|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  | | |
|  | | |
| 方圆ok1 | | | |
|  | **2017年闰秒对FusionInsight的影响评估** | | 附件1-16K |
|  | |
| **文档版本** | **01** |
| **发布尔日期** | **2015-06-10** |
|  | |
| 华为技术有限公司 | |
|  | | |

|  |
| --- |
| 版权所有 © 华为技术有限公司 2015。 保留一切权利。  非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。  商标声明  附件3-版权声明页图和其他华为商标均为华为技术有限公司的商标。  本文档提及的其他所有商标或注册商标，由各自的所有人拥有。  注意  您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |
| --- | --- |
| 华为技术有限公司 | |
| 地址： | 深圳市龙岗区坂田华为总部办公楼 邮编：518129 |
| 网址： | [http://www.huawei.com](http://www.huawei.com/) |
| 客户服务邮箱： | [support@huawei.com](mailto:Support@huawei.com) |
| 客户服务电话： | 4008302118 |

目 录

[1 背景 4](#_Toc423103885)

[1.1 目的 4](#_Toc423103886)

[1.2 背景 4](#_Toc423103887)

[2 SuSE与闰秒 5](#_Toc423103888)

[2.1 配置且运行NTP的系统 5](#_Toc423103889)

[2.2 没配置使用NTP的系统 5](#_Toc423103890)

[2.3 Linux内核Bug与闰秒的关系 6](#_Toc423103891)

[2.4 Linux内核受闰秒影响的情况 6](#_Toc423103892)

[2.5 Workaround 8](#_Toc423103893)

[3 RedHat与闰秒 10](#_Toc423103894)

[3.1 综述 10](#_Toc423103895)

[3.2 受影响的版本 11](#_Toc423103896)

[3.3 解决方案 11](#_Toc423103897)

[4 闰秒对FusionInsight的影响 12](#_Toc423103898)

[4.1 当前状态 12](#_Toc423103899)

[4.2 影响 12](#_Toc423103900)

[4.3 结论 13](#_Toc423103901)

# 背景

## 目的

本手册主要阐述2017年闰秒对FusionInsight产品的影响以及解决办法。

## 背景

2017年1月1日0时将出现闰秒，对计算机系统正常运行造成影响。

由于SuSE11.X版本的操作系统内核存在Bug，导致了配置了ntp时钟同步的SUSE11操作系统在出现了闰秒的时候将有可能出现deadlock死锁的情况，可能导致操作系统夯或CPU使用近100%

# SuSE与闰秒

SuSE对于闰秒的处理，应该分为以下两种情况：

## 配置且运行NTP的系统

在任何SuSE Linux 环境上，如果正确配置NTP，作为NTP客户端，使用NTP守护进程来同步本地的时间与NTP服务器的情况，SuSE Linux的内核会自动考虑到闰秒所带来的影响。在闰秒修正前的最后一天，NTP服务器会通知ntp客户端，在时间23:59:59 UTC将会有一个闰秒发生，内核会对这额外的一秒进行删除操作，并且在内核打印如下log：

"Clock: inserting leap second 23: 59: 60 UTC".

内核对闰秒进行正常处理之后，系统的计算时钟应该如下，内核将会出现两次59秒。

## 没配置使用NTP的系统

默认情况，如果没有配置NTP，并且使用NTP进行时间同步的系统，是不会自动修正闰秒的误差的，这将会造成系统时间比UTC时间快一秒。（注： 对于使用硬件时钟的系统，由于硬件时钟的准确性差，所以是否会修正闰秒，没有实际的意义。）

对于这种系统，如果你期望也能正常修正系统的闰秒，可以采用以下方案：将系统的timezone更新到包括2015闰秒处理的最新版本。

首先，复制/usr/share/zoneinfo/right目录下相应的文件到/etc/localtime ，然后，重新设定时钟到正确的本地时间。 /usr/share/zoneinfo/right目录下包含所有自大纪元对1970-01-01 00:00:00 UTC 开始以来发生的闰秒纠正本地时间信息。而/usr/share/zoneinfo目录下他时区文件没有闰秒修正补充。

