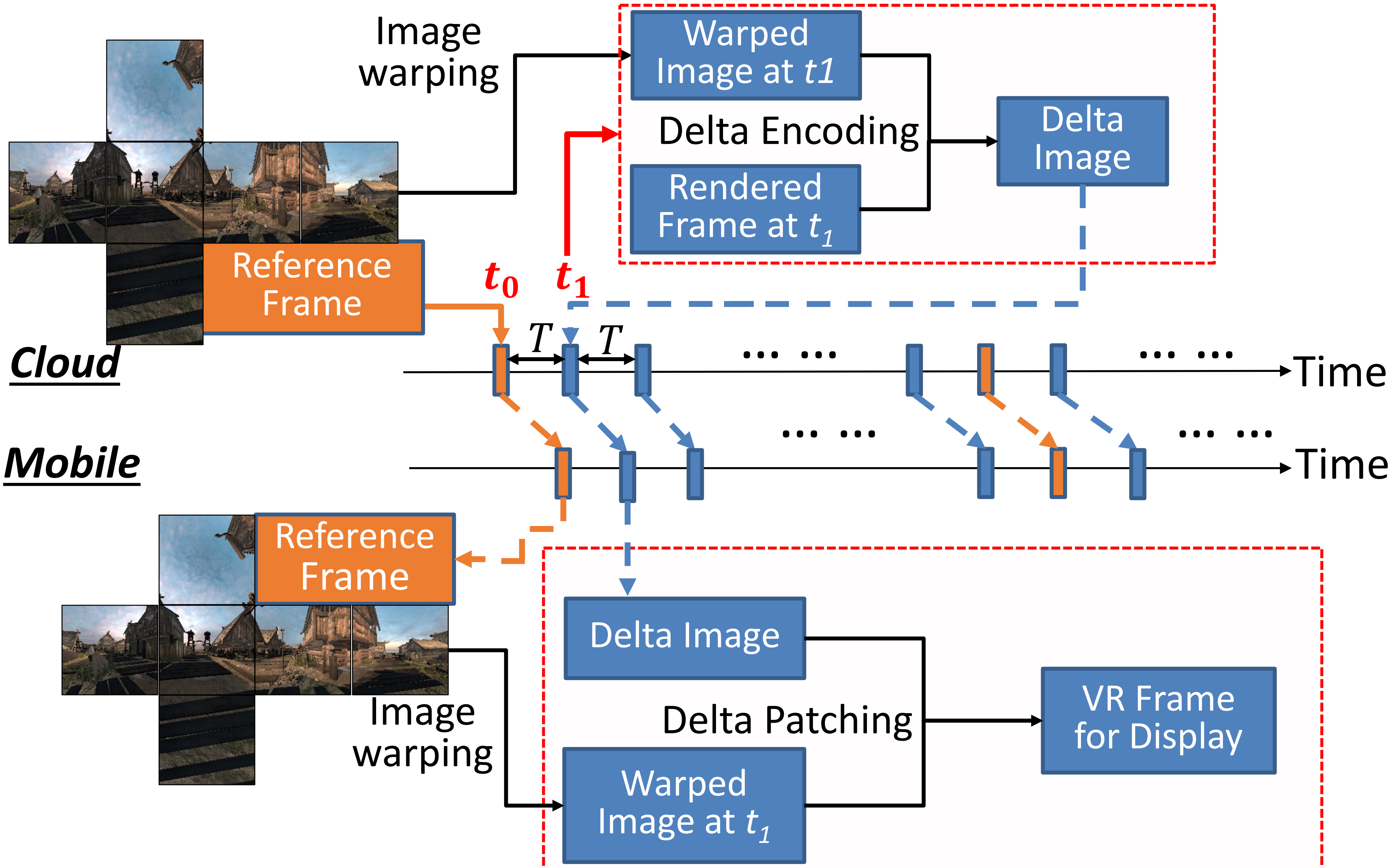
- Frame memoization to eliminate frame redundancy
- Exact match doesn't always exist



