|  |  |
| --- | --- |
| ICS | 点击此处添加ICS号 |
| CCS | |  | | --- | | D:\000000部门项目\09标准化插件开发\程序源代码\StandardEditor_ShanDongKeXieYuan\团标首页面字母T.pngD:\000000部门项目\09标准化插件开发\程序源代码\StandardEditor_ShanDongKeXieYuan\团标首页面字母T后面的反斜杠.png |   点击此处添加CCS号 |

团 体 标 准

T/CAAMTB XX—202XX

f g g

智能驾驶操作系统性能测试方法

Intelligent driving operating system performance testing methods

**征求意见稿**

**中国汽车基础软件生态标准专委会**

**（AUTOSEMO）**

XXXX - XX - XX发布

XXXX - XX - XX实施

       发布

目次

[前 言 1](#_Toc852112139)

[1 范围 2](#_Toc2027761977)

[2 规范性引用文件 2](#_Toc30069549)

[3 术语和定义、缩略语 2](#_Toc720252998)

[3.1 术语和定义 2](#_Toc2074302894)

[3.2 缩略语 4](#_Toc546731014)

[4 通用测试环境要求 4](#_Toc1973110432)

[4.1 硬件环境 4](#_Toc624553650)

[4.2 基础条件 4](#_Toc2120612661)

[4.2.1 稳定性要求 4](#_Toc1498387815)

[4.2.2 安全性要求 4](#_Toc2010761983)

[5 实时性技术要求与测试方法 5](#_Toc2073979089)

[5.1 调度实时性 5](#_Toc1559474366)

[5.1.1 调度延迟 5](#_Toc47757727)

[5.1.2 系统调用延迟 5](#_Toc1667145608)

[5.1.3 线程切换时间 6](#_Toc598878566)

[5.1.4 关抢占时间 6](#_Toc1756919661)

[5.2 进程间通信 7](#_Toc1118180891)

[5.2.1 信号延迟 7](#_Toc636840140)

[5.2.2 互斥锁延迟 7](#_Toc368143267)

[5.2.3 消息队列延迟 8](#_Toc1880051374)

[5.2.4 优先级反转 8](#_Toc417653174)

[5.2.5 共享内存延迟 9](#_Toc227736935)

[5.2.6 本地socket延迟 9](#_Toc314357975)

[5.2.7 信号量延迟 9](#_Toc2114097149)

[5.3 中断 10](#_Toc905692998)

[5.3.1 中断延迟时间 10](#_Toc618127450)

[5.3.2 中断运行时间 10](#_Toc31877873)

[6 计算性能技术要求与测试方法 11](#_Toc1071717499)

[6.1 CPU性能 11](#_Toc1410658304)

[6.2 内存性能 11](#_Toc646433658)

[7 通信性能技术要求与测试方法 12](#_Toc455957430)

[7.1 I/O性能 12](#_Toc1054873514)

[7.2 网络性能 13](#_Toc887981669)

[7.3 时钟 16](#_Toc874085298)

[7.4 CAN通讯性能 17](#_Toc475091313)

[7.5 UART通信性能 18](#_Toc1355977821)

[7.6 SPI通信性能 18](#_Toc700131407)

[7.7 PCIe通信性能 18](#_Toc784039712)

[7.8 I2C通信性能 19](#_Toc428339363)

[8 智驾相关性能技术要求与测试方法 19](#_Toc834147705)

[8.1 启动时间 19](#_Toc747230319)

[8.2 通信中间件性能 20](#_Toc2022698652)

[8.3 摄像头性能 20](#_Toc1391396294)

