CS438

Cgroup

Lecture 13

1st Semester 2021

Kasidit Chanchio

Department of Computer Science

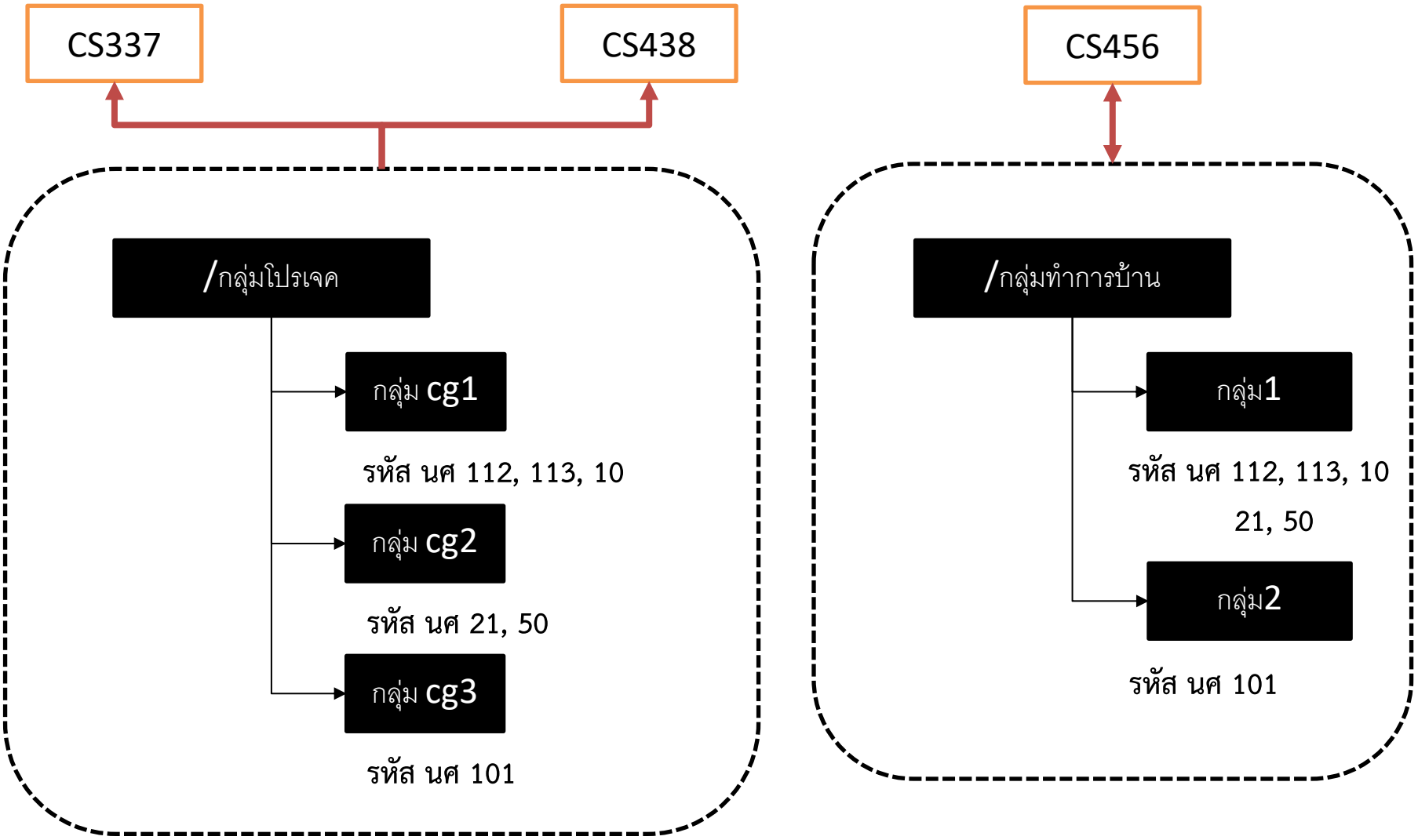
เนื้อหาเรียบเรียงจาก https://www.youtube.com/watch?v=sK5i-N34im8&t=211s

ซีกรุ๊ป

- ซีกรุ๊ปแบ่งทรัพยากรเป็นระบบย่อย (subsystem) ยกตัวอย่างเช่น ระบบย่อย หน่วยความจำ (Memory Subsystem) ระบบย่อยซีพียู (CPU subsystem) ระบบย่อยบล็อกไอโอ เป็นต้น
- ซีกรุ๊ปอนุญาตให้ผู้ใช้สร้างกรุ๊ปการใช้งานแต่ละระบบย่อย และกำหนดของเขตการใช้งานทรัพยากรในแต่ละระบบย่อยสำหรับแต่ละกรุ๊ปนั้น
- ผู้ใช้สามารถ monitor และ configure และ reconfigure และ deny access ของทรัพยากร
- สามารถเซฟและกำหนดค่า configuration ตอน boot time ได้ด้วย
- ทาสค์ (task) หมายถึงโพรเซสในลินุกซ์เคอร์เนิล

อ้างอิงจาก https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/resource_management_guide/ch01

เปรียบเทียบ (นศ, กลุ่ม, วิชา) = (โพรเซส, ซีกรุ๊ป, ระบบย่อย)



โครงสร้างต้นไม้ของซีกรุ๊ป

- ซีกรุ๊ปมีโครงสร้างประกอบไปด้วยโหนดและเอ็ดจ์เชื่อมต่อกันเป็นแผนภาพต้นไม้
- ซีกรุ๊ปโหมดมีความสัมพันธ์แบบโหนดพาเรนท์และโหนดลูก
- โหนดลูกจะได้รับคุณสมบัติของพาเรนโหนดเป็นมรดก
- ผู้ใช้สามารถสร้างแผนภาพต้นไม้ของซีกรุ๊ปได้หลายแผนภาพ (ที่ไม่เชื่อมต่อกัน)
- แผนภาพต้นไม้จะถูกนำไปแนบ (attach) กับทรัพยากรที่ถูกแบ่งออกเป็นระบบย่อย (subsystem) ดังที่จะได้กล่าวถึงต่อไป

