浪潮英信服务器 Whitley 平台

推荐实践

文档版本 V 1.0

发布日期 2021-10-15

环境保护

请将我方产品的包装物交废品收购站回收利用，以利于污染预防，共同营造绿色家园。

商标说明

Inspur 浪潮、Inspur、浪潮、英信是浪潮集团有限公司的注册商标。
本手册中提及的其他所有商标或注册商标，由各自的所有人拥有。

内容声明

您购买的产品、服务或特性等应受浪潮集团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，浪潮集团对本文档的所有内容不做任何明示或默示的声明或保证。文档中的示意图与产品实物可能有差别，请以实物为准。本文档仅作为使用指导，不对使用我们产品之前、期间或之后发生的任何损害负责，包括但不限于利益损失、信息丢失、业务中断、人身伤害，或其他任何间接损失。本文档默认读者对服务器产品有足够的认识，获得了足够的培训，在操作、维护过程中不会造成个人伤害或产品损坏。文档所含内容如有升级或更新，恕不另行通知。

技术支持

技术服务电话：4008600011

地　　址：中国济南市浪潮路 1036 号

浪潮电子信息产业股份有限公司

邮　　箱：lckf@inspur.com

邮　　编：250101

前言

目标受众

本手册主要适用于以下人员：

- 技术支持工程师
- 产品维护工程师

建议由具备服务器知识的专业工程师参考本手册进行服务器运维操作。

注意

- 请使用浪潮认证的驱动程序进行 OS 环境搭建。您可访问浪潮官网进行驱动下载，进入浪潮官网首页，顶部导航栏选择支持下载 > 产品支持 > 驱动下载，根据页面提示查找产品对应的驱动程序。如使用非浪潮认证的驱动程序，可能会引起兼容性问题并影响产品的正常使用，对此浪潮将不承担任何责任或义务。
- BIOS、BMC 的设置对配置您的服务器至关重要，如果没有特殊的需求，请您使用系统出厂时的默认值，请勿随意更改参数设置。首次登录时，请及时修改 BMC 用户密码。

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

图标	说明
 危险	如不当操作，可能会导致死亡或严重的人身伤害。
 警告	如不当操作，可能会导致人员损伤。
 注意	如不当操作，可能会导致设备损坏或数据丢失。

图标	说明
 提示	为确保设备成功安装或配置，而需要特别关注的操作或信息。
 说明	对操作内容的描述进行必要的补充和说明。

变更记录

版本	时间	变更内容
V1.0	2021-10-15	首版发布

目 录

1 介绍

本文档主要介绍如何在浪潮英信服务器上安装 OS 系统，以达到最佳的性能和稳定性。

本文档主要从固件、CPU、硬盘、内存、网卡、RAID 卡、OS 等方面给出浪潮英信服务器的推荐调优策略，以期帮助您在使用浪潮英信服务器时获得最佳的性能和稳定性。

2 BIOS 优化建议

2.1 固件版本

固件会随着版本升级修复安全漏洞和 bug 等。为了服务器安全稳定运行，建议查找官网，确认当前服务器的固件版本已经是最新的固件版本，如果不是最新版本，建议升级到最新版本。版本需要搭配使用，服务器固件包括 BIOS、BMC、CPLD 以及各部件的固件，固件之间存在搭配关系，例如，旧的 BIOS 版本搭配新的 BMC 版本可能出现问题，如果不能都升级到最新版本，建议查找官网或联系客服咨询版本之间是否存在不兼容问题。

2.2 能效调优综述

BIOS 默认设置是一种偏性能的均衡模式，用户需要针对具体应用场景做优化设置。

BIOS 提供了一个 Power and Performance Profile 的选项，用于一键应用场景设置。用户可以根据具体应用场景选择对应的选项。对于具体的应用场景可以在这六种应用场景基础上做微调，使用方法为：切换到某个场景下，然后再切回 Custom 模式做微调。

能效调优是一个系统工程。需要对计算机系统有深入的理解和整体把握，Application、OS、驱动、BIOS、硬件等各部件最佳配合才能达到最优效果。

调优时，需要充分了解应用场景的特点是 CPU 密集型、内存密集型、IO 密集型、虚拟化场景还是几种类型的混合类型，需要了解限制性能发挥的瓶颈在什么地方。然后针对瓶颈做针对性的优化，BIOS 主要是选项设定。