Reducing Communication Overhead

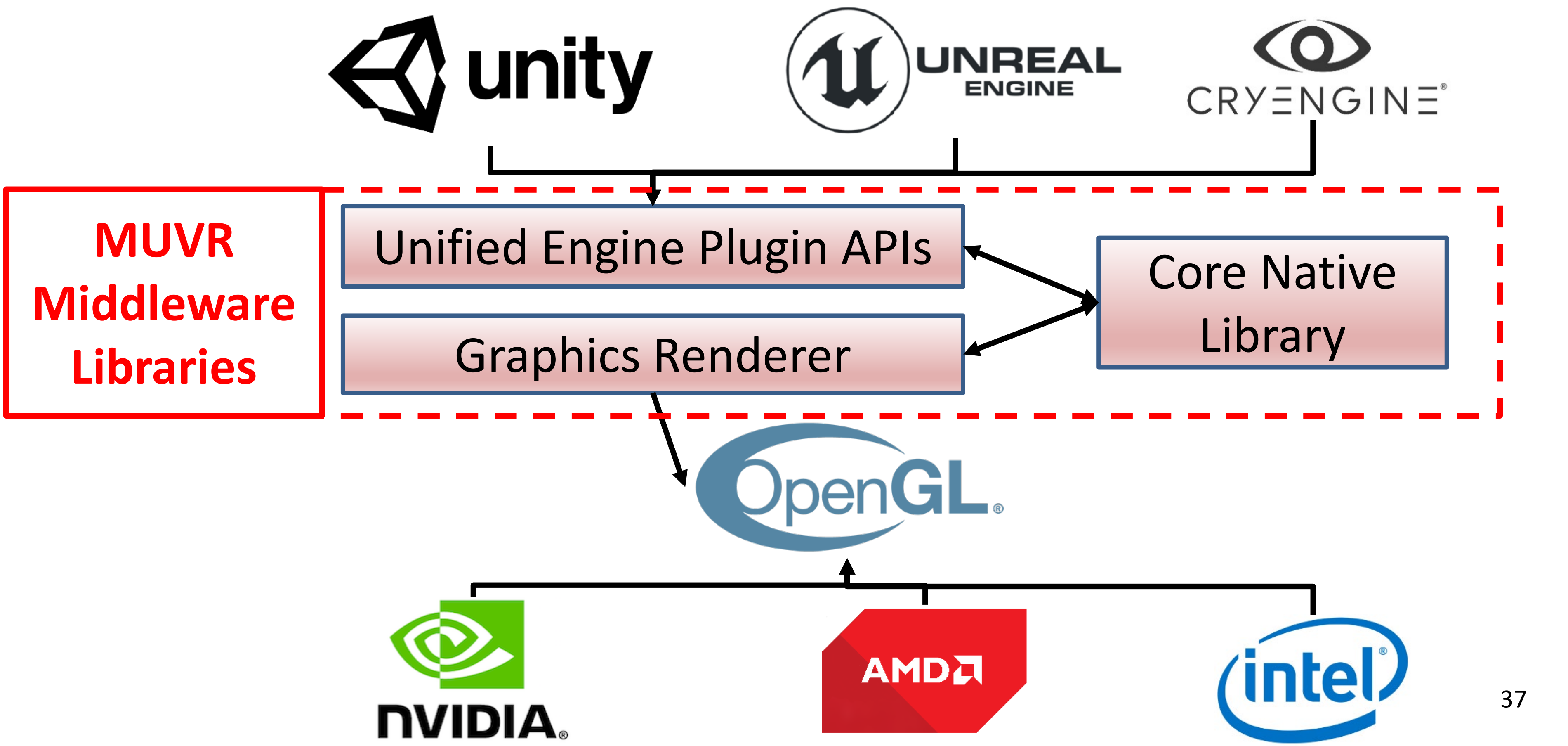
- Utilize image warping to capture redundant pixels
- Eliminate redundant pixel transmission
  - Only transmit the Deltas

Transmitting Deltas Only



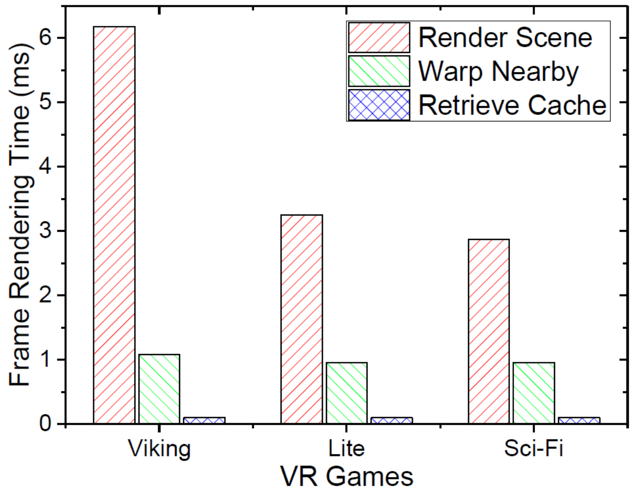
Implementation

- Key focus: Generality
  - Heterogeneous application engines
  - Heterogeneous hardware drivers