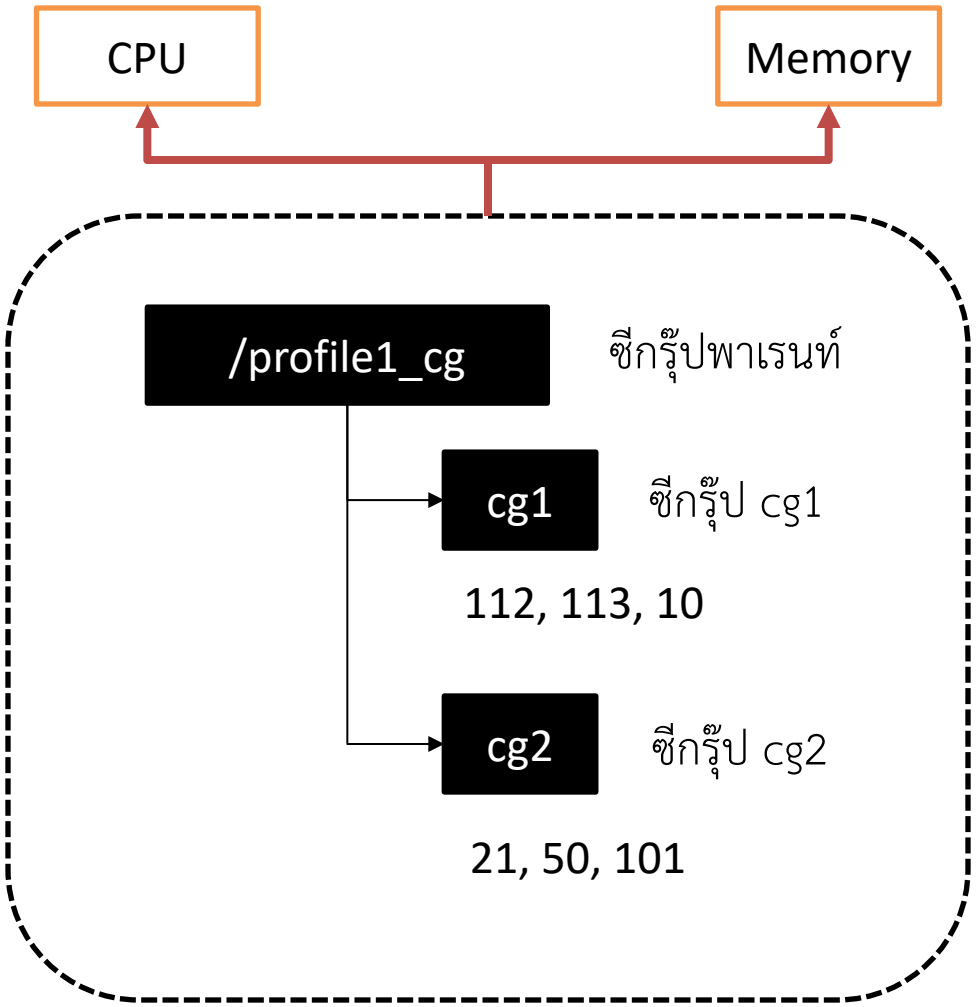
ระบบย่อย 10 แบบใน RH Linux

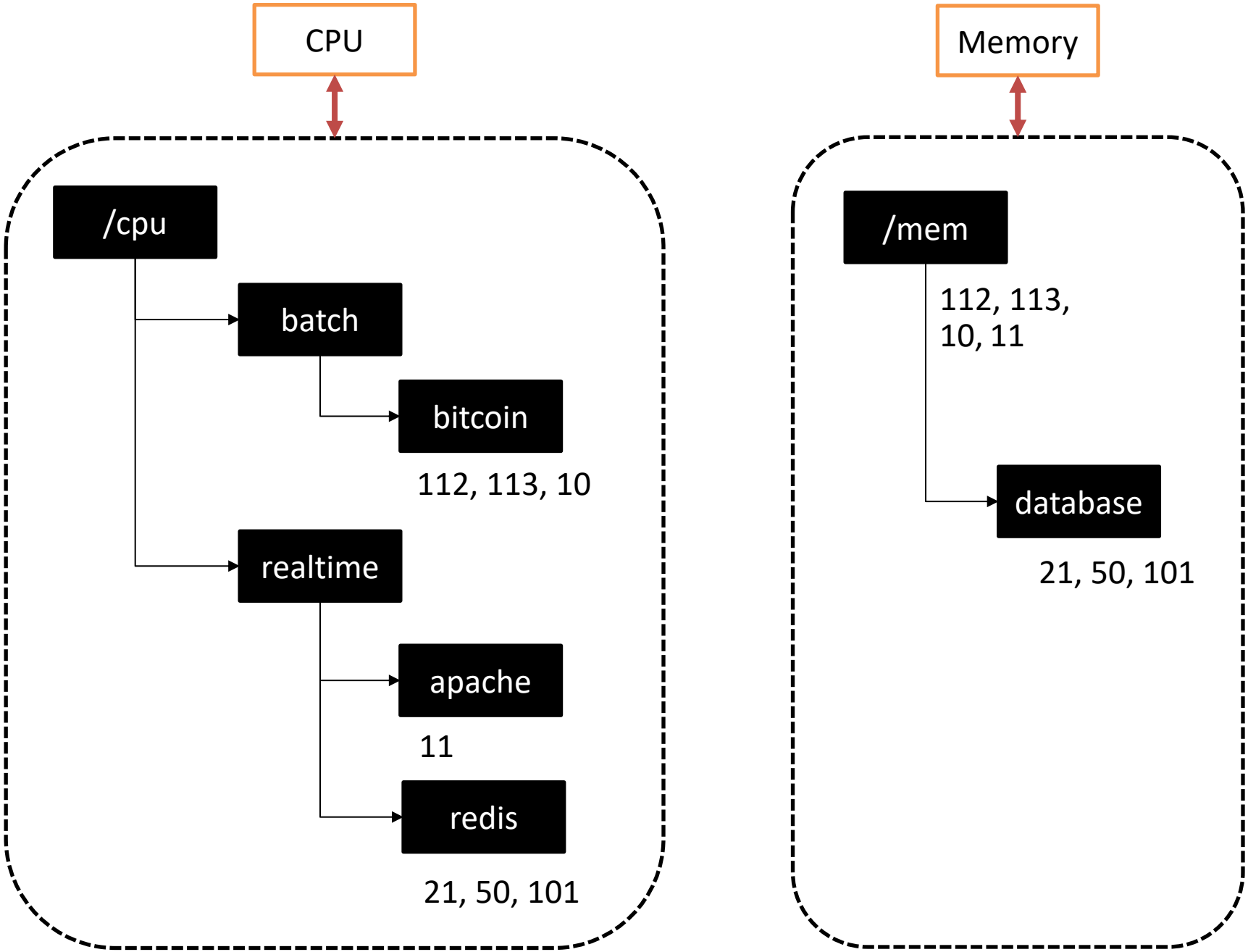
- cpu: ให้ scheduler ของลินุกซ์กำหนดการเข้าถึงซีพียูของทาสค์
- cpuacct: สร้างรายงานการใช้งานซีพียูของ(ทาสค์ใน)ซีกรุ๊ป
- cpusets: จัดสรรซีพียูคอร์และเมมโมรีโหนดให้(ทาสค์ใน)ซีกรุ๊ป
- devices: อนุญาตหรือปฏิเสธการเข้าถึงอุปกรณ์ให้(ทาสค์ใน)ซีกรุ๊ป
- blkio: กำหนดการจำกัดการเข้าถึงอุปกรณ์บล็อกไอโอ เช่น hdd,sdd,usb
- freezer: หยุดหรือรีซูมการประมวลผลของ (ทาสค์ใน)ซีกรุ๊ป
- memory: จำกัดการใช้งานหน่วยความจำและรายงานการใช้งานหน่วยความจำของ (ทาสค์ใน)ซีกรุ๊ป
- net_cls: สร้างแทก (tag) สำหรับแพคเกตสำหรับทาสค์ของซีกรุ๊ป
- net_pri: กำหนดไพออริตี้สำหรับแพคเกต ต่อ NIC
- ns: เกี่ยวกับเนมสเปซ และ
- perf_event: เช็คว่าทาสค์อยู่ในซีกรุ๊ปไหนและวัดปสภ