前言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件由中国汽车工业协会归口。

本文件主要起草单位：

国科础石 (重庆)软件有限公司

重庆长安科技有限责任公司

本文件编制参与单位：

上海汽车集团股份有限公司零束软件分公司

上汽集团创新研究开发总院

中汽创智科技有限公司

苏州挚途科技有限公司

华创惠元（成都）科技有限公司

国家新能源汽车创新中心

重庆中科汽车软件创新中心

OpenSDV汽车开源联盟

西安邮电大学软件学院

同济大学汽车学院

浙江极氪智能科技有限公司

广汽集团汽车工程研究院

长城汽车股份有限公司

中兴通讯股份有限公司

斑马网络技术有限公司

北京地平线机器人技术研发有限公司

黑芝麻智能科技有限公司

深圳联友科技有限公司

普华基础软件份有限公司

星河智联汽车科技有限公司

智能驾驶操作系统性能测试规范

* 1. 范围

本标准规定了智能驾驶操作系统内核性能的技术要求及测试方法，重点是智能驾驶操作系统内核实时性、基础性能的技术要求以及测试方法。

本标准适用于智能驾驶操作系统内核产品的开发、测试及性能评价等工作。

本标准描述的性能指标主要是操作系统内核相关性能指标，这些性能指标是智能驾驶应用整体性能的重要组成部分。

本标准不包含智能驾驶系统硬件性能指标。

除本标准所包含的操作系统内核性能指标外，智能驾驶应用整体性能指标还应当包含硬件性能、异构系统性能、GPU性能等非操作系统内核性能指标。在不同的业务场景和不同的系统配置中，本标准中所述性能指标评测结果会有不同。可以在不同的硬件规格、应用场景、参数和限制条件下，考察本规范性能指标的差异。应当在评测报告或者相关指标的技术要求中明确硬件规格、应用场景、参数和限制条件等。

* 1. 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 11457-2006 信息技术 软件工程术语

GB/T 25069-2022 信息安全技术 术语

GB/T 28457-2012 SSL协议应用测试规范

* 1. 术语和定义、缩略语

术语和定义  
下列术语和定义适用于本文件。

智能驾驶操作系统 intelligent driving operating system

运行于车载智能计算基础平台异构硬件之上，支撑智能网联汽车驾驶自动化功能实现和安全可靠运行的操作系统软件集合。

1. 车载智能计算基础平台是指支撑智能网联汽车驾驶自动化功能实现的软硬件一体化平台，包括芯片、模组、接口等硬件以及系统软件、功能软件，以适应传统电子控制单元向异构高性能处理器转变的趋势。

进程 process

计算机中的程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位。

线程 thread

操作系统能够进行运算调度的最小单位，一个或者一组相关线程组成进程。

延迟 latency

一个事件被触发到操作系统首次对事件做出动作所需的时间间隔。本标准中的延迟是指由于操作系统导致的应用处理延迟。

调度延迟 scheduling latency

线程被事件唤醒到线程首次实际获得计算资源之间的间隔时间。

系统调用延迟 system call latency

由于系统调用在内核空间运行导致的调度延迟。在不可抢占系统中，线程在执行系统调用期间，不允许被中断，导致高优先级线程无法获得计算资源。

进程切换时间 scheduling time

操作系统切换线程运行上下文所耗费的时间。

关抢占时间 preempt disable time

在抢占式系统中，虽然在系统调用上下文可以被抢占，但是操作系统可以显式的禁止/打开抢占。关抢占时间是指内核禁止抢占到再次允许抢占之间的间隔时间。

信号延迟 signal latency

通常指一个进程发送信号到另一个进程接收到该信号并开始处理的时间延迟。

互斥锁延迟 thread mutex latency

一个线程释放互斥锁到另一个线程获取互斥锁之间的间隔时间。

消息队列延迟 queue latency

通过消息队列机制，从一个线程发送消息到另一个线程，直到另一个线程接收到消息的时间。

优先级反转 priority inversion scheduling

线程调度过程中的一种特殊现象。高优先级线程因为所需资源被低优先级进程占用而被阻塞，占用该资源的低优先级线程因其优先级低于其他进程也无法执行而释放资源，造成最高优先级进程反而在一段时间内无法执行，系统性能下降的情况。

进程间通讯耗时 inter-process communication time consuming

在两个测试线程之间，通过管道和网络套接字进行数据传输所需要的时间。

中断 interrupt

在CPU执行程序的过程中，出现了某种紧急或异常的事件（中断请求），CPU需暂停正在执行的程序，转去处理该事件（执行中断服务程序），并在处理完毕后返回断点处继续执行被暂停的程序，这一过程称为中断。

