목차

Oracle DB 사용을 위한 RHEL 튜닝 3
tuned profile 사용 3
수동 설정 3
/etc/sysctl.conf의 메모리 설정 3
HugePages 4
/etc/security/limits.conf의 제한 설정 5
공유 메모리 5
세마포어 6
Oracle 계정을 위한 open files 6
Red Hat Enterprise Linux 6 및 7에만 권장되는 TPP(Transparent hugepages) 비활성화 6
I / O 스케줄러 7
NUMA 밸런싱 7
참조링크 7

Oracle DB 사용을 위한 RHEL 튜닝

Oracle Database를 사용하기 위한 RHEL 튜닝방법은 3가지가 있다.

- `tuned`에서 제공하는 `tuned-profiles-oracle` 프로필 사용
- https://access.redhat.com/labs/rheltfo/ 에서 제공하는 스크립트 사용
- 수동 설정

tuned profile 사용

Oracle RDBMS를 위한 `tuned-profiles-oracle`은 기본으로 제공되지는 않는다.

- RHEL6 : `rhel-6-server-optional-rpms`
- RHEL7 : `rhel-7-server-optional-rpms`
- RHEL8 : `rhel-8-for-x86_64-baseos-rpms`

위의 채널(레포지토리)를 등록하면 사용할 수 있다.

수동 설정

> 이 내용에 언급 된 모든 수치는 일반적인 것이 아니며 Red Hat Enterprise Linux에 상주하는 데이터베이스 워크로드에 긍정적인 영향을 미치는 것으로 나타났습니다. 즉, 특정 값은 사용자 환경에 따라 다르며 추가 조정이 필요할 수 있습니다.

/etc/sysctl.conf의 메모리 설정

- Oracle 사용시 Swap을 사용하는것은 좋지 않으며 가능한 한 피해야합니다. 다음 내용은 커널이 스왑을 덜 사용하도록 합니다.

```
vm.swappiness=10
```

- `dirty page`를 가질 수있는 활성 메모리의 최대 백분율. 예를 들어 시스템에 `1000`개의 메모리 페이지가 있고 `dirty_background_ratio`가 `3%`로 설정되어 있으면 30페이지가 더러워 지면 쓰기 저장이 시작됩니다 .

```
vm.dirty_background_ratio=3
```

- `dirty page`를 가질 수있는 총 메모리의 최대 백분율. `1000`개의 페이지가 있는 시스템 에서 `40%`로 설정되어 있으면 `400`번째 페이지가 더러워질때까지 대기합니다. 이 메커니즘은 시스템이 더티 페이지를 확인하는 속도를 느리게 합니다.

```
vm.dirty_ratio=40
```

- 만료되기 전에 데이터가 페이지 캐시에있을 수있는 기간 (1/100초):

```
vm.dirty_expire_centisecs=500
```

- 더티 페이지를 정리하기 위해 pdflush가 활성화되는 빈도 (1/100초) :

```
vm.dirty_writeback_centisecs=100
```

HugePages

- /proc/meminfo 에서 Hugepagesize 얻기

```
$ grep Huge /proc/meminfo
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
Hugepagesize:       2048 kB
```

- Oracle의 Hugepagesize 및 SGA (System Global Area) 값을 사용하여 아래 공식을 사용하여 권장되는 HugePages 수를 계산합니다. SGA에 대한 자세한 내용과 해당 값을 찾을 수있는 위치는 Oracle 설명서를 참조하십시오. PGA (Program Global Area)는 Hugepages를 사용하지 않습니다.

```
SGA / huge page size
```

다음은 20GB SGA에 맞추기 위해 필요한 10240 Hugepages의 양을 보여주는 예 입니다. 값은 계산을 위해 모두 킬로바이트로 변환되었습니다.

```
Hugepagesize = 2048 KB
20 GB SGA = 20971520 KB

(20971520 / 2048) = 10240
```

- /etc/sysctl.conf설정에 해당 번호를 추가 합니다.

```
vm.nr_hugepages=10240
vm.hugetlb_shm_group=<insert oracle group ID here>
```

- 위의 설정이 테스트되고 필요한 성능을 제공하는 것으로 확인되면 HugePage 할당을 grub 구성

의 커널 줄로 부팅 옵션으로 이동하는 것이 좋습니다. 이를 통해 HugePages를 더 빠르고 깔끔하게 할당 할 수 있습니다.

```
kernel /vmlinuz-<kernel-version> ro root=/dev/vg01/lv01 hugepages=10240
<other boot options>
```

- Hugepages가 커널 명령 줄에 설정되어 있지 않은 경우에서 nr_hugepages값을 설정하여 동적으로 할당 할 수 있습니다.
- 시스템에 메모리 부족이 발생하면 대용량 페이지를 교체 할 수 없습니다.
- Huge Pages를 사용하도록 Oracle 데이터베이스 구성을 변경하십시오. 도움이 필요한 경우 Oracle 지원에 문의하십시오.