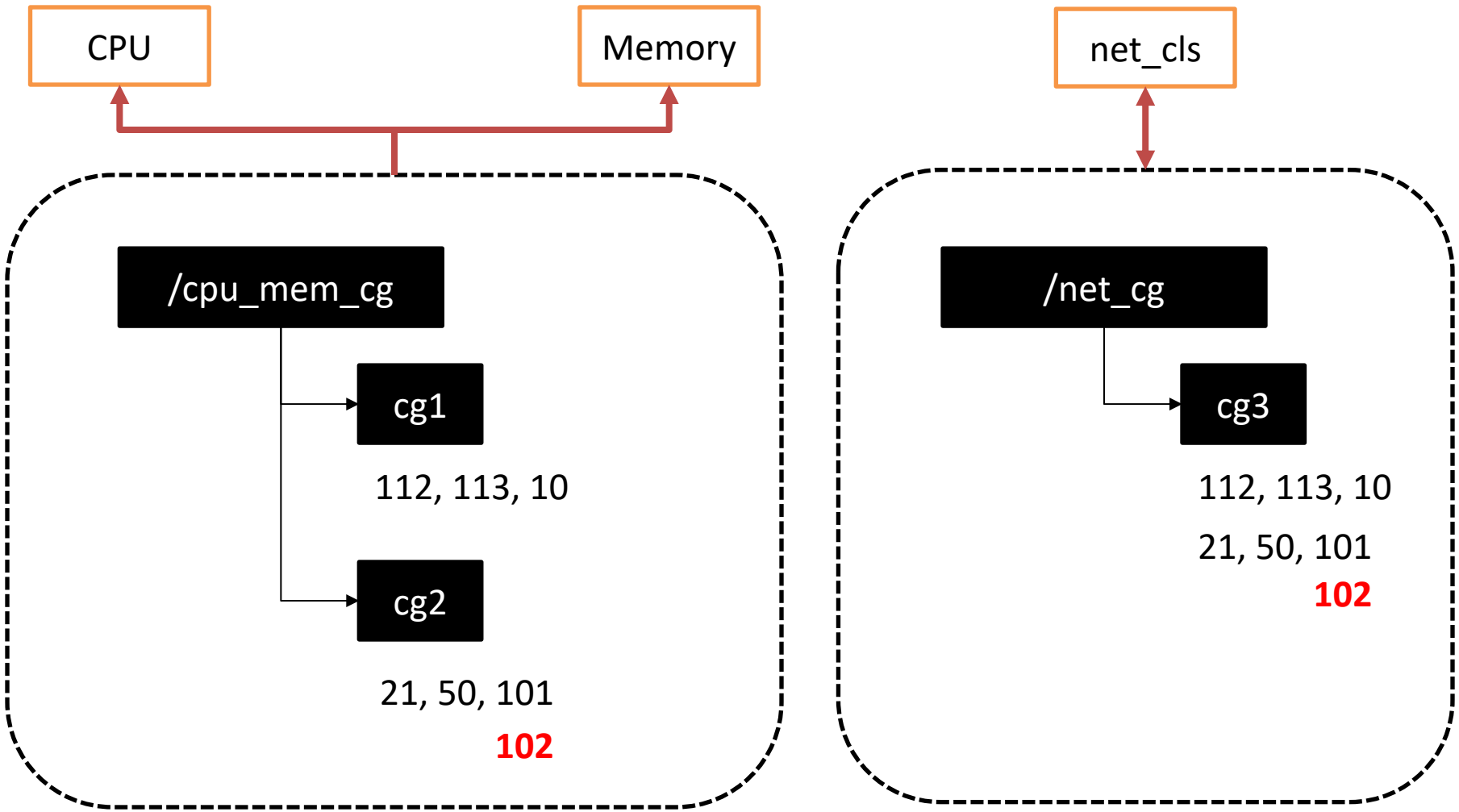
กฎความสัมพันธ์ระหว่างซีกรุ๊ปและระบบย่อย

1. เราสามารถใช้แผนภาพต้นไม้ซีกรุ๊ปแผนภาพหนึ่ง สำหรับกำหนดข้อจำกัดการใช้งานทรัพยากรในระบบย่อยได้มากกว่าหนึ่งระบบย่อย
2. ระบบไม่อนุญาติให้กำหนดให้ระบบย่อยที่มีแผนภาพต้นไม้ซีกรุ๊ปแนบอยู่แล้ว ถูกแนบเข้ากับซีกรุ๊ปที่มีระบบย่อยอื่นแนบอยู่ด้วย
3. เมื่อเริ่มต้นทาสค์จะเป็นสมาชิกของรูทโหนดของแผนภาพต้นไม้ของซีกรุ๊ป ทาสค์จะเป็นสมาชิกของซีกรุ๊ปเพียงโหนดเดียวในแผนภาพต้นไม้เดียวกัน
4. เมื่อทาสค์ในซีกรุ๊ปสร้างทาวค์ลูก ทาวค์ลูกจะได้รับมรดกและอยู่ในซีกรุ๊ปเดียวกันกับพาเรนท์