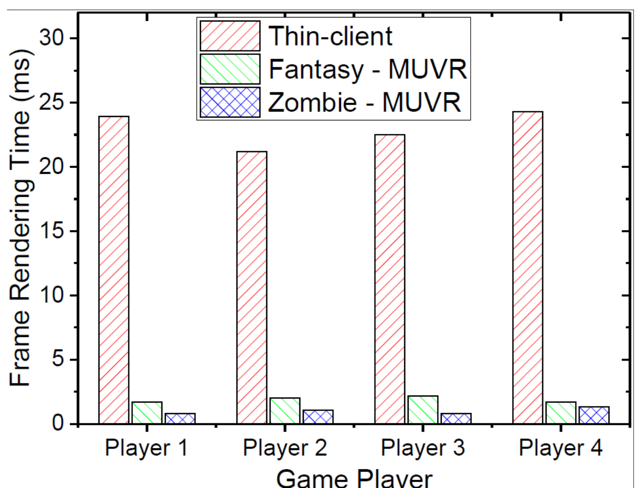
Computation Reduction

- Benchmark of background rendering



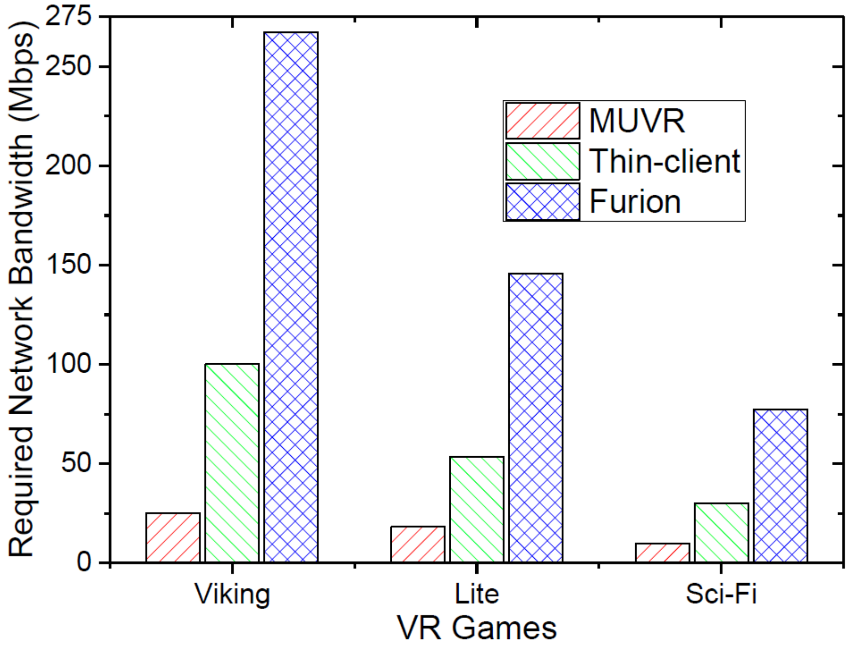
- Negligible with an exact match in the cache
- **3x** speedup with reusing a nearby image