调优过程中可以使用一些单项测试软件或者和应用场景相符的综合基准测试软件辅助调优，对选项调整的效果做量化和定性分析。最后由实际应用做最后检验。

系统运行时有一个最大功耗的限制，并非关闭电源管理即可达到最大性能，需要针对具体的应用做不同的设定。

例如，当某些 Core 进入 C6 时，可以节省出更多的功耗空间给其他核心使用，使得工作的核心运行在更高的频率。当 Static Turbo 打开时，每颗 CPU 无论是否有负载都运行在睿频，因为使用的功耗更多，从而限制了有负载的核心可以达到的最高频率。所以对于负载不重，时延要求不高的应用场景不建议打开 Static Turbo 和关闭 C6。对于时延敏感且任务小的场景才需要打开 Static Turbo 和关闭 C6。对于大而长的任务，需要的是平均性能，而不是短响应时间，这时候关闭电源管理可能也不是好的选择，让功耗在不同部件之间动态切换是不错的选择。

总之，需要根据具体的应用场景来做针对性调整才能让系统运行在最佳状态。

2.3 重要参数说明

2.3.1 预取

表 2-1 BIOS 重要参数

选项名	说明	使用建议
DCU Streamer Prefetcher	• Cpu Level 1 Cache（Data Cache Unit）流式预取 • DCU流预取功能可以预读取CPU的数据，从而缩减数据的读取时间	• 建议设置为Enable • Power Efficiency场景下可以关闭
DCU IP Prefetcher	• Cpu Level 1 Cache（Data Cache）基于IP的预取 • DCU IP预取功能依据历史记录执行预取算法，从而缩减数据的读取时间	• 建议设置为Enable • Power Efficiency场景下可以关闭
Hardware Prefetcher	• Cpu Level 2 Cache (Middle Level Cache）预取，属于Streamer Prefetcher • CPU将可能使用到的指令或数据预取到L2 Cache中，以此提高系统性能	• 建议设置为Enable • Power Efficiency场景下可以关闭
Adjacent Cache Prefetch	• Cpu Level 2 Cache (Middle Level Cache)预取，属于Spatial Prefetcher • CPU在读取数据时，会将要读取的数据旁边或	• 建议设置为Enable • Power Efficiency场景下可以关闭

选项名	说明	使用建议
	邻近的数据也预先读取到Cache中，可以提高Cache命中率	
LLC Prefetch	• Cpu Level 3 Cache (Last Level Cache)预取 • 使能LLC预取功能后，core prefetcher可以将数据直接预取到LLC中，无需填写MLC	• 一般建议设置为Enable • 内存时延敏感的场景，建议打开LLC Prefetch，以提高Cache命中率，缩减内存访问时延 • Power Efficiency场景下可以关闭
XPT Prefetch	Extended prediction table (XPT)预取是一种用于减少本地内存访问延迟的功能。每个核中的一个“LLC miss predictor”，当预测LLC查找“miss”时，XPT会在LLC查找的同时发出一个投机性的DRAM读请求。XPT预取的数据将在内存控制器中等待很短的时间，只有真正发生LLC miss情况下才会返回到内核。因为LLC查找和投机性DRAM读并行进行，从而可以缩减本地内存访问延迟	• 内存时延敏感的场景，建议打开 • Power Efficiency场景下可以关闭 说明： • XPT 不能与 2LM 操作一起使用，它依赖于 SNC 操作。因此，XPT 不能用于 UMA 的内存映射 • 在不同机型，可能会表示为 RDCur XPT Prefetch
XPT Remote Prefetch	Extended prediction table (XPT)远程预取开关	• 内存时延敏感的场景，建议打开 • Power Efficiency场景下可以关闭

选项名	说明	使用建议
KTI Prefetch	Kti预取开关	• 内存时延敏感的场景，建议打开 • Power Efficiency场景下可以关闭
L2 RFO Prefetch Disable	• MLC RFO（Request for Ownership）预取开关 • 可缩减内存时延，增加内存访问吞吐量 • 对CPU密集型应用场景可能有副作用	• 内存时延敏感的场景，建议打开 • Power Efficiency场景下可以关闭