แผนภาพต้นไม้ของซีพียูและหน่วยความจำ

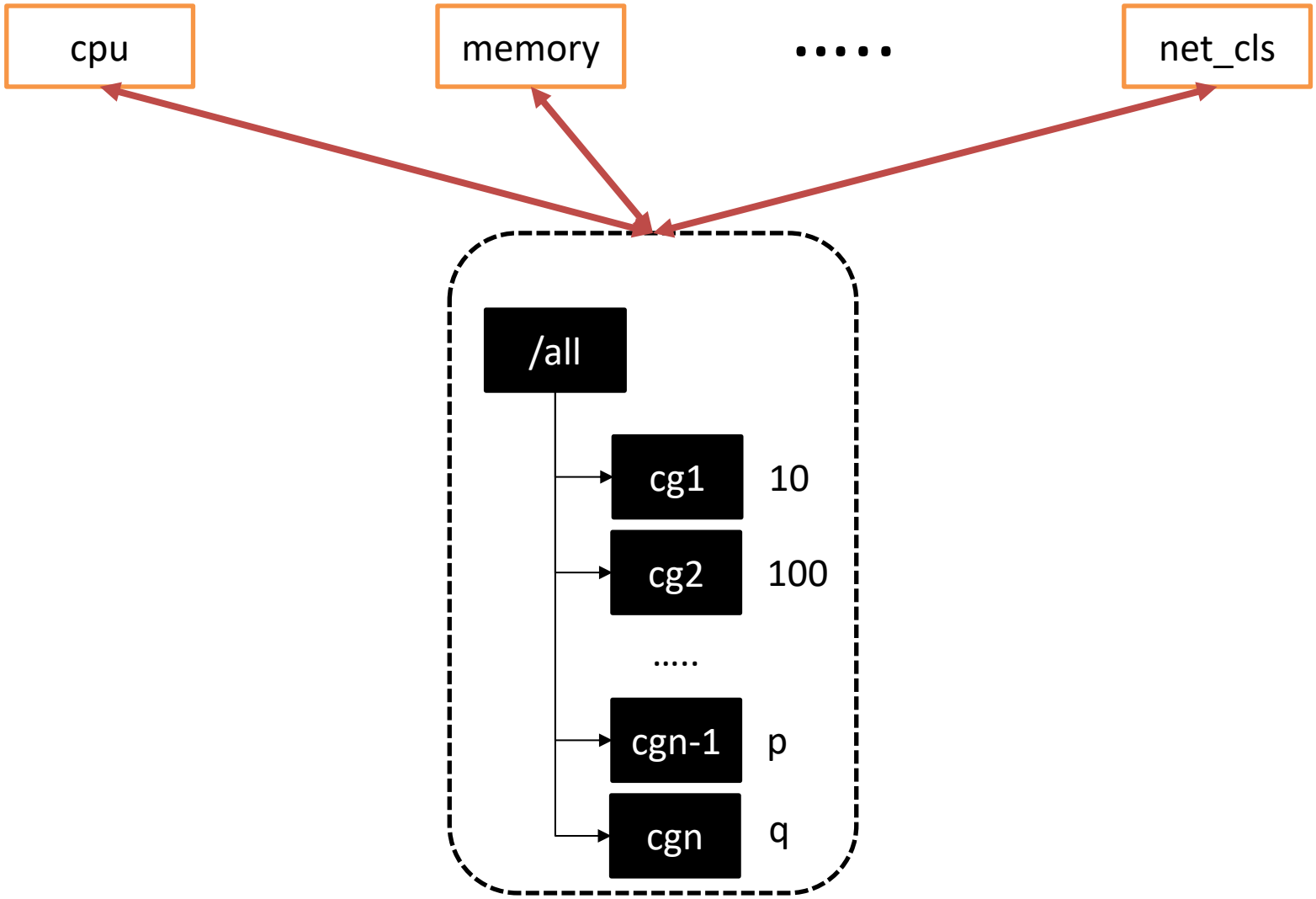


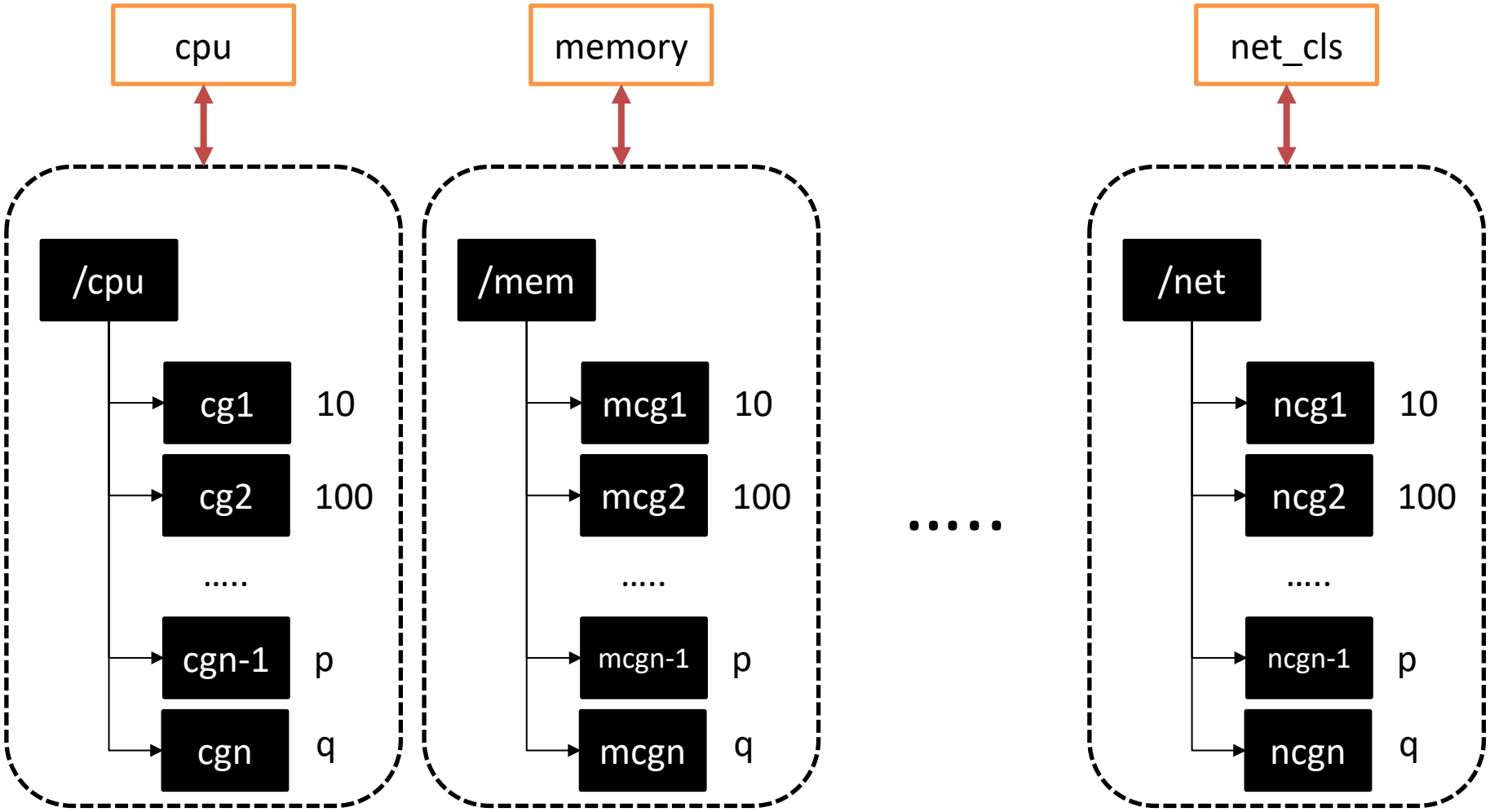

CPU
Memory
/cpu
batch
bitcoin
112, 113, 10
realtime
apache
11
redis
21, 50, 101
/mem
112, 113,
10, 11
database
21, 50, 101


CPU
Memory
net_cls
/cpu_mem_cg
cg1
112, 113, 10
cg2
21, 50, 101
102
/net_cg
cg3
112, 113, 10
21, 50, 101
102

ผลจากกฎของความสัมพันธ์

- สำหรับแผนภาพต้นไม้ของซีกรุ๊ปที่แนบเข้ากับระบบย่อยใดๆ โพรเซสหนึ่ง จะอยู่ภายใต้ข้อจำกัดของซีกรุ๊ปเดียวในแผนภาพต้นไม้นั้นเท่านั้น
- เราสามารถแนบแผนภาพต้นไม้เดียวเข้ากับระบบย่อยทุกระบบได้
- บางครั้งเราอาจจำเป็นต้องสร้างแผนภาพต้นไม้ใหม่ และแยกระบบย่อยบางระบบที่แนบกับโครงสร้างต้นไม้เดิมออกไป แล้วแนบโครงสร้างต้นไม้ใหม่เข้ากับระบบย่อยที่แยกออกไปนั้น
- ในกรณีที่ไม่จำเป็นต้องแยกระบบย่อย เราสามารถนำระบบย่อยมาแนบกับโครงสร้างต้นไม้เดิม
- เราสามารถสร้างซีกรุ๊ปเฉพาะเจาะจงสำหรับทาสค์กลุมหนึ่งได้ โดยสร้างโครงสร้างต้นไม้ที่มีซีกรุ๊ปโหนดที่มีข้อกำหนดเฉพาะสำหรับทาสค์กลุ่มนั้น
- เราสามารถกำหนดให้มีโครงสร้างต้นไม้ที่ทุกโหนดมีโพรเซสเดียวและสร้างแผนภาพต้นไม้แบบเดียวกันสำหรับทุกระบบย่อยได้ ในกรณีนี้เราสามารถกำหนดข้อจำกัดของเฉพาะของโปรเซสทุกโพรเซสโดยแยกเฉพาะสำหรับระบบย่อยแต่ละระบบย่อยได้


cpu
memory
.....
net_cls
/all
cg1
10
cg2
100
.....
cgn-1
p
cgn
q


