BabelFish: Fusing Address Translations for Containers

Dimitrios Skarlatos, Umur Darbaz, Bhargava Gopireddy, Nam Sung Kim, and Josep Torrellas
University of Illinois at Urbana-Champaign
skarlat2@illinois.edu

ISCA 2020
Conventional Cloud Computing

Virtual Machines

- Hypervisor
- Guest OS
- Lib
- App

Hardware
New Era in Cloud Computing

1. Lightweight
2. Faster bringup
3. Higher consolidation

Containers

OS automatically shares resources!
Containers in the Cloud

Pod

- Replicated Containers
- Scalability
- Fault Tolerance
- Load Balancing

Application Binary
Library
Container Engine
Operating System
Hardware

Shared Data Pages

High number of containers
Very frequent context switches
Serverless & Function-as-a-Service (FaaS)
  - Cold start effects
  - Very short runtime
Problem: Replicated VA → PA Translations

1. TLB thrashing
2. Redundant page table work

Replicated Containers
Scalability
Fault Tolerance
Load Balancing

Function-as-a-Service (FaaS)
- Cold start effects
- Very short runtime

Replicated VA
Pod
Application Binary
Library

High number of containers
Very frequent context switches
Serverless & Function-as-a-Service (FaaS)

Cold start effects
Very short runtime
Solution: Share VA→PA Translations

1. Minimize TLB thrashing
2. Eliminate redundant page table work

Replicated Containers
- Scalability
- Fault Tolerance
- Load Balancing

Serverless & Function-as-a-Service (FaaS)
- Cold start effects
- Very short runtime
Contribution: **BabelFish**

HW and OS support to share translations across containers

1. Introduce Container Context IDentifiers (CCID)
2. Extend TLB design
3. Share page tables

Performance improvement

- Data-serving: 11%-18%
- Compute: 11%
- Function-as-a-Service: 10-55%
- Container bring-up: 8%

“The Babel fish is small, yellow, leech-like, and probably the oddest thing in the Universe. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language.”

*The Hitchhiker's Guide to the Galaxy – Douglas Adams*
Problem: TLB Bloat
Problem: TLB Bloat

Container A

Core

L1 Cache

L2 Cache

L3 Cache

Main Memory

Page Tables A

Issue LD VA 1

TLB

TLB Miss → “Page Walk” = Fetch entry from page table
Problem: TLB Bloat

Container A

Core → L1 Cache → L2 Cache → L3 Cache → TLB

Main Memory

Page Tables A

PA 4

PA 4

VA 1

VA 1

PCID

Existing Process Context Identifiers (PCID)
Problem: TLB Bloat

What if we could identify shared entries?

TLB Miss → “Page Walk” = Fetch entry from page table
Problem: TLB Bloat

What if we could identify shared entries?
Solution: BabelFish TLB

Container Context Identifier (CCID)

Core → L1 Cache → L2 Cache → L3 Cache

TLB

Main Memory

Page Tables B

Unlimited sharing of translations among any number of containers!
Solution: BabelFish TLB

Container Context Identifier (CCID)

Container A  Container B  Container C

Core  L1 Cache  L2 Cache  L3 Cache

Main Memory

Page Tables B

PA 4

PrivateCopy (PC)

VA→PA shared for some containers & private for others
Solution: BabelFish TLB

Container Context Identifier (CCID)

Container A  Container B  Container C

Core  L1 Cache  L2 Cache  L3 Cache

TLB

Main Memory

Container Context Identifier (CCID)

PrivateCopy (PC)

VA→PA shared for some containers & private for others
Problem: Page Table Entry Bloat
Problem: Page Table Entry Bloat
Problem: Lazy Page Table Management

Container A

Core → L1 Cache → L2 Cache → L3 Cache → TLB

Present bit

Main Memory

Page Tables A

Page Tables B

PA 4

VA 1
PA 4
P

VA 1
PA 4
P
Problem: Lazy Page Table Management

Container A

Present bit currently cleared

Main Memory

Page Tables A

VA 1 PA 4 P

Page Tables B

VA 1 PA 4 P

Core

L1 Cache

L2 Cache

L3 Cache

TLB
Problem: Lazy Page Table Management

Container A

Core

L1 Cache

L2 Cache

L3 Cache

TLB

Page Fault!