2.3.2 P State and C state

- SpeedStep（Pstates）P-state 开关，一般需要打开，关闭后，CPU 无法运行在 Turo 频率。
- Turbo Mode：睿频开关，可以使 CPU 运行比标称频率更高的频率。
- Enable Monitor MWAIT：自动 Core C 状态开关。
- CPU C6 Report：C6 状态开关。
- Enhanced Halt State (C1E)：C1E 状态开关。
- OS ACPI Cx：CPU C 状态和 ACPI C 状态的映射关系。
- Package C-State：Package C State 限制。
- Hardware P-States（HWPM）：有 Native mode、Native Mode without Legacy support、Out of Band Mode、Disable 四个选项。
- Native mode，OS 可选择 HWPM 接口或 Legacy p-state 接口控制 CPU p-state。
  - Native Mode without Legacy support 下，OS 只能使用 HWPM 接口控制 CPU p-state。如果 OS 不支持 HWPM，则 HWP 自动以默认值运行。
  - OOB 模式是一种不依赖于 OS 的运行模式，带外控制器可以通过 PECI 接口管理 CPU p-state。
  - Disable 时，OS 只能使用 Legacy p-state 接口控制 CPU p-state。

- EPP Enable：EPP（Energy Performance Preference）开关，关闭时使用 EPB 设定作为 EPP 设定。
- EPP Profile：选择 HWPM EPP（Energy Performance Preference），有 Performance、Balanced Performance、Balanced Power、Power 四个选项，根据应用场景设定。
- Energy Performance Bias（EPB）：有 Performance，Balanced Performance，Balanced Power，Power 四个选项，根据应用场景设定。
- Power Performance Tuning：选择由谁控制 EPB，有 OS 控制、BIOS 控制、PECI 控制三种选择，默认设置 OS 控制。

2.3.3 RAPL

- PL1 Limit：启用/禁用 PL1 功能。

  如果禁用此选项，BIOS 将为“PL1 Power Limit”和“PL1 Time Window”设置默认值。

- PL1 Power Limit：PL1 功率限制(瓦)。

  取值范围为 0 ~ Fused value。如果该值为 0，则设置为 Fused value，大于 Fused value 值将不会被设置。

- PL1 Time Window：PL1 值，单位为秒。

  取值范围为 0 ~ 56。指示应该维护 TDP 值的时间窗口。如果该值为 0，则使用 fused value。

- PL2 Limit：启用/禁用 PL2 功能。

  如果禁用此选项，BIOS 将为“PL2 Power Limit”和“PL2 Time Window”设置默认值。

- PL2 Power Limit：PL2 功率限制(瓦)。

  取值范围为 0 ~ Fused value。如果该值为 0，则设置为 Fused value。大于 Fused value 值将不会被设置。

- PL2 Time Window：PL2 值，单位为秒。

  取值范围为 0 ~ 56。指示应该维护 TDP 值的时间窗口。如果该值为 0，则使用 fused value。

应用建议：对于高性能场景，将 PL1 和 PL2 的 power limit 和 time window 设置为最大，使得应用可以稳定运行在需要时可以持续运行在高性能状态。

2.3.4 Power and Performance Profile

- Power and Performance Profile：有 HPC、Balanced、Power Saving、Low Latency、OLTP、Virtualization、Custom 等几种应用场景。
- Custom：不是应用场景，是供用户自行定制使用的一种模式。
- 用于一键设置应用场景，可根据应用场景选择。
- 对于具体的应用场景可以在这六种应用场景基础上做微调。
  - 使用方法：切换到某个场景下，然后再切回 Custom 模式做微调。

2.3.5 NUMA and SNC

Non Uniform Memory Access（NUMA）架构将处理器、内存和 I/O 总线等硬件资源的集合根据亲和程度划分为不同的 NUMA 节点，处理器访问本地 NUMA 节点内的内存或 I/O 资源通常比处理器访问本地 NUMA 节点外的内存或 I/O 资源（通过节点互连进行访问）更快。ACPI 定义了接口，允许平台在启动时静态地向 OSPM 传递 NUMA 节点拓扑信息，并在运行时动态地向系统中添加或删除资源。OS 可根据这些信息优化资源分配，以提升性能。一般开启此选项可以提升性能。

SNC（Sub Numa Cluster）将 LLC 分解为不关联的 cluster，每个 cluster 和一组内存控制器绑定，可以缩减 LLC 访问的平均延迟。在 SNC 模式下，LLC 片根据它们与每个 Cluster 的内存控制器的接近程度，被分给不同的 Cluster。