cpu
memory
net_cls
/cpu
cg1
10
cg2
100
.....
cgn-1
p
cgn
q
/mem
mcg1
10
mcg2
100
.....
mcgn-1
p
mcgn
q
.....
/net
ncg1
10
ncg2
100
.....
ncgn-1
p
ncgn
q

การจัดการซีกรุ๊ปของหน่วยความจำจริง

- เคอร์เนิลจะนับและจำกัดจำนวนเมมโมรีเพจจริงของเครื่องที่ถูกใช้ในแต่ละกรุ๊ป
- เคอร์เนิลจะบันทึกเก็บบัญชีว่าเพจไหนอยู่ในกรุ๊ปไหน โดยแบ่งชนิดของเพจในสองมิติ มิติแรกคือการแบ่งชนิดของเพจตามลักษณะของข้อมูล
  - เพจจากไฟล์ (file) เป็นเพจที่เกิดจากการโหลดจากไฟล์ด้วย read หรือ mmap อาจเป็นข้อมูลหรือโค๊ด จากอุปกรณ์บล็อก เช่น hdd sdd
  - เพจไม่มีชื่อ (Anonymous) เป็นเพจที่ไม่มีเนื้อหาเกี่ยวข้องกับไฟล์เช่นเพจใน heap และ stack segment เป็นต้น
- มิติที่สองคือการใช้งาน
  - แอคทีฟ คือเพจที่ถูกอ้างอิงบ่อย และ
  - อินแอคทีฟเมมโมรี่คือเพจทีไม่ค่อยถูกใช้งาน

การนับเพจของซีกรุ๊ปของหน่วยความจำ

1. เพจจริงหนึ่งๆ (a physical memory page) จะถูกนับว่าเป็นของซีกรุ๊ปอินสแตนซ์อันใดอันหนึ่งเท่านั้น
2. ถ้าโพรเซสหนึ่งจบการประมวลผล และเพจที่โพรเซสเคยใช้งานถูกใช้โดยโพรเซสจากกรุ๊ปอื่น เพจนั้นจะถูกนับว่าเป็นเพจของกรุ๊ปใหม่
3. ในกรณีที่โพรเซสหลายโพรเซสแชร์ข้อมูลจากไฟล์เดียวกัน (เช่นแชร์ไลบรารี่) โพรเซสเหล่านั้นอาจมาจากหลายกรุ๊ป เพจจึงอาจถูกแชร์โดยหลายกรุ๊ปได้ ในกรณีนี้ระบบจะนับว่าเพจที่ถูกแชร์เป็นของกรุ๊ปใดกรุ๊ปหนึ่งเท่านั้น (เงื่อนไขนี้มำให้ข้อ 1 เป็นจริง)

