

Copy On Write Image

ภาคการศึกษา 1 2564

กษิติศ ขาญเขียว

สาขาวิชาวิทยาการคอมพิวเตอร์

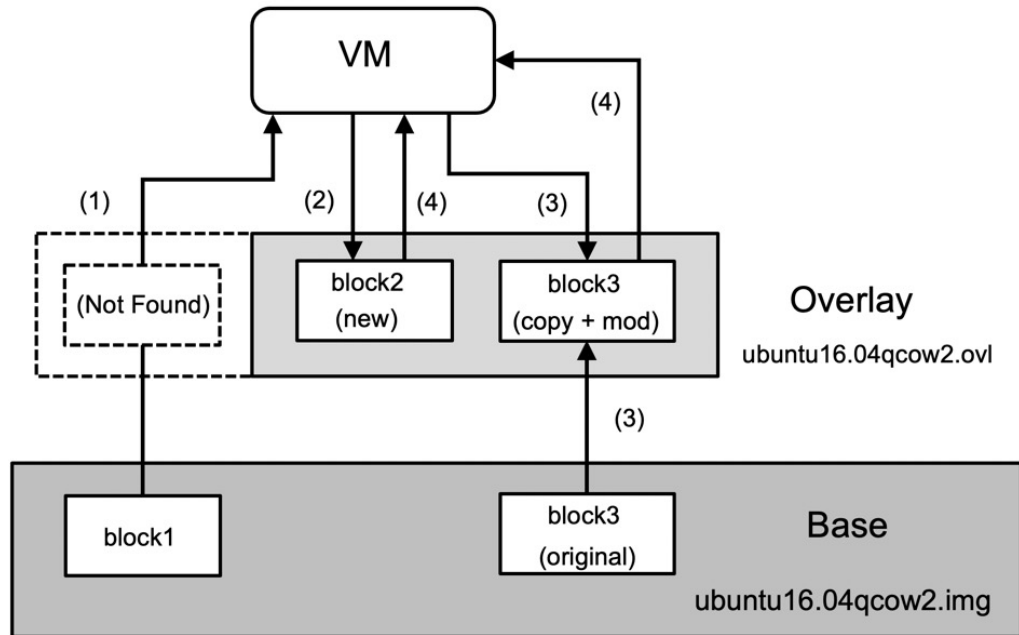
คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัย ธรรมศาสตร์

ก๊อปปี้ออนไรท์ (Copy on Write) อิมเมจ

- อนุญาตให้ผู้สร้างอิมเมจ อย่างรวดเร็วโดย ดาว์นโหลดเฉพาะอิมเมจที่จำเป็นต่อการใช้งาน แทนที่จะสร้างหรือดาว์นโหลดไฟล์ซิสเต็มขนาดใหญ่มาทีเดียว
- เบสอิมเมจ (base) คือข้อมูลตั้งต้น (อาจเป็นส่วนใหญ่ที่มักจะเปลี่ยนแปลง)
- โอเวอร์เลย์ คือข้อมูลการเปลี่ยนแปลงที่เกิดกับเบสอิมเมจ หรือโอเวอร์เลย์อิมเมจชั้นก่อนหน้า
- โอเวอร์เลย์ สามารถมีได้หลายชั้น โดยมีโอเวอร์เลย์อื่นเป็นเบส
- การเปลี่ยนแปลงในปัจจุบัน จะเกิดขึ้นบนวีเอ็มชั้นบนสุด