因为 LLC 被划分为更小的片，容易造成性能波动，可根据具体应用场景优化。

2.3.6 虚拟化

- VMX：CPU 虚拟化开关
- Vt-d：IO 虚拟化开关
- SR-IOV：虚拟化 IO 开关

应用建议：

支持虚拟化特性时需要打开。虚拟化技术总体上提升了设备的利用率，因为虚拟设备之间有资源共享等原因，可能对某些性能要求高的应用可能会造成副作用。

在 IO sensitive 场景需要关闭 Vt-d 以提升 IO 性能。

2.3.7 IO

- Workload Configuration：IO 场景下设置为 IO Sentive，其它场景设置为 Balanced。

- Relaxed Ordering：乱序总线事务。

  对于某些应用，可以有效地提高数据传送效率，但是支持 Relaxed Ordering 的设备，需要较多的数据缓存和硬件逻辑处理这些乱序。

- ASPM：Active State Power Management (ASPM)，用于降低 active 状态即 D0 下 PCIE Link 的功耗，在高性能和低时延应用场景下可以关闭，省电应用场景下需要打开。

2.3.8 杂项

Stale/Directory A-to-S (A2S)：是目录行状态的转换。

内存中的目录有三种状态：I、A 和 S。

- 无效（I）状态意味着数据是干净的，不存在于任何其他 CPU Socket 的缓存中。
- snoopAll （A）状态意味着数据可能存在于另一个 CPU Socket 中，处于排他或修改状态。
- 共享（S）状态意味着数据是干净的，可以跨一个或多个 CPU Socket 的缓存共享。

当 A2S 使能时，在 A 状态的目录行只返回 snoop miss 的情况下，该行将转换到 S 状态。这样，对该行的后续读取，因为该目录行在 S 状态，而不必窥探，从而节省了延迟和窥探带宽。

Stale A-to-S 在有大量跨 Socket 访问的工作负载中可能是有益的。对于大于两个 CPU Socket 的配置，建议启用此选项。这可以在某些情况下改善远程延迟和带宽，特别是在大量使用 Intel® UPI 链路时。

3 内存优化建议

内存DIMM条的部署选型建议遵循以下配置原则：

- DDR4 的内存 DIMM 条性能更好，DDR3 更成熟稳定。
- DDR4 的内存：2Rank 规格的内存性能更好。
- 内存实际运行的频率越高性能越好。
- 尽量保证每个内存通道都部署有内存 DIMM 条。
- 采用平衡部署，每个内存通道上配置同样多的内存，配置型号一致的内存。
- 如果DIMM条数目不是内存通道的倍数，Whitley平台可参考如下部署：
  - 单 CPU 内存插法：推荐配置的 DIMM 数量为 1、2、4、6、8、12、16。

图 3-1 单 CPU 内存插法

DDR4 Qty	CPU0															
	iMC0				iMC1				iMC2				iMC3			
	C0		C1		C2		C3		C4		C5		C6		C7	
	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1
1	v															
2	v								v							
4	v				v				v				v			
6	v		v		v				v		v		v			
8	v		v		v		v		v		v		v		v	
12	v	v	v	v	v	v			v	v	v	v	v	v		
16	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v

  - 双 CPU 内存插法：推荐配置的 DIMM 数量为 1、2、3、4、8、12、16、24、32。

图 3-2 双 CPU 内存插法

DDR4 Qty	CPU0																CPU1															
	iMC0				iMC1				iMC2				iMC3				iMC0				iMC1				iMC2				iMC3			
	C0		C1		C2		C3		C4		C5		C6		C7		C0		C1		C2		C3		C4		C5		C6		C7	
	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1	D0	D1
1	v																															
2	v																v															
3	v								v								v															
4	v								v								v								v							
8	v				v				v		v						v				v				v				v			
12	v		v		v				v		v		v				v		v		v				v		v		v			
16	v		v		v		v		v		v		v				v		v		v		v		v		v		v		v	
24	v	v	v	v	v	v			v	v	v	v	v	v			v	v	v	v	v	v			v	v	v	v	v	v		
32	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v