การจำกัดการใช้งานหน่วยความจำในซีกรุ๊ป

- ผู้ใช้สามารถกำหนดให้ซีกรุ๊ปแต่ละกรุ๊ปมีข้อจำกัด (limit) ในการใช้งานหน่วยความจำ
- ผู้ใช้สามารถเลือกที่จะกำหนดข้อจำกัดหรือไม่ก็ได้ โดยดีฟอลต์จะไม่กำหนด ดังนั้นโดยดีฟอลต์โพรเซสในกลุ่มใดๆสามารถใช้หน่วยความจำได้ตามที่ต้องการ
- ในกรณีที่ผู้ใช้ต้องการกำหนดข้อจำกัด สามารถทำได้สองแบบคือ กำหนด ฮาร์ดลิมิต (hard limit) และซอฟต์ลิมิต (soft limit)

การจำกัดการใช้งานหน่วยความจำในซีกรุ๊ป

- ฮาร์ดลิมิต (hard limit) คือการกำหนดค่าขนาดของหน่วยความจำจริงที่ถ้าการใช้งานหน่วยความจำจริงของโพรเซสในซีกรุ๊ปเกินขนาดนี้แล้ว เคอร์เนิลจะสั่งให้ระบบจัดการการใช้งานหน่วยความจำเกินขนาด (Out Of Memory Killer) หรือระบบ OOM สั่งให้โพรเซสจำนวนหนึ่งในซีกรุ๊ปที่ใช้หน่วยความจำเกินจบการประมวลผล
- ซอฟต์ลิมิต (soft limit) ไม่บังคับให้โพรเซสจบการประมวลผลเหมือนฮาร์ดลิมิต แต่จะปล่อยให้โพรเซสในซีกรุ๊ปใช้หน่วยความจำเกินขนาดซอฟต์ลิมิตในกรณีที่มีโพรเซสใดจากซีกรุ๊ปที่ใช้หน่วยความจำน้อยกว่าซีกรุ๊ปนั้นต้องการใช้หน่วยความจำ เคอร์เนิลจะนำหน่ยความจำจริงของซีกรุ๊ปที่เกินซอฟต์ลิมิตมากที่สุดมให้กับโพรเซสในซีกรุ๊ปที่ใช้หน่วยความจำน้อยกว่า

การจำกัดการใช้งานหน่วยความจำในซีกรุ๊ป

- การจำกัดค่าลิมิตของการใช้งานหน่วยความจำสามารถกำหนดได้สำหรับหน่วยความจำสามแบบ ได้แก่
- หน่วยความจำจริง (physical memory) หมายถึงขนาดการใช้งานหน่วยความจำจริง
- หน่วยความจำจริงสำหรับเคอร์เนิล (kernel memory) ขนาดของหน่วยความจำเรสสิเด้นของเคอร์เนิลในหน่วยความจำจริงและขนาดหน่วยความจำที่เคอร์เนิลใช้ (เพื่อป้องกันไม่ให้เยอะเกินไปซึ่งอาจเป็นผลมาจากการโจมตีเป็นต้น)
- หน่วยความจำจริงรวม (total memory) หมายถึงขนาดการใช้งานหน่วยความจำจริง และพื้นที่ในสวอปสเปซ (swap space)

ระบบเตือนการใช้งานหน่วยความเกิน OOM Notifier

- **แทนที่จะใช้ OOM Killer สำหรับจัดการฮาร์ดลิมิต ผู้ใช้สามารถใช้ระบบเตือนการใช้หน่วยความจำเกินฮาร์ดลิมิต (OOM Notifier) แทน**
- **เมื่อเกิดการใช้งานเกินขึ้น ระบบ OOM Notifier จะทำดังนี้**
  1. **หยุดการประมวลผลของโพรเซสทั้งหมดในซีกรุ๊ป**
  2. **ส่งสัญญาณเตือนให้กับยูเซอร์โปรแกรมที่จัดการทรัพยากร**
  3. **โปรแกรมหรือผู้ใช้สามารถเลือกที่จะจบโพรเซสบางโพรเซสในซีกรุ๊ป หรือเพิ่มขนาดของลิมิตของซีกรุ๊ป หรือไมเกรตคอนเทนเนอร์ไปรันบนเครื่องอื่น**
  4. **หลังจากนั้น สั่งให้โพรเซสในซีกรุ๊ปทั้งหมดประมวลผลต่อ**

การจัดการซีกรุ๊ปของหน่วยความจำ

- เมื่อเคอร์เนิลให้เพจขอหน่วยความจำจริงให้กับโพรเซสในซีกรุ๊ป หรือนำเพจที่โพรเซสเคยใช้ไปให้กับโพรเซสในซีกรุ๊ปอื่น เคอร์เนิลจะต้องนับจำนวนเพจและทำบัญชีของจำนวนเพจในระบบ ซึ่งทำให้มีโอเวอร์เฮด
- ผู้ใช้ไม่สามารถปิดการนับได้โดยการสั่งโพรเซส แต่ทำได้โดยการกำหนดค่า boot เท่านั้น
- ในกรณีที่เพจถูกใช้โดยโพรเซสจากหลายซีกรุ๊ป เคอร์เนิลจะนับว่าแชร์เพจนั้นเป็นของซีกรุ๊ปแรกที่ใช้ และถ้าซีกรุ๊ปแรกเลิกใช้เพจ เพจนั้นจะถูกนับว่าเป็นของซีกรุ๊ปถัดไป
- Hugh TLB Cgroup กำหนดปริมาณของ huge pages ที่โพรเซสในซีกรุ๊ปสามารถใช้ได้

ซีพียูซีกรุ๊ป

- สามารถนับเวลาการใช้งานซีพียู (โดยแอปและเคอร์เนิล) ของโพรเซสในซีกรุ๊ป
- สามารถวัดปริมาณการใช้งานซีพียูแต่ละซีพียูโดยโพรเซสในซีกรุ๊ป
- อนุญาตให้กำหนดค่าน้ำหนักว่าจะให้โพรเซสในซีกรุ๊ปใดมีสิทธิ์ใช้ซีพียูใดได้มากกว่ากัน