中断延迟 interrupt latency

外部设备发生中断事件，到驱动程序开始处理设备事件之间的时间间隔。

吞吐量 throughput

单位时间内传输有效数据的数量，或者系统在单位时间内处理请求的数量。

响应时间 response time  
 I/O 请求从发出到得到响应的间隔时间。

传输速率 transmission rate

在数据传输中，两个设备之间数据流动的物理速度称为传输速率。

[带宽](https://blog.csdn.net/weixin_45939085/article/details/125752840" \l "_29" \t "https://blog.csdn.net/weixin_45939085/article/details/_self) bandwidth

指在一个固定的时间内，从一端传送到另一端的最大数据量，即最大速率。

网络延迟 network latency

网络报文从发送到接收经过的延迟时间，一般由传输延迟和处理延迟组成。

网络抖动 network jitter

最大延迟与最小延迟的时间差。

[丢包率](https://blog.csdn.net/weixin_45939085/article/details/125752840" \l "_93" \t "https://blog.csdn.net/weixin_45939085/article/details/_self) loss rate

在一定的时间范围内，传输过程中丢失的报文数量与总报文数量的比率。

缩略语

下列缩略语适用于本文件。

CPU：中央处理器（Central Processing Unit）

IOPS：每秒的 I/O 请求数（Input/Output Per Second）

P95：表示95百分位数，一组n个值按数值大小排序，处于P%位置的值称第P百分位数。

* 1. 通用测试环境要求

通用测试环境要求主要包括硬件环境和基础条件。

* + 1. 硬件环境

测试结果一般会受到硬件规格、应用场景、参数和限制条件的影响。本标准中不规定测试环境的硬件规格，但是对于应用此标准生成的测试报告，需要明确测试硬件规格。

* + 1. 基础条件
       1. 稳定性要求

本标准规定的性能指标，是基于操作系统能够稳定运行的基础上进行测试的，其功能和稳定性应提前得到保障。

在测试本标准中规定的性能指标时，应该在测试过程中增加一些持续的压力。对于应用此标准生成的测试报告，需要标明测试环境所处的压力水平。

测试本标准中的指标前，至少要通过下面稳定性测试：

|  |  |
| --- | --- |
| **稳定性测试类型** | **稳定性要求说明** |
| 综合测试 | 连续稳定运行72小时，无宕机等异常 |
| 启动稳定性测试 | 重复启动100次，无异常 |
| 文件读写稳定性测试 | 测试10次，无异常 |
| 内存稳定性测试 | 测试内存占总内存90%，无错误报警 |

* + - 1. 安全性要求

操作系统内核的安全性（包括功能安全、信息安全）等不在本标准讨论范围内，测试的对象需要按需开启，并在测试报告中标明安全性策略。

测试方法和推荐方案的实施，应不能影响操作系统本身的安全策略。

* 1. 实时性技术要求与测试方法
     1. 调度实时性

调度实时性测评指标包括调度延迟、系统调用延迟、线程切换时间、关抢占时间。

* + - 1. 调度延迟

操作系统调度延迟是指操作系统的线程（最小调度单位）从变为可执行状态到首次被实际获得计算资源的时间间隔，是评价操作系统调度性能、响应速度和公平性的重要指标。调度延迟受操作系统调度器设计、调度算法、内核抢占机制和系统负载等多种因素的影响。

调度延迟最大值是操作系统在一定观测时间段内所有线程经历的最大调度延迟时间，反映了业务在最坏情况下的响应实时性情况，在系统设计、特别是设计确定性任务调度时，需要参考此值。

调度延迟平均值是操作系统在一定的观测时间段内所有调度延迟的平均值，反映了操作系统在一般情况的调度性能，可以为评估系统调度能力提供参考。

5.1.1.1 技术要求

调度延迟可能会影响关键线程的实时性、确定性，建议对关键线程进行调度延迟测试。

考虑到需要充分的量化调度延迟对关键线程的最大影响、平均影响，建议求出测试过程中的最大值和平均值。

实时任务一般要求1~10ms，关键线程调度延迟最大值应小于1毫秒。

5.1.1.2 测试方法

使用nanosleep函数设置任务睡眠时间t，记录开始睡眠的时间t1，理论上程序被唤醒后开始执行时间为t2 = t1 + t；测试过程建议持续一小时以上，调度切换次数至少一亿次。

由于调度延迟等原因，导致线程真实执行时间为t3, 则认为调度延迟为delta = t3 - t2;