4 硬盘优化建议

4.1 SSD 型号选择

选择高耐写性（Endurance）、高带宽与高 IOPS、低 Latency 的 SSD 产品，例如 DWPD 3 以上或 Optane 产品。

4.2 固件配置选择

- 选择对应服务器系列浪潮官网上最新的 BMC 和 BIOS 版本。
- 建议使用最新的 SSD FW 版本，规避旧版本可能带来的问题风险。
- BIOS 设置中的 Maxpayload 值建议取 256bytes。
- BIOS 设置中打开“超线程”技术，提升系统吞吐量。

表 4-1 超线程参数

BIOS配置项	建议参数	配置项说明
Intel（R）HT Technology	Enabled	Intel超线程技术开关选项

- BIOS 设置中打开 NUMA 特性，提升系统的应用处理效率。

表 4-2 NUMA Support

BIOS配置项	建议参数	配置项说明
NUMA Support	Enable	NUMA特性开关选项

- BIOS 设置中关闭“节能”模式，提升系统整体性能。

表 4-3 节能配置项

BIOS配置项	建议参数	配置项说明
Package C State limit	C0/C1 state	CPU C状态限度配置
CPU C3 report	Disable	向OS报告C3状态
CPU C6 report	Disable	向OS报告C6状态
Enhanced Halt State (C1E)	Disable	--

4.3 软件版本选择

- 使用 SATA/SAS SSD 搭配控制器时，需要安装控制器的最新驱动版本。
- 使用 NVMe SSD 时，需要更新与 OS 适配的最新 VMD 驱动。
- 测试工具建议选择高版本的 FIO 工具。

4.4 系统配置选择

- 选择多核数、多线程、高频率的 CPU 型号，以提升整机系统性能。
- 系统内存满配，确保每个内存通道中都部署内存 DIMM 条，保证平衡部署，每个内存通道上配置同样数量的内存，尽量选择高频率的内存。
- 选择带缓存且缓存容量更大的 RAID 卡，RAID 配置对容量要求较高选择 RAID 5，对性能要求较高选择 RAID 10，对可靠性要求较高选择 RAID 6。

4.5 OS 下参数选择

- SATA SSD 顺序带宽建议队列深度 32，线程数 1；随机 IOPS 选择队列深度 32，线程数 4。
- SAS SSD 和 NVMe SSD 顺序带宽建议队列深度 128，线程数 1；随机 IOPS 选择队列深度 128，线程数 4。

4.6 测试手法选择

- 性能测试前确保 SSD 已经进入稳态，否则会有性能抖动。
- SSD 性能测试前需要进行 secure erase 操作，Optane 产品可不做 erase 操作。

5 网卡优化建议

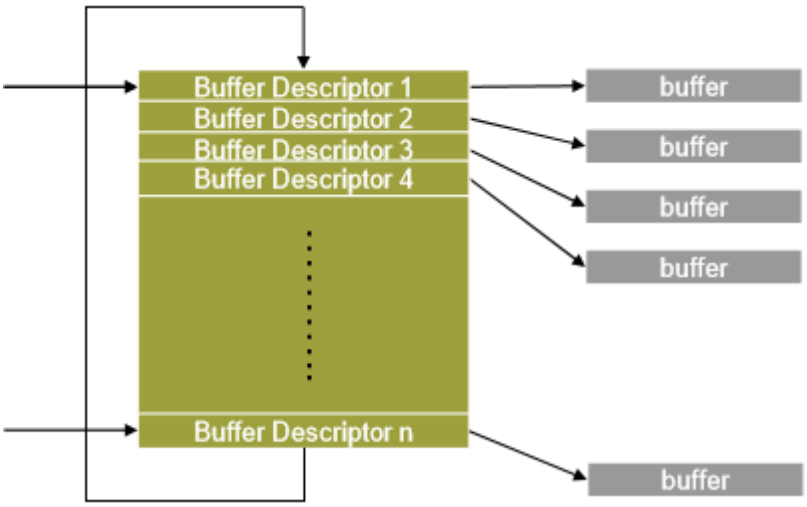
5.1 安装网卡原厂驱动

OS 自带的驱动通常版本较低，有很多原厂已经解掉的问题和新增的功能并没有包含在 inbox 驱动中，建议升级到原厂最新的驱动版本使用。

5.2 网卡 Ring Size 调整

- TX/RX ring size 是 TX/RX ring 的缓冲描述符编号。
- 网卡硬件和驱动程序使用描述符来发送和接收数据包。
- Larger ring size 提高了对数据包丢失的恢复能力，但增加了缓存/内存的占用，导致性能下降。
- 使用 ethtool -G <devname> tx 2047 rx 2047 进行调整。