ซีพียูซีกรุ๊ป

- แต่ไม่สามารถจำกัดการใช้งานซีพียูของโพรเซสในซีกรุ๊ปได้
- เช่น ต้องการให้โพรเซสในซีกรุ๊ป A ใช้ซีพียูคอร์ X ได้เพียงแค่ 10% ของความสามารถของซีพียู ไม่สามารถทำได้ เนื่องจากถ้าซีพียูมีการใช้งานน้อย เครื่องอาจลดคล็อกสปีดของซีพียูลงทำให้ซีพียูทำงานช้าลง และโพรเซสในซีกรุ๊ปก็จะทำงานช้าลงตามไปด้วย
- หรือ ถ้าต้องการนับจำนวนคล็อกไซเคิล ก็ทำได้ยากเพราะคำสั่งใน ISA ของซีพียูเช่น x86 แต่ละคำสั่งอาจใช้จำนวนคล็อกไซเคิลไม่เท่ากัน และการสร้างไบนารีโค๊ดของแต่ละคอมไพเลอร์ก็ไม่เหมือนกันด้วย

ซีพียูเซ็ต ซีกรุ๊ป

- CPUset Cgroups อนุญาตให้ pin โพรเซสในซีกรุ๊ปบนซีพียู หรือเซ็ตของซีพียู เพื่อให้รันบนซียูที่ pin เท่านั้น
- กำหนดเซ็ตของซีพียูสำหรับแอปใดแอปหนึ่งได้
- เพื่อให้เข้าถึงข้อมูลได้เร็วในกรณีของ NUMA หรือ
- ป้องกันการไมเกรตข้ามซอกเกตเป็นต้น

บล็อกไอโอ ซีกรุ๊ป (Blkio Cgroup)

- เรียกอีกอย่างว่า Block IO Controller
- เป็นการเก็บบัญชีการใช้งานบล็อกไอโอสำหรับแต่ละซีกรุ๊ปว่าแต่ละกรุ๊ป
  - ใช้งาน อุปกรณ์บล็อกดีไวซ์ (block device) หนึ่งไปเท่าไร
  - อ่านข้อมูลและเขียนข้อมูล กับบล็อกดีไวซ์ไปเท่าไร
  - อ่านเขียนข้อมูลแบบ ซิงโครนัส หรือ อะซิงโครนัส ไปเท่าไร
- สามารถ ทร็อทเทอริ่ง (throttling) (หมายถึงการจำกัดแบนวิดธ์หรือจำกัดอัตราการปฏิบิติงาน) สำหรับแต่ละซีกรุ๊ป
  - ให้มีอัตราการใช้งานบล็อกดีไวซ์หนึ่งๆเท่าไร
  - กำหนดอัตราการอ่านและเขียนข้อมูล (เป็น OPS หรือ Bytes)

บล็อกไอโอ ซีกรุ๊ป (Blkio Cgroup)

- สามารถกำหนดให้ซีกรุ๊ปมีน้ำหนักในการใช้งานอุปกรณ์บล็อกดีไวซ์ที่แตกต่างกันได้
- การกำหนดอัตราการเขียนข้อมูลลงสู่บล็อกดีไวซ์ มีสิ่งที่ต้องระลึกถึงเมื่อผู้ใช้อ่านค่าประสิทธิภาพ
  - เช่น ผู้ใช้อาจจะจำกัดค่าแบนวิธสำหรับเขียนข้อมูลลงในอุปกรณ์บล็อกไอโอระดับหนึ่ง แต่ค่าที่วัดได้อาจสูงกว่าที่กำหนดไว้มาก ทั้งนี้เพราะโอเออาจเก็บข้อมูลไว้ในหน่วยความจำที่ใช้เป็นหน่วยความจำแคชก่อนแล้วจึงทยอยเขียนลงสู่หน่วยเก็บข้อมูล
  - การเขียนข้อมูลสู่แคชจะทำให้เห็นประสิทธิภาพสูงและไม่ถูกจำกัดตามที่กำหนด แต่การเขียนข้อมูลลงสู่บล็อกดีไวซ์จะเป็นไปตามข้อจำกัดที่กำหนดแต่ผู้ใช้จะไม่เห็น

เน็ต_ซีแอลเอส ซีกรุ๊ป (net_cls Cgroup)

- อนุญาตให้ผู้ใช้กำหนดให้ เน็ตเวิร์คแพคเกต (ข้อมูลส่งออก) ที่สร้างโดยโพรเซสในซีกรุ๊ป มีแท็กเป็นค่า classid
- ผู้ใช้สามารถใช้ tc utility (traffic control) เพื่อควบคุมแบนด์วิธของข้อมูลที่มีค่าแท็กที่สนใจ
  - tc class add dev eth0 parent 10: classid 10:1 htb rate 40mbit
- ผู้ใช้สามารถใช้ iptable เพื่อจัดการแพคเกตที่มีค่าแท็กที่สนใจ
  - iptables -A OUTPUT -m cgroup ! --cgroup 0x100001 -j DROP

เน็ต_ไพรออร์ ซีกรุ๊ป (net_prio Cgroup)

- อนุญาตให้แอดมินกำหนดค่าไพออริตี้ (priority) ของเน็ตเวิร์คทราฟฟิคที่ส่งออกทาง เน็ตเวิร์คอินเตอร์เฟส ให้โพรเซสจากซีกรุ๊ปที่ต่างกันมีค่าไพออริตี้สำหรับการส่งออก การเก็บในคิว หรือการยกเลิก ที่ต่างกัน
- การกำหนดค่าไพออริตี้ของเน็ตเวิร์คทราฟฟิคสามารถทำได้โดยการกำหนดค่า ซอกเก็ตออปชั่นในแอพพลิเคชั่น แต่เรามักจะกำหนดค่าไพออริตี้ตอนนำแอพมาใช้งานมากกว่า
- แอดมินสามารถใช้ tc utility กำหนดค่าไพออริตี้ของเน็ตเวิร์คทราฟฟิค ได้เช่นกัน
- echo "eth0 5" > /cgroup/net_prio/iscsi/net_prio.ifpriomap

ดีไวซ์ ซีกรุ๊ป (Device Cgroup)