考虑到测试误差、系统不同时刻运行压力不同等原因，建议分多次测试，持续一小时以上的时间，求取调度延迟平均值和最大值。

推荐测试方案：

（1）测试工具：测试套件rt-tests中的cyclictest工具。

（2）测试命令：cyclictest -i 100 -p 80 -n -t 1 -D 24h -N --policy=fifo，其中：

——i 唤醒频率：线程睡眠的时间，即实时线程100 微秒被唤醒一次；

——p 指定实时线程的优先级；

——n 纳秒睡眠 nanosleep；

——t 处理器分配的测试线程数；

——N 测试结果使用ns显示；

——policy 调度策略。

（3）此测试应当在足够的CPU、内存、IO、网络负载下进行。

* + - 1. 系统调用延迟

5.1.2.1 技术要求

在非实时系统中，系统调用延迟可能会影响关键线程的执行实时性、确定性，建议对系统调用延迟进行测试。

考虑到需要充分的量化指标反映调用延迟对关键业务的最大影响，建议求取测试过程中的最大值和P95值，其中P95值是指采集的数据按照从小到大排列位于95%位置的值。在非抢占系统中，系统调用延迟最大值应小于1毫秒。在抢占系统中，此指标不影响系统实时性，不用进行测试。

5.1.2.2 测试方法

系统调用延迟时间测试内容包括系统调用延迟最大值和P95值。测试过程需要持续一小时以上，系统调用至少一亿次。

测试方法如下：

1. 记录各个CPU最近一次的执行进入系统调用的时间点t1；
2. 记录各个CPU退出系统调用的时间点t2；
3. 则系统调用延迟的值为t2-t1；
4. 如果系统调用过程中发生系统调度，则重新开始计时；
5. 测试时间：一小时；

(6) 获取所有的系统调用延迟时间最大值和P95值。

推荐测试方案：根据测试方法，编写专用工具进行测试。

* + 1. 线程切换时间

线程切换时间，通常也称为上下文切换时间或线程切换开销，是操作系统从一个线程切换到另一个线程所需要的时间。线程切换时间，包括操作系统中保存当前线程状态和内存管理信息以及恢复下一个线程状态和信息的操作。

线程切换时间受操作系统硬件上下文切换、软件上下文切换、调度算法和负载等影响，会影响业务的实时性和稳定性。

线程切换时间最大值反映操作系统在一定的观测时间段内，发生的线程切换时间的最大值，反映了系统在最坏的情况下，业务实时性情况。在系统设计、特别是设计确定性任务调度时，需要参考此值。

线程切换时间平均值反映在一定的观测时间段内，操作系统处理的能力，作为设计参考。

5.1.3.1 技术要求

测试过程持续一小时以上，线程切换次数至少一亿次，建议求取测试过程中的最大值和平均值。

线程切换时间最大值应小于1 微秒。

5.1.3.2 测试方法

创建多个线程，并将所有线程组成一个环。

通过管道令牌在环上的传递，轮流调度各个线程运行。

记录总运行时间、切换次数和循环体的额外开销，计算出平均线程切换时间。

推荐测试方案：

1. 通过专有工具测试，在schedule函数前后增加时间戳，计算延迟；
2. 参考ftrace中的sched\_switch事件。

判断线程切换时间是否满足要求：线程切换时间 最大值< 1微秒。

* + 1. 关抢占时间

在某些临界区范围内，为了防止高优先级线程抢占当前线程，或者将当前线程迁移到其他CPU，需要保证当前线程不被抢占，这段不允许被抢占时间被称为关抢占时间。关抢占时间会影响关键线程的执行实时性、确定性。

关抢占时间最大值反映操作系统在一定的观测时间段内关抢占时间的最大值。为了提高系统确定性调度的精度，在系统设计时，特别是设计确定性任务调度时，需要参考此值。

5.1.4.1 技术要求

考虑到需要充分的量化关抢占时间对关业务的影响，建议求取测试过程中的最大值和P95值。为保证高优先级任务的实时性，关抢占时间应小于最大调度延迟，最大值应小于1000 微秒。