图 5-1 Buffer



5.3 网卡 GRO（Generic Receive Offloading）

- GRO 将来自一个流的多个接收数据包合并成一个大数据包。这只允许处理单个数据包并减少开销，从而获得更好的性能。
- 在数据包无法聚合的情况下，GRO 就没有帮助，并可能带来性能损失。

- 使用 ethtool -K <devname> gro on 进行调整。

5.4 网卡 RSS（Receive Side Scaling）

- 为接收端负载分配提供了一种机制，可以将不同的数据流分散到不同的接收队列以均匀地分布负载。
- 使用 ethtool -L <devname> combined 16 进行调整，其中 16 代表设置为 16 个队列。

5.5 网卡 IRQ 亲和

- 将中断绑定到不同的 CPU 核，使数据包可以分散到不同 CPU 内核进行处理，可以有效的利用 CPU 多核的特性，提高整体性能。
- Linux 中有 irqbalance 会自动平衡所有 CPU 内核中断进程，但多数情况下 Linux 自动的操作并不是最高效的。
- 建议关闭系统自带的 irqbalance，并手动调整中断绑定。
- 使用 systemctl stop irqbalance 关闭系统自带中断调整。
- 使用 echo 1 >/proc/irq/[interrupt number]/smp_affinity_list 手动分配 CPU 核心绑定中断。

6 RAID 卡优化建议

服务器使用本地硬盘提供存储功能，RAID 卡的优化首先根据业务模型，当前业务需求，包括容量、业务压力、可靠性综合选择合适的 RAID 和硬盘，保证存储硬件性能不小于当前的业务需求，再进行合理的优化发挥服务器存储的最大性能。

- 选择带缓存且缓存容量更大的 RAID 卡。
- 针对 HDD，RAID 卡选择 Write Back/Controller Cache。针对 SSD，RAID 卡选择 Write Through/SSD IO bypass。
- RAID 配置对容量要求较高选择 RAID 5，对性能要求较高选择 RAID 10，对可靠性要求较高选择 RAID 6。
- 根据业务系统 IO 特点，选择合适的 RAID 条带，混合负载场景建议采用默认的 256KB 条带大小。
- 建议关闭磁盘自身的缓存，如果 RAID 卡不带超级电容，建议打开硬盘缓存。
- 已经做 RAID 条带化的硬盘，在操作系统上不要再进行拆分，做软条带。

7 OS 优化建议

OS 可以考虑从如下方面设置低时延和最佳性能调优。

7.1 CPU 节能优化

7.1.1 Linux 系统

CPU 节能方面的优化，建议优先从 BIOS 中设置(CState/PState/Hardware PState/Turbo 等)；若 BIOS 无法设置或设置后不起作用，可以在 OS 中通过如下步骤将 CPU 设置为最大性能模式：

1. 修改 grub 设置（以 RHEL 7 为例）：

   # vim /etc/default/grub

   在 GRUB_CMDLINE_LINUX 行的引号内增加：

   intel_idle.max_cstate=0 processor.max_cstate=0 idle=poll intel_pstate=disable

   重新生成 grub.cfg 文件

   在 leagcy 模式执行：

   # grub2-mkconfig -o /boot/grub2/grub.cfg

   在 UEFI 模式执行：

   # grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg

2. 设置 CPU 性能状态为 performance：

   # cpupower frequency-set -g performance

   # vim /etc/sysconfig/cpupower

   CPUPOWER_START_OPTS="frequency-set -g performance"

   CPUPOWER_STOP_OPTS="frequency-set -g performance"