Issue LD VA 1

Main Memory

Page Tables A

Page Tables B

Storage

PA 4

VA 1

PA 4

P

PA 4

VA 1

PA 4

P
Problem: Lazy Page Table Management

Container A

Core

L1 Cache

L2 Cache

L3 Cache

TLB

Present bit set

Present bit remains cleared

Issue LD VA 1

Page Fault!

Main Memory

Storage

Page Tables A

Page Tables B

Pa 4

Va 1

Pa 4

P
Problem: Lazy Page Table Management

Container B

Core

L1 Cache

L2 Cache

L3 Cache

TLB

Main Memory

Page Tables A

Page Tables B

PA 4

VA 1

PA 4

P

Issue LD VA 1

Present bit set

Page Fault!
Conventional Page Tables

Virtual Address

47 … 39  38 … 30  29 … 21  20 … 12  11 … 0

9-bits  9-bits  9-bits  9-bits  12-bits

CR3₀

PGD

PUD

PMD

PTE

To TLB
Conventional Page Tables
Conventional Page Tables

- **Container A**
  - CR30
  - PGD
- **Container B**
  - CR31
  - PGD

Virtual Address:
- 47 ... 39
- 38 ... 30
- 29 ... 21
- 20 ... 12
- 11 ... 0

PTE:
- pmd_t
- pte_t

To TLB
Solution: BabelFish Page Table Sharing

Container A

Container B

Virtual Address

47 ... 39  38 ... 30  29 ... 21  20 ... 12  11 ... 0

9-bits  9-bits  9-bits  9-bits  12-bits

CR3_0

CR3_1

PGD  PUD  PMD  PTE

pgm_t  pte_t

To TLB
Example

Container A: Runs on Core 0, requests VA → PA

Container B: Runs on Core 1, requests VA → PA

Container C: Runs on Core 0, requests VA → PA
Example

Conventional

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
</table>

Page Walk Cache (PWC)

L2 Cache Miss, L3 Cache Miss, Main Memory Hit
## Example

<table>
<thead>
<tr>
<th>Conventional</th>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Issue LD V1</td>
<td>L1 TLB Miss</td>
<td>L2 TLB Miss</td>
<td>PWC Miss</td>
<td>PGD Walk</td>
<td>PUD Walk</td>
<td>PMD Walk</td>
<td>PTE Walk</td>
<td>Page Fault</td>
</tr>
<tr>
<td>Core 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

<table>
<thead>
<tr>
<th>Conventional</th>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conventional Core 0

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conventional Core 1

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conventional Core 1

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example

<table>
<thead>
<tr>
<th>Time</th>
<th>Conventional</th>
<th>BabelFish</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Issue LD V1</td>
<td>Issue LD V1</td>
</tr>
<tr>
<td>Core 0</td>
<td>L1 TLB Miss</td>
<td>L1 TLB Miss</td>
</tr>
<tr>
<td></td>
<td>L2 TLB Miss</td>
<td>L2 TLB Miss</td>
</tr>
<tr>
<td></td>
<td>PWC Miss</td>
<td>PWC Miss</td>
</tr>
<tr>
<td></td>
<td>PGD Walk</td>
<td>PGD Walk</td>
</tr>
<tr>
<td></td>
<td>PUD Walk</td>
<td>PUD Walk</td>
</tr>
<tr>
<td></td>
<td>PMD Walk</td>
<td>PMD Walk</td>
</tr>
<tr>
<td></td>
<td>PTE Walk</td>
<td>PTE Walk</td>
</tr>
<tr>
<td></td>
<td>Page Fault</td>
<td>Page Fault</td>
</tr>
</tbody>
</table>

#### Conventional
- Issue LD V1 Core 0
- L1 TLB Miss
- L2 TLB Miss
- PWC Miss
- PGD Walk
- PUD Walk
- PMD Walk
- PTE Walk
- Page Fault

#### BabelFish
- Issue LD V1 Core 0
- L1 TLB Miss
- L2 TLB Miss
- PWC Miss
- PGD Walk
- PUD Walk
- PMD Walk
- PTE Walk
- Page Fault
### Example