5.1.4.2 测试方法

关抢占时间测试内容包括测试获取关抢占时间的最大值和P95值，测试过程需要持续一小时以上。

测试方法如下：

（1）在操作系统内核关抢占时刻通过调用时间获取函数获取时间t1,在开抢占时刻获取时间t2，则该单次测试系统的关抢占时间delta= t2 - t1；

（2）测试时间：一小时；

（3）获取关抢占时间最大值、P95值。

推荐测试方案：

Linux内核通过preemptoff tracer来记录系统最大关抢占时间。通过preemptoff tracer监视系统每次抢占的开启和关闭，记录相关时间戳。通过分析这些时间戳，可以得出一次抢占关闭的持续时间，进而得到最大关抢占时间。一次关抢占时间的计算方法如下：

（1）在抢占禁用期间，记录时间戳t1；

（2）在抢占重新启用时，记录时间戳t2；

（3）计算本次抢占禁用的持续时间t = t2 - t1；

（4）获取关抢占时间最大值、P95值；

（5）需要注意嵌套关抢占的处理。

* + 1. 进程间通信

进程间通信测评指标包括信号延迟、互斥锁延迟、消息队列延迟、优先级反转、共享内存延迟、本地scokect延迟和信号量延迟。

5.2.1 信号延迟

信号是进程间通信的一种方式，信号延迟通常指一个进程发送信号到另一个进程接收到该信号并开始处理的时间延迟。信号延迟受操作系统的调度机制、负载情况等影响，指标过大表示业务进程间通信延迟大、系统的响应速度和性能不高。

信号延迟最大值反映操作系统在一定的观测时间段内信号延迟的最大值，在系统设计、特别是设计确定性任务调度时，需要参考此值。

信号延迟平均值反映在一定的观测时间段内操作系统处理的能力，作为设计参考。

5.2.1.1 技术要求

信号延迟会影响关键进程的执行实时性、确定性，建议对信号延迟进行测试，

考虑到需要充分的量化信号延迟对关键业务的最大影响、平均影响，建议求出测试过程中的最大值、P95值和平均值。信号延迟时间基于调度和中断都在最低延迟情况下的测量最大值应小于40 微秒，重点考虑信号逻辑本身的复杂性和实时性。

5.2.1.2 测试方法

创建实时线程A和B，线程A发送信号给B，B接收信号；

通过时间获取函数记录实时任务线程A发送信号的时间t1，B线程收到信号后记录时间t2，则该次测试系统的信号延迟时间delta=t2-t1；

系统信号延迟测试包括信号延迟时间最大值和平均值。

测试过程需要持续一小时以上，信号发送接收次数超过一亿次。

推荐测试方案：

1. 测试工具：测试套件rt-tests中的signaltest工具；
2. 测试命令：signaltest -l 10000 -p80 -t2，其中：

——l 循环测试次数；

——p 优先级；

——t 测试线程数。

5.2.2 互斥锁延迟

互斥锁用于保护共享资源不被多个线程同时访问，避免数据竞争和不一致问题。互斥锁延迟过高可能导致任务执行延迟，影响系统的整体性能和响应时间。因此在智能驾驶操作系统性能测试中，互斥锁（Mutex）延迟是一个关键指标。

互斥锁延迟的最大值反映操作系统在一定的观测时间段内，获取互斥锁需要等待的最长时间，影响业务实时性。在系统设计、特别是设计确定性任务调度时，需要参考此值。

互斥锁延迟的平均值反映在一定的时间段内，操作系统处理的能力，作为设计参考。

5.2.2.1 技术要求

互斥锁延迟可能会影响关键进程的执行实时性、确定性，建议对系统互斥锁延迟进行测试，考虑到需要充分的量化互斥锁延迟对关业务的最大影响、平均影响，建议求出测试过程中的最大值、P95值和平均值。互斥锁延迟时间最大值应小于30 微秒。

5.2.2.2 测试方法

创建实时线程A,B，并使A先获得互斥锁，然后线程B等待线程A释放后获取互斥锁，通过时间获取函数记录A释放锁的时间t1,线程B获得互斥锁的时间t2，则该次测试的线程互斥延迟时间delta=t2-t1；测试过程需要持续一小时以上，互斥锁操作超过一亿次。