- อนุญาตให้ผู้ใช้กำหนดได้ว่าจะให้โพรเซสในซีกรุ๊ปสามารถ อ่าน เขียน หรือ สร้างไฟล์ device ได้
- ผู้ใช้สามารถกำหนดให้โพรเซสในคอนเทนเนอร์ สามารถอ่านเขียนดีไวซ์ที่จะไม่ทำให้เกิดอันตรายต่อทั้งระบบเช่น /dev/tty /dev/random /dev/null และปิดการเข้าถึงอุปกรณ์อื่นที่เหลือ
- ผู้ใช้สามารถกำหนดให้โพรเซสในซีกรุ๊ปมองเห็นไฟล์อุปกรณ์ที่สนใจ
  - /dev/kvm : ถ้าต้องการรัน qemu-kvm ในซีกรุ๊ป (หรือคอนเทนเนอร์)
  - /dev/net/tun : โพรเซสในซีกรุ๊ปเห็นอุปกรณ์ tun ของวีพีเอ็น
  - /dev/fuse : โพรเซสในซีกรุ๊ปสามารถสร้างไฟล์ซิสเต็มในยูเซอร์สแปซ
  - /dev/dri : โพรเซสในซีกรุ๊ปสามารถใช้ จีพียู

ฟรีซเซอร์ ซีกรุ๊ป (Freezer Cgroup)

- อนุญาตให้ผู้ใช้หยุดการประมวลผลของโพรเซสทั้งหมดในซีกรุ๊ป และรีซูมการประมวลผล
- คล้ายการส่ง SIGSTOP ซิกแนลไปหยุดโพรเซสเดี่ยวและ SIGCONT ซิกแนลไปรีซูมโพรเซส
- สามารถใช้ได้กับการ scheduling โพรเซสทั้งซีกรุ๊ป
- ในกรณีที่มีทั้ง client และ server process สามารถหยุดได้ทั้งสองพร้อมๆ กัน เพื่อที่ไม่ให้โพรเซสหนึ่งสังเกตุเห็นว่าอีกโพรเซสหนึ่งหยุดและส่งผลกระทบกับโลจิคการประมวลผลของโปรแกรม
- สามารถหยุดทั้งคอนเทนเนอร์แล้ว process migration ไปยังเครื่องอื่น หรือ process checkpointing

การกำหนดให้โพรเซสอยู่ในซีกรุ๊ป

- โพรเซส PID 1 จะอยู่ที่รูทซีกรุ๊ป ของทุกแผนภาพต้นไม้
- เมื่อสร้างโพรเซสลูก ลูกจะอยู่ในซีกรุ๊ปเดียวกันกับพ่อแม่
- มี utility และไลบรารี่สำหรับจัดการซีกรุ๊ปเช่น cgmanager
- เราสามารถใช้ psudo-fs เพื่อจัดการซีกรุ๊ปและการกำหนดให้โพรเซสอยู่ภายใต้สังกัดของซีกรุ๊ป
  - /sys/fs/cgroup
- psudo-fs ทำให้สร้าง cgroup ได้ด้วยคำสั่ง mkdir
  - mkdir /sys/fs/cgroup/memory/agroup
- กำหนดหรือให้โพรเซสอยู่ใน cgroup หรือ ย้ายโพรเซสจากซีกรุ๊ปหนึ่งไปยังอีกอันหนึ่ง ได้ด้วยคำสั่ง echo
  - echo $PID > /sys/fs/cgroup/memory/agroup/tasks

คอนเทนเนอร์

- คือการประมวลผลแบบเสมือนอย่างเบา (lightweight VM) ที่สร้างภาพเสมือนของการใช้งานคอนเทนเนอร์เหมือนกับเป็นเครื่องจริงเครื่องหนึ่ง ที่ผู้ใช้สามารถ
  - รันแอพพลิเคชั่น
  - ประมวลผลเป็นซูเปอร์ยูเซอร์
  - รันเซอร์วิส
  - ติดตั้งแพคเกจ
  - กำหนดค่าเน็ตเวิร์ค เช่นไอพีแอดเดรส
  - มีอุปกรณ์เครือข่าย (เสมือน) เป็นของตนเอง

คอนเทนเนอร์

- แต่คอนเทนเนอร์ไม่ใช่วีเอ็ม
- คอนเทนเนอร์เป็นกลุ่มของโพรเซสที่รันอยู่บนโฮสโอเอส
- คอนเทนเนอร์สร้างภาพเสมือนของการแยกกลุ่มของโพรเซสออกจากกลุ่มอื่น และแต่ละกลุ่มจะมองไม่เห็นโพรเซสในกลุ่มอื่น
- โฮสโอเอสเห็นโพรเซสในคอนเทนเนอร์ แต่โพรเซสในคอนเทนเนอร์จะเห็นเฉพาะโพรเซสที่อยู่ในคอนเทนเนอร์เดียวกันเท่านั้น
- ไม่สามารถรันต่างโอเอสจากโฮสได้ และไม่สามารถมีเคอร์เนิลโมดูลแยกจากของโฮสได้ และทุกคอนเทนเนอร์แชร์เคอร์เนิลร่วมกัน
- แต่ละคอนเทนเนอร์มีโพรเซสที่มีพีไอดี 1 แต่ไม่ใช้ init ไม่มี cron

คอนเทนเนอร์

- คอนเทนเนอร์ เป็นสิ่งทีสร้างขึ้นมาจากส่วนประกอบของเคอร์เนิลเรียกว่า ซีกรุ๊ป (Cgroup) เนมสเปซ (Namespace) และก็อปปี้ออนไร้ท์ อิมเมจ (Copy On Write) หรือ (Cow)
- ซีกรุ๊ป อนุญาตให้เราวัดและกำหนดปริมาณการใช้งานทรัพยากร เช่น หน่วยความจำ ซีพียู บล็อกไอโอ เน็ตเวิร์คไอโอ
- เนมสเปซ อนุญาตให้มีการสร้างอินสแตนซ์ของคุณสมบัติหลายอย่างของเคอร์เนิลให้มีมากกว่าหนึ่งอินสแตนซ์ และเคอร์เนิลจะแยกอินสแตนซ์เหล่านั้นออกจากกัน
- ก็อปปี้ออนไร้ท์ อิมเมจ อนุญาตให้คอนเทนเนอร์สร้างไฟล์ซิสเต็มอิมเมจของตนเอง และสามารถปรับแต่งอิมเมจตามความต้องการได้

สรุป

- แนะนำซีกรุ๊ปขั้นพื้นฐาน
- ที่เป็นเทคโนโลยีเปิดทางสำหรับสร้างคอนเทนเนอร์

เนื้อหาเรียบเรียงจาก https://www.youtube.com/watch?v=sK5i-N34im8&t=211s