   # systemctl restart cpupower; systemctl enable cpupower

3. 对于 RHEL/CentOS 系统还需要通过 tuned 设置性能策略：

   # tuned-adm profile latency-performance

   （或直接关闭 tuned 服务：systemctl stop tuned ; systemctl disable tuned）

4. 重启生效：

    # reboot

7.1.2 ESXi 系统

BIOS 中关闭节能相关选项后，CPU 会默认工作在最大频率，以 BIOS 设置为准，系统无法使用节能技术设置电源管理策略，不需要更多设置。

若 BIOS 无法关闭节能选项，可以通过如下方法，将系统的电源管理策略设置为“高性能”：

1. 登录 EXSi Web 管理界面。
2. 选择[管理/硬件/电源管理]，单击“更改策略”。
3. 选择“高性能”，确定。

7.2 进程调度优化

默认情况下，Linux 会使用 irqbalance 服务将硬件中断均匀的分布到多处理器系统的各个处理器核心上，以便能够使处理器性能负载均衡，但是这样做可能有三个缺点：

- 会导致多个进程争夺 CPU 资源。
- 影响 CPU Cache 的命中。
- 自动分配的硬件设备的中断处理核心与硬件如网卡的归属 CPU numa node 不同，会导致出现跨 NUMA 访问的情况，影响处理速度。

系统的 irqbalance 服务，在复杂的系统进程调度下，受很多因素的影响，可能无法实现可靠的负载均衡，故在实际低延时高性能的应用场景中，建议关闭 irqbalance 服务，对需要低延迟高性能工作的硬件设备的中断或进程，设置手动绑定，有如下几种方式。

7.2.1 进程绑定

通过 taskset 可以设置进程 CPU 的绑定，使用该工具将不同的进程绑定到不同的核，保持彼此的独立运行，互不干扰，能提升单进程单任务的处理效率，降低延时。

使用如下命令，替换 mask 为指定 CPU 核心编号，将 pid 对应的进程绑定到 CPU 核心：

    # taskset -p mask pid

使用如下命令，将 mask 替换为 CPU 核心编号，program 替换为要运行的程序，将该程序绑定到指定的 CPU 核心：

    # taskset mask -- program

    使用如下命令，将程序绑定到多个 CPU 核心：

taskset -c 0,5,7-9 -- program

7.2.2 进程优先级

系统进程的 nice 值表示进程的执行优先级，范围是-20~19，值越小系统执行的优先级越高。

对于低延时的应用场景，如果在计算系统中存在多个对响应要求不同的系统，可以通过 nice 工具为系统下运行的进程指定不同的任务优先级，对低延时要求的进程分配高优先级，获得更多的 CPU 处理时间，以将 763 进程的优先级从-8 调整到-12 为例，方法如下：

- 查询进程当前优先级：

  # ps -alx | grep audispd

  F UID PID PPID PRI NI VSZ RSS WCHAN STAT TTY TIME COMMAND

  4 0 763 761 12 -8 84500 820 futex_ S<sl ? 0:00 /sbin/audispd

- 调整进程的优先级：

  # renice -n -12 -p 763

  763 (process ID) old priority -8, new priority -12

- 确认进程优先级调整生效：

  # ps -alx | grep audispd

  F UID PID PPID PRI NI VSZ RSS WCHAN STAT TTY TIME COMMAND

  4 0 763 761 8 -12 84500 820 futex_ S<sl ? 0:00 /sbin/audispd

7.2.3 NUMA 绑定

NUMA 架构下，远端内存访问的延时会比本地延时大 2 倍以上，低延时场景需要 CPU 尽可能的访问本地内存降低延时。

自动 NUMA 平衡是 Linux 3.8 内核引入的新特性，可以自动优化应用程序在 NUMA 系统运行，在 RHEL 7 系统中默认情况下是打开的；对于类似 Oracle 这种已经做过 NUMA 优化的应用程序，不建议开启该功能，由应用程序自身根据最优方式调用，对于一些需要手动使用 numactl 进行节点绑定优化的场景也不建议打开；

临时关闭，立即生效：

echo 0 > /proc/sys/kernel/numa_balancing

写入 sysctl.conf 文件，永久生效：

sysctl -w kernel.numa_balancing=0

sysctl -p

numactl 工具可以将应用程序绑定到指定的 CPU Node 和 Memory Node 上；

较常用的参数如下：

--hardware	查看节点的 CPU core 及内存分布情况
--cpunodebind={nodes}	把执行的 command 全部绑到节点上，节点有可能包括多个 CPU
--physcpubind={cpu core}	把执行的进程全部绑定到对应 CPU 上，CPU 号根据 cpuinfo 里显示设置
--membind={nodes}	只在对应的节点上分配内存

可以通过如下指令获取更多使用方法：

```
# man numactl
```