推荐测试方案：

互斥锁延迟测试内容包括延迟时间最大值和平均值。测试方法如下：

1. 测试工具：测试套件rt-tests中的ptsematest工具；
2. 测试命令：ptsematest -l 10000 -p80 -t1，其中：

——l 循环次数；

——p 优先级；

——t 线程对数。

5.2.3 消息队列延迟

消息队列是一种进程间通信的方式，允许进程之间传递消息。消息队列延迟是指从一个进程发送消息，到另一个进程成功接收消息的时间间隔。在智能驾驶操作系统性能测试中，消息队列延迟直接关系到系统实时性，以及多线程之间通信的能力，是一个关键指标。

消息队列延迟的最大值反映操作系统在一定的观测时间段内，消息从放入队列到被取出所经历的最长等待时间，反映了业务实时性情况，在系统设计、特别是设计确定性任务调度时，需要参考此值。

消息队列延迟的平均值反映在一定的观测时间段内，操作系统处理的能力，作为设计参考。

5.2.3.1 技术要求

消息队列延迟可能会影响关键进程的执行实时性、确定性，建议对系统消息队列延迟进行测试。考虑到需要充分的量化消息队列延迟对关业务的最大影响、平均影响和最小影响，建议求取测试过程中的最大值、P95值和平均值。消息队列延迟时间最大值应小于100 微秒。

5.2.3.2 测试方法

创建实时任务线程A和B，线程A通过消息队列发送消息给B，使用时间获取函数记录实时任务线程A发送消息的时间t1，B线程收到消息的时间t2，则该次测试系统的消息队列接收延迟时间delta=t2-t1。测试过程需要持续一小时以上，消息发送接收次数超过一亿次。

推荐测试方案：

消息队列延迟测试内容包括延迟时间最大值、延迟时间P95值和延迟时间平均值。测试方法如下：

1. 测试工具：测试套件rt-tests中的pmqtest工具；
2. 测试命令：pmqtest -l 10000 -p80 -t1，其中：

——l 循环次数；

——p 优先级；

——t 线程对数。

5.2.4 优先级反转

优先级反转是线程调度过程中的一种特殊现象。高优先级线程因为所需资源被低优先级进程占用而被阻塞，占用该资源的低优先级线程因其优先级低于其他进程也无法执行而释放资源，造成最高优先级进程反而在一段时间内无法执行，系统性能下降的情况。在智能驾驶操作系统性能测试中，优先级反转是一个关键指标，核心业务进程会因优先级反转而被低优先级任务阻塞，从而产生任务延迟和性能下降，导致执行不符合预期。

5.2.4.1 技术要求

系统应具备优先级继承能力，以避免优先级反转的情况。优先级反转延迟指低优先级线程持有资源切换到高优先级的时间，不考虑系统调度延迟的情况下，最大值应小于100微秒。

5.2.4.2 测试方法

从主线程创建三个调度优先级分别为低、中、高的三个工作线程A、B、C。线程 A先持有互斥量，而后线程B尝试持有互斥量，此时线程A的优先级应该被提升为和线程B的优先级相同。线程C检测线程A的优先级是否被提升到线程B的优先级。

推荐测试方案：

优先级反转测试的主要内容是测试系统中是否具备优先级继承功能。测试方法如下：

1. 测试工具：测试套件rt-tests中的pi\_stress工具；
2. 测试命令：pi\_stress --groups=4 --inversions=10000，其中：

——groups 测试组数（设置反转组的数量）；

——inversions 循环次数（每组反转数）。

5.2.5 共享内存延迟

共享内存（Shared Memory）是一种高效的进程间通信（IPC）的机制。它允许多个进程访问同一块物理内存区域，从而实现数据的快速共享和交换。

5.2.5.1技术要求

共享内存延迟最大值应小于100微秒。

5.2.5.2测试方法

创建1对父子进程，父进程并创建两个信号量（读信号量、写信号量），用户父子进程同步，再创建1个指定SIZE大小的共享内存区，得到共享内存区ID，通过共享内存区ID与父进程虚拟地址空间建立映射。再将该共享内存区域映射到子进程的虚拟地址空间，父进程向共享内存空间写入SIZE大小数据，记录时间t1,子进程读取SIZE大小数据，记录时间t2, 共享内存通信延迟deata=t2-t1。

