
Article

[Mark Bolinsky](#) · Jan 29, 2016 4m read

Linux Transparent HugePages and the impact to InterSystems IRIS

** Revised Feb-12, 2018

While this article is about InterSystems IRIS, it also applies to Caché, Ensemble, and HealthShare distributions.

Introduction

Memory is managed in pages. The default page size is 4KB on Linux systems. Red Hat Enterprise Linux 6, SUSE Linux Enterprise Server 11, and Oracle Linux 6 introduced a method to provide an increased page size in 2MB or 1GB sizes depending on system configuration know as HugePages.

At first HugePages required to be assigned at boot time, and if not managed or calculated appropriately could result in wasted resources. As a result various Linux distributions introduced Transparent HugePages with the 2.6.38 kernel as enabled by default. This was meant as a means to automate creating, managing, and using HugePages. Prior kernel versions may have this feature as well however may not be marked as [always] and potentially set to [madvise].

Transparent Huge Pages (THP) is a Linux memory management system that reduces the overhead of Translation Lookaside Buffer (TLB) lookups on machines with large amounts of memory by using larger memory pages. However in current Linux releases THP can only map individual process heap and stack space.

The Problem

The majority of memory allocation in any Cache' system is the shared memory segments (global and routine buffers pools) and because THP does not handle these shared memory segments. As a result THP are not used for shared memory, and are only used for each individual process. This can be confirmed using a simple shell command.

The following is an example from a test system at InterSystems which shows 2MB THP allocated to Cache' processes:

```
# grep -e AnonHugePages /proc/*/smaps | awk '{ if($2>4) print $0} ' | awk -F "/" '{print $0; system("ps -fp " $3)} '
```

```
/proc/2945/smaps:AnonHugePages:          2048 kB
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	2945	1	0	2015	?	01:35:41	/usr/sbin/rsyslogd -n

```
/proc/70937/smaps:AnonHugePages:        2048 kB
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70937	70897	0	Jan27	pts/0	00:01:58	/bench/EJR/ycsb161b641/bin/cache WD

/proc/70938/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70938	70897	0	Jan27	pts/0	00:00:00	/bench/EJR/ycsb161b641/bin/cache GC

/proc/70939/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70939	70897	0	Jan27	pts/0	00:00:39	/bench/EJR/ycsb161b641/bin/cache JD

/proc/70939/smmaps:AnonHugePages: 4096 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70939	70897	0	Jan27	pts/0	00:00:39	/bench/EJR/ycsb161b641/bin/cache JD

/proc/70940/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70940	70897	0	Jan27	pts/0	00:00:29	/bench/EJR/ycsb161b641/bin/cache SW
D 1							

/proc/70941/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70941	70897	0	Jan27	pts/0	00:00:29	/bench/EJR/ycsb161b641/bin/cache SW
D 2							

/proc/70942/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70942	70897	0	Jan27	pts/0	00:00:29	/bench/EJR/ycsb161b641/bin/cache SW
D 3							

/proc/70943/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70943	70897	0	Jan27	pts/0	00:00:33	/bench/EJR/ycsb161b641/bin/cache SW
D 7							

/proc/70944/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70944	70897	0	Jan27	pts/0	00:00:29	/bench/EJR/ycsb161b641/bin/cache SW
D 4							

/proc/70945/smmaps:AnonHugePages: 2048 kB

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70945	70897	0	Jan27	pts/0	00:00:30	/bench/EJR/ycsb161b641/bin/cache SW
D 5							

```
/proc/70946/smaps:AnonHugePages:      2048 kB
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	70946	70897	0	Jan27	pts/0	00:00:30	/bench/EJR/yicsbl61b641/bin/cache SW D 6

```
/proc/70947/smaps:AnonHugePages:      4096 kB
```

In addition, there are potential performance penalties in the form of memory allocation delays at runtime especially for applications that may have a high rate of job or process creation.

The Recommendation

InterSystems recommends for the time being to disable THP as the intended performance gain is not applicable to IRIS shared memory segment, and the potential for a negative performance impact in some applications.

Check to see if your Linux system has Transparent HugePages enabled by running of the following commands:

For Red Hat Enterprise Linux kernels:

```
# cat /sys/kernel/mm/redhattransparenthugepage/enabled
```

For other kernels:

```
# cat /sys/kernel/mm/transparenthugepage/enabled
```

The above command will display whether the [always], [madvise], or [never] flag is enabled. If THP is removed from the kernel then the `/sys/kernel/mm/redhattransparenthugepage` or `/sys/kernel/mm/redhat/transparenthugepage` files do not exist.

To disable Transparent HugePages during boot perform the following steps:

1. Add the following entry to the kernel boot line in the `/etc/grub.conf` file:

```
transparenthugepage=never
```

2. Reboot the operating system

There is a method to also disable THP on-the-fly, however this may not provide the desired result as that method will only stop the creation and usage of THP for new processes. THP already created will not be disassembled into regular memory pages. It is advised to completely reboot the system to have THP disabled at boot time.

*Note: It is recommended to confirm with your respective Linux distributor to confirm the methods used for disabling THP.

[#Best Practices](#) [#InterSystems Business Solutions and Architectures](#) [#System Administration](#) [#Cache](#)
[#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)