<table>
<thead>
<tr>
<th>Core 0</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue LD V1</td>
<td>L1 TLB Miss</td>
<td>L2 TLB Miss</td>
<td>PWC Miss</td>
<td>PGD Walk</td>
<td>PUD Walk</td>
<td>PMD Walk</td>
<td>PTE Walk</td>
<td>Page Fault</td>
</tr>
<tr>
<td>Issue LD V1</td>
<td>L1 TLB Miss</td>
<td>L2 TLB Miss</td>
<td>PWC Miss</td>
<td>PGD Walk</td>
<td>PUD Walk</td>
<td>PMD Walk</td>
<td>PTE Walk</td>
<td>Page Fault</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Core 1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue LD V1</td>
<td>L1 TLB Miss</td>
<td>L2 TLB Miss</td>
<td>PWC Miss</td>
<td>PGD Walk</td>
<td>PUD Walk</td>
<td>PMD Walk</td>
<td>PTE Walk</td>
<td>Page Fault</td>
</tr>
<tr>
<td>Issue LD V1</td>
<td>L1 TLB Miss</td>
<td>L2 TLB Miss</td>
<td>PWC Miss</td>
<td>PGD Walk</td>
<td>PUD Walk</td>
<td>PMD Walk</td>
<td>PTE Walk</td>
<td>Page Fault</td>
</tr>
</tbody>
</table>

**Conventional**

**BabelFish**

L2 Cache Miss, L3 Cache Hit
### Example

#### Conventional

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### BabelFish

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **L2 Cache Miss, L3 Cache Hit**
Example

Conventional

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BabelFish

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Miss</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PUD Walk</td>
<td>PMD Walk</td>
<td>PTE Walk</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue LD V1</th>
<th>L1 TLB Hit</th>
<th>L2 TLB Miss</th>
<th>PWC Miss</th>
<th>PGD Walk</th>
<th>PUD Walk</th>
<th>PMD Walk</th>
<th>PTE Walk</th>
<th>Page Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Result: Faster Execution Time!

L2 Cache Miss, L3 Cache Hit
Workloads & Methodology

Workloads:
- Data serving: ArangoDB, MongoDB, HTTPd
- Compute: GraphChi, FIO
- FaaS: Parse, Hash, Marshal (both dense and sparse)

Full system simulations with Simics + SST
Page Table Sharing for Two Containers

Data Serving and Compute

<table>
<thead>
<tr>
<th>PTE Entries (%)</th>
<th>MongoDB</th>
<th>ArangoDB</th>
<th>HTTPd</th>
<th>Graphchi</th>
<th>FIO</th>
<th>Average</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5-minute run on a real machine
Page Table Sharing for Two Containers

![Bar chart showing PTE Entries (%) for different applications and functions. The chart indicates a 5-minute run on a real machine.]
Page Table Sharing for Two Containers

- Unshareable
- Shareable

Data Serving and Compute

Functions

PTE Entries (%)

MongoDB
ArangoDB
HTTPd
Graphchi
FIO
Average
Functions

5-minute run on a real machine
Page Table Sharing for Two Containers

5-minute run on a real machine

Data Serving and Compute

- MongoDB
- ArangoDB
- HTTPd
- Graphchi
- FIO

Unshareable
Shareable

Active

Functions

Average
Page Table Sharing for Two Containers

5-minute run on a real machine
Page Table Sharing for Two Containers

Page Sharing is Common
→ Major Reduction in Translations

5-minute run on a real machine
Performance

Major Performance Gains
→ Both from TLB and page tables
More in the Paper

Copy-on-Write pages
Address Space Layout Randomization (ASLR) for security
Other security considerations
More evaluation
  Area, energy, sharing, comparison vs larger TLB
Takeaway: **BabelFish**

**Sharing Translations in TLB and Page Tables**

Introduces Container Context ID (CCID) at OS and HW
TLB extensions & Page Table sharing support
Substantial speedup in application execution and container bring-up

Future work → CCID in OS and HW
- Virtualized environments and nested page tables
- Other resources
BabelFish: Fusing Address Translations for Containers

Dimitrios Skarlatos, Umur Darbaz, Bhargava Gopireddy, Nam Sung Kim, and Josep Torrellas
University of Illinois at Urbana-Champaign
skarlat2@illinois.edu

ISCA 2020