推荐测试方案：

编写测试程序，建议数据大小SIZE选取128KB、256KB、512KB、1024KB、2048KB中的一个或者多个。

5.2.6 本地socket延迟

本地socket用于同一台主机的进程间通讯（IPC）。它不需要经过网络协议栈，不需要打包拆包、计算校验、维护序列号应答等，只是将应用层数据从一个进程拷贝到另一个进程。

5.2.6.1技术要求

本地socket延迟最大值应小于100微秒。

5.2.6.2测试方法

创建服务端进程和客服端进程，服务端进程调用socket函数创建1个socket描述符，并准备通信地址，对通信地址和socket描述符进行绑定（使用bind函数），客户端进程调用socket函数创建1个socket描述符，并准备通信地址，对通信地址和socket描述符进行连接（使用socket函数），客户端进程写入数据，记录时间t1，服务端进程读取数据，记录时间t2，本地socket通信延迟deata=t2-t1。

推荐测试方案：

使用开源测试工具lmbench，测试方法如下：

1. 测试工具：lat\_unix测试程序；
2. 测试命令：lat\_unix -P NUM –N 10000,其中:

——P 表示进行通信的CPU数量，建议设置为可用CPU数；

——N 表示测试次数，建议不小于10000次；

5.2.7 信号量延迟

信号量是一种重要的进程间同步和互斥机制，用于确保多个进程在访问共享资源时能够有序地执行。

5.2.7.1技术要求

信号量最大延迟应小于100微秒。

5.2.7.2测试方法

创建两个通过信号量同步的进程，并测量在一边释放信号量和在另一边获取信号量之间的延迟。

推荐测试方案：

使用开源实时性测试套件rt-tests，测试方法如下：

测试工具：svsematest测试程序；

1. 测试命令：svsematest -p 99 -t $NUM -i $INTV -D $TIME ,其中：

-p, --prio=PRIO 设置进程优先级；

-t，--threads[=NUM] 设置测试线程的数量，建议设置为可用CPU数；

-i，--interval=INTV 设置线程的基本间隔；

-D，--duration=TIME 设置测试时长，建议不低于1小时。

* + 1. 中断

中断测评指标包括中断延迟时间和中断运行时间。

5.3.1 中断延迟时间

中断延迟时间是指从硬件中断信号产生到操作系统开始执行对应中断服务程序的时间间隔，直接关系到系统中断响应的实时性，反映了对外部事件的响应速度，是一个关键指标。

中断延迟的最大值和P95值反映操作系统在一定的观测时间段内，系统的中断响应最长延迟时间。

中断延迟的平均值反映在一定的时间段内，操作系统处理的能力，仅供设计参考。

5.3.1.1 技术要求

中断延迟可能会影响系统执行的实时性、确定性，建议对系统中断延迟进行测试。考虑到需要充分的量化中断延迟对关业务的最大影响、平均影响，建议求取测试过程中的最大值、P95值和平均值。中断延迟时间最大值应小于500 微秒。

5.3.1.2 测试方法

设置定时器中断，设置中断期望产生时间t1，等待定时器中断产生并进入中断处理函数记录时间t2，计算中断延迟为t2-t1。

推荐测试方案：

1. 内核模块在内核态创建每个CPU的高精度定时器，定时周期设定为t；
2. 在启动定时器时，调用时间获取函数获取当前时间t1,在每个CPU定时器回调处理函数中，获取当前时间t2，则每个CPU的中断延迟时间为delta = t2 - t1 - t；
3. 测试时间：一小时;
4. 获取所有中断延迟时间，计算最大值、P95值和平均值。

5.3.2 中断运行时间

中断运行时间是指操作系统从开始中断服务程序到中断服务程序处理完毕的执行时间，它直接关系到系统响应的实时性，太长时间的硬中断、软中断运行时间会对系统实时性产生影响，是一个关键指标。如果出现太长中断运行时间，需要考虑优化中断处理程序、延后处理中断等优化方案。

中断运行时间的最大值反映操作系统在一定的观测时间段内，系统的中断服务程序单次执行的最长时间，对系统实时性影响很大。

中断运行时间的平均值反映在一定的时间段内，反映了操作系统处理的能力，作为设计参考。

5.3.2.1 技术要求

中断运行时间可能会影响系统执行的实时性、确定性，考虑到需要充分的量化中断延迟对关业务的最大影响、平均影响，建议求取测试过程中的最大值、P95值和平均值；

根据经验和实际测试，硬件中断函数执行时间指标值应小于调度延迟，最大值不应该超过1毫秒。线程化中断运行时间由用户决定。

5.3.2.2 测试方法

记录硬中断、软中断和定时器处理函数开始和结束的时刻，通过时间获取函数分别获取当前时间t1和t2，则本次中断处理的时间delta=t2-t1。测试过程需要持续一小时以上。