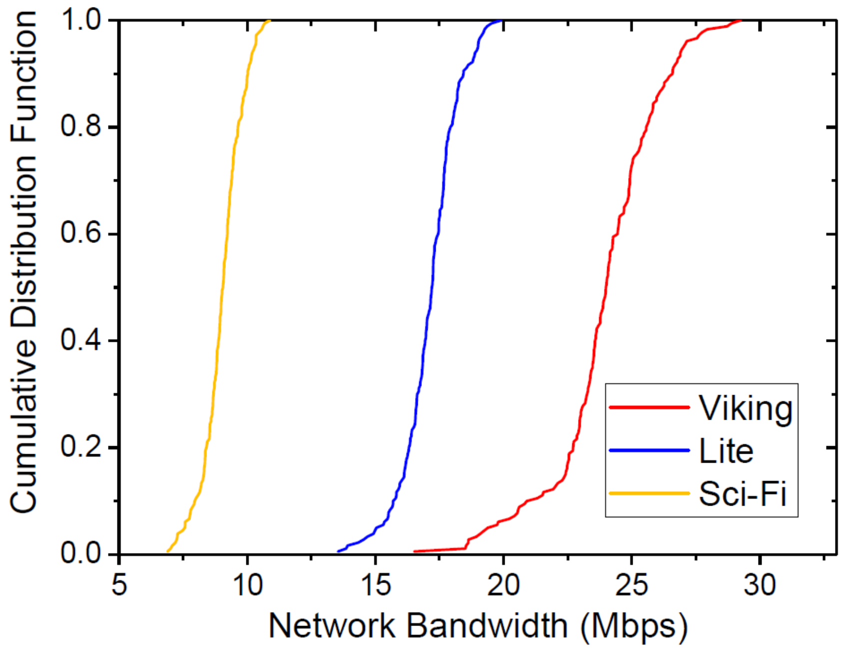
- Workload reduction



Reduce more than **90%** rendering workload

Communication Reduction

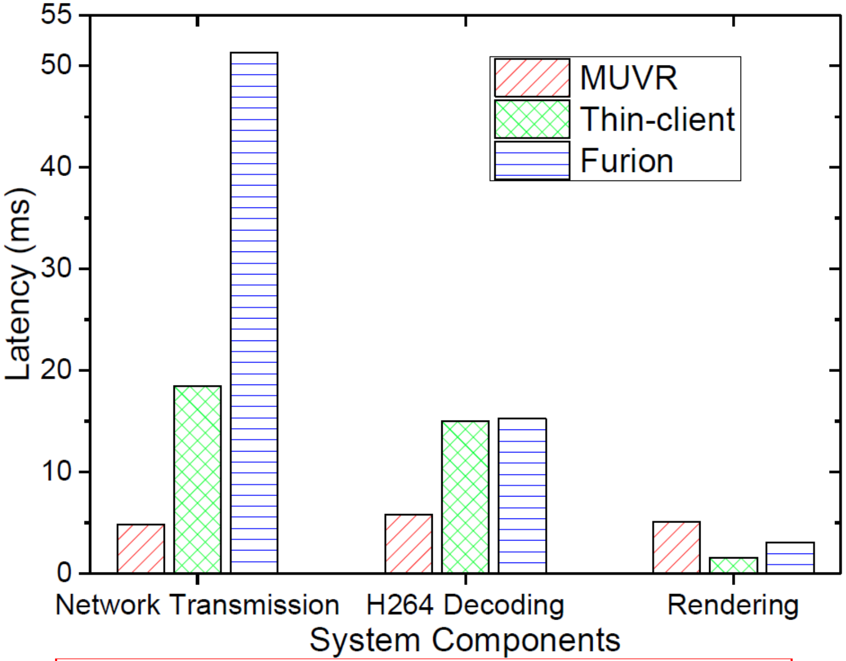




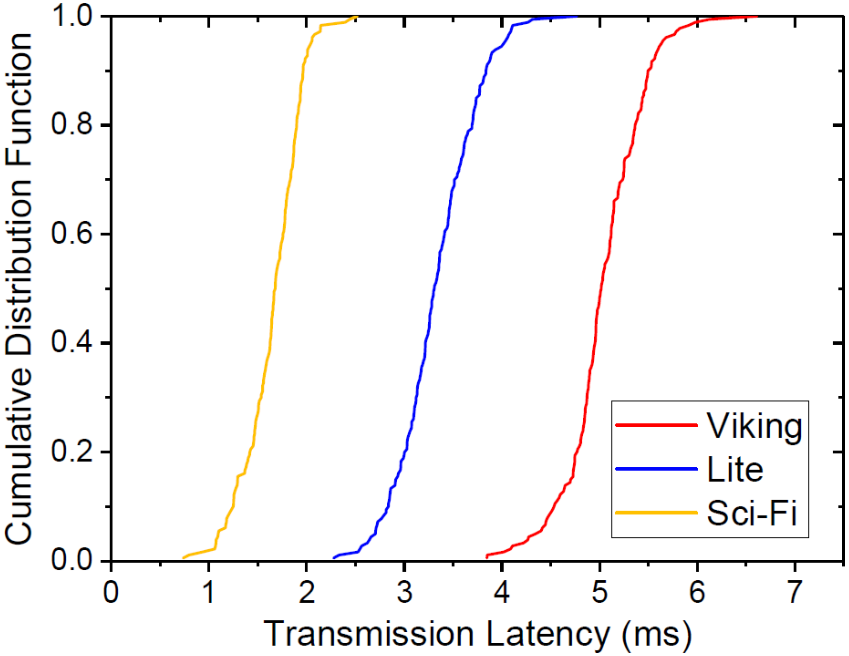
25Mbps to support a VR user

Peak bandwidth < **30Mbps**

Latency



< 16ms motion-to-photon latency



< 6ms transmission latency of delta images

Visual Quality

VR IMAGE QUALITY (SSIM)

Rendering Scheme	Viking	Lite	Sci-Fi
Local Frame Rendering	0.8133	0.8766	0.8832
Thin-client	0.8783	0.8834	0.9263
MUVR w/ Stationary User	0.9569	0.9599	0.9681
MUVR w/ Moving User	0.9241	0.9210	0.9557

SSIM > **0.92**: High image quality