举个例子，如果一个系统是在美国/ Los\_Angeles（美国太平洋）时区，你可以重新配置系统，通过运行以下命令，重新设定时钟太平洋时间报告闰秒校正时间：

cp /usr/share/zoneinfo/right/America/Los\_Angeles /etc/localtime

注意：/usr/share/zoneinfo/right下的时区包，不要与NTP服务同时使用。

## Linux内核Bug与闰秒的关系

首先，必须明确一个概念，闰秒跟内核的bug没有直接必然的联系。 当闰秒的发生，内核需要对闰秒进行处理，而某些SLES版本（SLES11）内核进行处理的代码部分存在了Bug, 从而造成当闰秒发生的时候，有可能造成系统的挂死。

其次,存在内核Bug的SLES版本的触发条件是：

1. 服务器正确配置ntp，并且作为客户端跟ntp服务器同步时间

2. ntp的工作模式采用了adjtimex同步时钟的方式。

所以采用zoneinfo 时区包更新闰秒的方式，不会触发任何内核bug，或者ntp的工作模式为“slew”,也不会触发内核bug。

## Linux内核受闰秒影响的情况

SLES10 和SLES9都不受闰秒的影响，闰秒不会触发SLES10和SLES9的以下相关的bug，SLES11受影响情况如下，并且PTF版本已经可以下载。

bnc#767684（printk）

SLES 11

SP1 - kernel 2.6.32.59-0.7和更新版本不受影响

SP2 - kernel 3.0.38-0.5 和更新版本不受影响

SP3 - 不受影响

bnc#768632 and bnc#769841

timer deadlock caused by do\_adjtimex()

hang in ktime\_get+54

SLES 11

SP1 - kernel 2.6.32.59-0.7 和更新版本不受影响

SP2 - kernel 3.0.38-0.5 和更新版本不受影响

SP3 - 不受影响

bnc#771619

100% CPU load in applications calling futex with a timeout shorterthan one second

SLES11

SP1 - kernel 2.6.32.59-0.7 和更新版本不受影响

SP2 - kernel 3.0.38-0.5 和更新版本不受影响

SP3 - 不受影响

bsc#915335

Deadlock after fix 771619

SLES11 – 都受影响，下一个内核版本升级修复

SP1 - kernel 2.6.32.59-0.19.1 和更新版本不受影响

SP2 - kernel 3.0.101-0.7.29.1 和更新版本不受影响

SP3 - kernel 3.0.101-0.47.52.1 和更新版本不受影响

## Workaround

如果期望不通过升级内核的方式，修复这些Bug，可以采用以下

workaround：

一．将在发生闰秒前24小时 将NTP的模式更改为“slew”方式，然后等待闰秒发生过去之后，July 1日将NTP的工作模式修改为正常方式。修改为“slew”模式方法如下

1. 编辑/etc/sysconfig/ntp， 添加”-x” 到NTPD\_OPTIONS，结果如下:

NTPD\_OPTIONS="-x -g -u ntp: ntp"

2. 重启NTP服务

rcntp restart

请确保在23:59:59 UTC 06/29/2015日前做以上操作

技术原理：

改变NTP 服务的“slew”模式下运行，这将避免使用adjtime（）的系统调用。 因为系统调用带来TIME\_INS状态的内核，它会在UTC 23:59:59插入闰秒并且打印通知。 printk拥有xtime\_lock锁的情况下被调用，在非常罕见的情况下，这就造成死锁从而造成非常高的系统负载。 对adjtime（）的系统调用可以在之前的闰秒发生前24小时更新， 所以需要在发生闰秒24小时以前更改NTP工作模式。 如果等闰秒过去之后NTP可以安全改回来了。

2．由于受bnc#771619 的影响，有可能造成使用FUTEX 的应用的CPU占有率100%。 对于这个问题的workaround，运行以下命令，更新系统的闰秒影响。（如果升级第三章的对于内核后，不会出现这个问题，推荐升级内核）

date -s "$(LC\_ALL=C date) "