/etc/security/limits.conf의 제한 설정

Oracle 데이터베이스가 Red Hat Enterprise Linux에서 Huge Pages를 사용하려면 /etc/security/limits.conf에서 oracle 사용자에 대한 ulimit 매개 변수 memlock을 늘려야 할 수도 있습니다. memlock 설정은 KB 단위로 지정되며 Oracle이 할당 할 수있는 Huge Pages 수의 메모리 크기와 일치해야합니다. 따라서 Oracle 데이터베이스가 512 개의 Huge Pages를 사용할 수 있어야한다면 memlock을 최소한 (512 * Hugepagesize)로 설정해야합니다. 기본 시스템에서는 1048576KB (512 * 2048)가 됩니다.

```
oracle     soft     memlock     1048576
oracle     hard     memlock     1048576
```

공유 메모리

- SHMALL은 시스템 전체에서 사용할 수있는 공유 메모리 페이지의 최대 총량입니다. 따라서 SHMALL는 항상 최소 ceil(SHMMAX/PAGE_SIZE)보다 커야 합니다.
- SHMMAX은 만들 수있는 단일 공유 메모리 세그먼트의 최대 크기 (바이트)입니다.
- 시스템의 총 메모리 (바이트)를 얻습니다.

```
mem=$(free -b | awk '/Mem/ {print $2}')
```

- 페이지 크기 (바이트) 가져 오기 :

```
page=$(getconf PAGE_SIZE)
```

- 다음에 SHMALL대한 페이지에서 시스템 총 메모리의 75%를 계산합니다.

```
all=$(expr $mem \* 75 / 100 / $page + 1)
```

- SHMALL값에 페이지 크기를 곱하면 다음 SHMMAX을 얻을 수 있습니다.

```
max=$(expr $all \* $page)
```

- /etc/sysctl.conf파일 에서 SHMMAX및 SHMALL값을 설정 하십시오.

```
echo "kernel.shmmax = $max" >> /etc/sysctl.conf
echo "kernel.shmall = $all" >> /etc/sysctl.conf
```

- /etc/sysctl.conf파일 에서 SHMMNI와 공유 메모리 세그먼트의 최대 수를 설정 하십시오.

```
echo "kernel.shmmni = 4096" >> /etc/sysctl.conf
```

세마포어

- 세마포어 작업에 권장되는 최소값 :

```
kernel.sem="250 32000 100 128"
```

- 첫 번째 값인 SEMMSL은 세마포어 세트당 최대 세마포어 수입니다.
- 두 번째 값인 SEMMNS는 시스템의 총 세마포 수를 정의합니다.
- 세 번째 값인 SEMOPM은 세마포어 호출 당 최대 세마포어 작업 수를 정의합니다.
- 마지막 값인 SEMMNI는 시스템의 전체 세마포어 세트 수를 정의합니다.

```
# sysctl -w "kernel.sem = 250 32000 100 128"
# echo "kernel.sem = 250 32000 100 128" >> /etc/sysctl.conf
```

Oracle 계정을 위한 open files

```
    # vi /etc/security/limits.conf

    #<domain>      <type>  <item>         <value>
    oracle          hard    nofile          10000
```

Red Hat Enterprise Linux 6 및 7에만 권장되는 TPP(Transparent hugepages) 비활성화

- tuned가 설치된 경우 tuned 및 ktune서비스 비활성화

```
# service tuned stop
# chkconfig tuned off
# service ktune stop
# chkconfig ktune off
```

```
# tuned-adm off
```

> 이 tuned-adm명령은 모든 설정을 조정 시작 전의 설정으로 되돌리고 부팅시 조정 서비스가 실행되지 않도록합니다. 대안으로, 사용자 정의 된 조정 된 프로필을 생성하여 이전 설정을 대신하고 THP 만 비활성화 할 수 있습니다. Red Hat Enterprise Linux 6 에서 TPP (Transparent hugepages) 비활성화가 적용되지 않음을 참조하십시오.

- /boot/grub/grub.conf 파일의 커널 명령 줄에 transparent_hugepage=never추가 합니다.

```
kernel /boot/vmlinuz-2.6.32-358.el6.x86_64 ro
root=UUID=a216d1e5-884f-4e5c-859a-6e2e2530d486 rhgb quiet
transparent_hugepage=never
```

- 변경 사항을 적용하려면 서버를 재부팅하십시오.

```
# reboot
```

I / O 스케줄러

기본 CFQ I/O 스케줄러는 대부분의 워크로드에 적합하지만 데이터베이스 환경에 최적의 성능을 제공하지 않습니다. 대신 다음 IO 스케줄러 중 하나를 사용해야합니다.

- deadline 스케줄러는 물리적 스토리지에 권장
- noop 스케줄러는 가상 스토리지에 권장

NUMA 밸런싱

- 자동 NUMA 밸런싱
- RHEL7에서는 기본적으로 numa_balancing이 활성화됩니다. 이것은 Oracle 워크로드에 대한 영향을 볼 수 있으므로 모범 사례는 NUMA 균형 조정을 방지하는 것입니다.
- /etc/sysctl.conf에서 kernel.numa_balancing = 0을 구성하고 sysctl -p를 실행하십시오.

참조링크

- https://access.redhat.com/solutions/39188

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/oracle_db_%EC%82%AC%EC%9A%A9%EC%9D%84_%EC%9C%84%ED%95%9C_rhel_%ED%8A%9C%EB%8B%9D

Last update: **2021/04/08 00:52**