หลักการก๊อปปี้อนไรท์

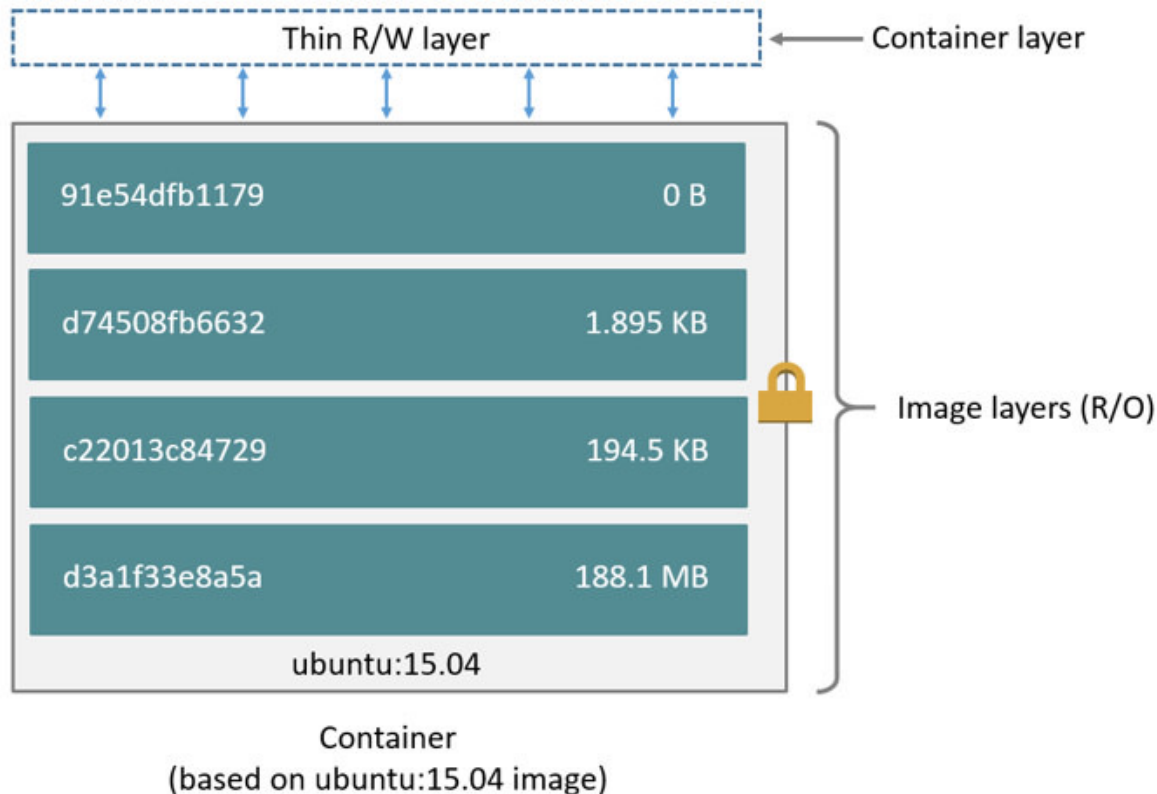


1. ถ้าหา block ใน ovl ไม่พบก็จะมองหาใน base (recursive)
2. ถ้ามีการเพิ่มข้อมูลใหม่จะเพิ่มใน ovl
3. การเปลี่ยนแปลงข้อมูล จะก๊อปปี้อบล็อกข้อมูลจากเบสมาที่ ovl ก่อนแล้วจึงเปลี่ยนค่าใน ovl

ก๊อปปี้ออนไรท์ (Copy on Write) อิมเมจ

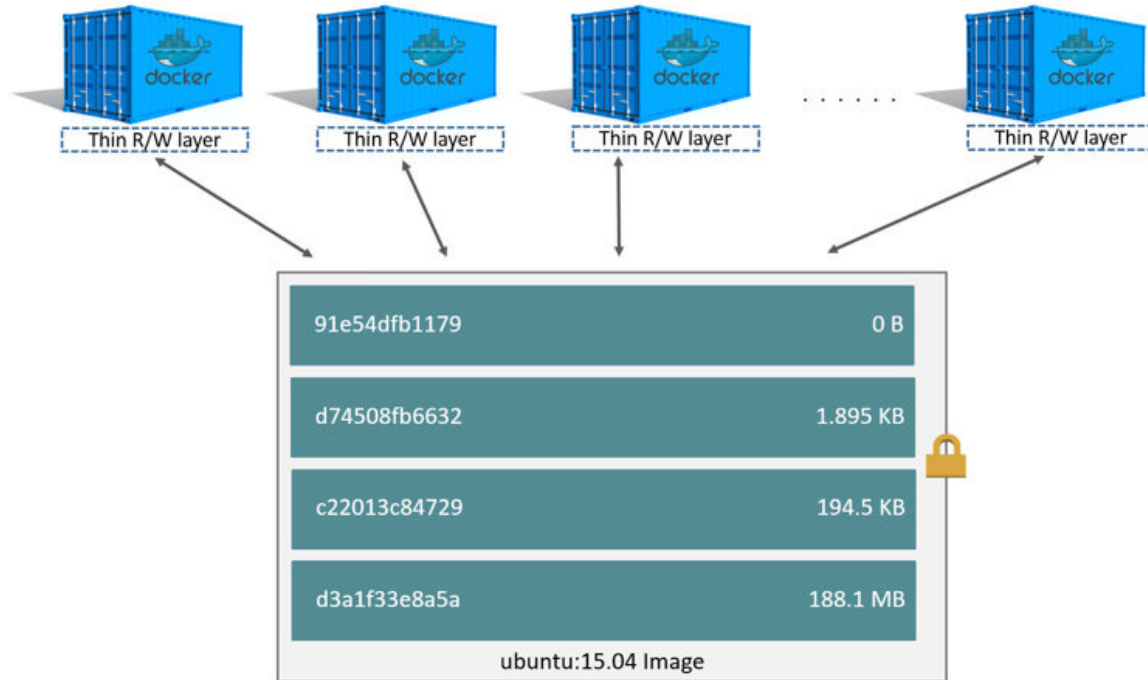
- คอนเทนเนอร์จะติดตามความเปลี่ยนแปลงที่เกิดขึ้นในอิมเมจเสมอ
- มีระบบจัดการข้อมูลแบบ CoW หลายแบบ ได้แก่
 - AUFS และ Overlay เป็นการทำ COW ในระดับไฟล์
 - device mapper เป็นการทำ COW ในระดับบล็อกข้อมูล
 - ZFS, BTRFS เป็นการทำ COW ในระดับไฟล์ซิสเต็ม
- เนื่องจากอิมเมจมีขนาดเล็ก ทำให้สร้าง และบูทได้อย่างรวดเร็ว

โครงสร้างของอิมเมจแบบ CoW ของ Docker (1)



- Dockerd จัดการเก็บอิมเมจเป็น layer
- จัดการ layer ด้วยหลักการ CoW
- แต่ละ layer เก็บการเปลี่ยนแปลงที่ไม่ใหญ่มาก ทำให้สามารถ pull และนำมาสร้าง image ใหม่ได้อย่างรวดเร็ว
- การจัดการเก็บจริงขึ้นอยู่กับ storage driver เช่น Overlay2, aufs, ZFS, Btrfs, etc.

โครงสร้างของอิมเมจแบบ CoW ของ Docker (1)



- หลักการ CoW อนุญาตให้ หลาย container สามารถ share read-only layer ร่วมกันได้
- การเปลี่ยนแปลงจะถูกเก็บใน storage layer ชั้นบนสุด
- อิมเมจไฟล์ของแต่ละคอนเทนเนอร์อยู่ที่ `/var/lib/docker/<storage driver>`

สรุป

- แนะนำหลักการ Copy On Write