技术原理

因为闰秒发生后，系统本该调用clock\_was\_set，但是bug的内核没用调用，而设置日期/时间将触发clock\_was\_set（）系统调用。 如果没用调用clock\_was\_set，内核继续运行并且每个CPU hrtimer base不与实时运行同步。 开始一个新的进程并没有解决这个问题。 如果一个新的进程调用 absolute hrtimer，它会等待比正常少等待一秒钟。 如果timer调度少于当前的real time不到一秒，timer将立即触发。

# RedHat与闰秒

## 综述

如果ntp客户端使用到slew（渐变）模式，ntp有一个bug会导致时间回退。以下是实验的结果：

环境：

ntp server: ntp & tzdata RHEL 6

ntp client: ntp & tzdata RHEL 6 in slew mode

2015-06-30T23:59:59,793932397+0000

2015-06-30T23:59:59,896986650+0000

2015-07-01T00:00:00,000057570+0000

2015-07-01T00:00:00,103566746+0000

2015-07-01T00:00:00,207228003+0000

2015-07-01T00:00:00,310636390+0000

2015-07-01T00:00:00,414048402+0000

2015-07-01T00:00:00,517227837+0000

2015-07-01T00:00:00,620638190+0000

2015-06-30T23:59:59,723691422+0000

2015-06-30T23:59:59,826933854+0000

2015-06-30T23:59:59,930085780+0000

2015-07-01T00:00:00,033083915+0000

2015-07-01T00:00:00,149927250+0000

2015-07-01T00:00:00,253094324+0000

红色的部分是时间出现了问题，出现了回退。

## 受影响的版本

使用RHEL 6.4 及其以上版本，使用ntpd –x(渐变模式)的应用服务器。

## 解决方案

升级ntp和ntpdate包到4.2.6p5-3.el6\_6.x86\_64版本。

# 闰秒对FusionInsight的影响

## 当前状态

当前FusionInsight的NTP采用如下架构：



在当前的FusionInsight集群中，所有的数据节点和备OMS节点都是向主OMS进行时间同步，即将主OMS作为ntp时钟源；数据节点和备OMS的ntp的工作模式为“slew”。

主OMS和外部ntp服务器进行时间同步，但是主要是通过一个脚本定时检测主OMS与外部ntp服务器之间的时间差距，然后再通过调整tick校正时间。因为其是通过脚本来校正时间，而不是通过ntp协议校正时间，所以集群并不感知存在闰秒，集群只感觉集群的时间比外部ntp服务器快了一秒。

## 影响

从2.1和2.3的分析可知，数据节点和备OMS节点、主OMS节点都不会触发SuSE的内核Bug，其正确地配置使用ntp服务，并且将ntp的工作模式设置为“slew”。

当发生闰秒调整时，外部ntp服务器的UTC时间较FusionInsight集群的UTC时间多一秒，当主OMS节点的ntp监控脚本发现时间差异之后，会调整本机的tick缓慢地校正时间，使其慢慢地和外部NTP服务器的时间一致，最长校正时间需要2.7小时。

另外主OMS与外部ntp的时间相差一秒，不会对FusionInsight的业务造成影响，因为安全认证容忍的时间差是最多是5分钟，远远没有达到这个值。

而数据节点和备OMS节点时时刻刻都通过ntp客户端和主OMS保持着时间同步，并且工作模式是“slew”，所以，数据节点和备OMS节点慢慢地就会将时间校正了。

因为FusionInsight集群是通过这种缓慢同步的方式校正时间，上层应用几乎无法感知到时间有发生变化，所以对上层应用没有影响。

## 结论

通过以上的分析可以得出：

闰秒以及SuSE的内核Bug对FusionInsight集群没有影响。

闰秒以及RedHat的内核Bug对FusionInsight集群没有影响。

因为上述分析对FusionInsight没有影响，所以上述描述的解决办法可以都不处理，并不会影响集群的运行。