推荐测试方案：

1. 按照测试方法，在内核模块通过在硬中断、软中断和定时器处理函数开始和结束的时刻，通过时间获取函数分别获取当前时间t1和t2，则本次中断处理的时间delta=t2-t1；
2. 测试时间：一小时；
3. 获取所有中断运行时间。
4. 计算中断运行时间的最大值和P95值。
   1. 计算性能技术要求与测试方法
      1. CPU性能

CPU性能主要是对CPU综合处理能力的评估，涵盖了整型计算、浮点计算、分支预测、缓存效率等多个方面的性能。本规范包含整型（定点）性能和浮点性能测试。

整型性能指的是对整数进行的算术运算，通常包括加法、减法、乘法和除法等。整数可以是有符号的或无符号的，表示范围和大小取决于所用的数据类型。整型计算在计算机中非常普遍，许多程序和算法都涉及到大量的整数操作。

浮点性能涉及到实数（即带有小数点的数值），用于表示更大范围的数值，因为计算机只能存储整数，所以实数都是约数。浮点数的计算比整数计算更为复杂，需要处理数值的符号、指数和尾数三个部分，所以计算要慢而且会有误差。

6.1.1 技术要求

CPU算力大于 10K DMIPS。

6.1.2 测试方法

数学运算测试方法如下：

1. 测试开始，记录开始时间Ts；
2. 执行N次相应的数学运算（执行过程中不释放cpu）；
3. 测试结束，记录结束时间Te；
4. 计算一次数学运算的平均执行时间t=（Te-Ts）/ N。

数学运算包括整型数的位运算、加、乘、除、取模和浮点数的加、乘、除及混合运算。

为了避免测试误差，需要多次测试并计算其平均性能。

测试时间不小于一小时。

推荐测试方案：

根据实际硬件配置，在CPU基准测试工具中按情况挑选相应工具进行测试：

1. 如车用硬件配置低（如：CPU为ARM M或者R核，内存不超过4G），可使用 Nbench、Linpack等工具进行评测；
2. Linpack 只关注浮点性能。
3. 如硬件配置较高（如：内存超过4G，硬盘大于128G），可选用SPECCPU 等benchmark工具进行评测。
   * 1. 内存性能

操作系统内存性能指标主要关注内存带宽、内存读取延迟、内存申请耗时、内存释放耗时等。

内存带宽是指系统中可持续运行的内存带宽的最大值，通常以每秒字节数（MB/s或GB/s）表示，其直接影响CPU从内存中获取数据的速度。内存读取延迟是指CPU与内存之间的读取延迟。内存申请耗时是指应用执行内存分配函数耗时（有多种场景，高中低负载、只分配虚拟内存/虚拟及物理内存一次分配，水位线影响等）。内存释放耗时是指应用执行内存释放函数耗时。内存碎片率用来衡量内存使用效率，反映了内存中未被充分利用的空间比例。

6.2.1 技术要求

通过单线程对8KB~8MB数据进行malloc/free和new/delete操作，评测内存申请耗时和内存释放耗时内存申请耗时<200纳秒，内存释放耗时<200纳秒，内存碎片率<75%。

6.2.2 测试方法

**（1）内存带宽**

在智能驾驶域控制器上运行测试工具stream获取带宽值，注意保证STREAM\_ARRAY\_SIZE > 4 \* L3 Cache。

**（2）内存延迟**

运行lmbench中的内存延迟测试工具lat\_mem\_rd，测试不同内存大小与跨度下的内存读取延迟，输出结果单位为纳秒。

注：内存访问延迟实际与CPU的缓存有关，包括一级缓存、二级缓存与三级缓存，当访存大小低于缓存时，可能是直接从缓存中读取的，因此需要重点关注访存大小大于缓存时的内存延迟情况。建议内存访问前先做cache flush。

**（3）内存申请耗时**

在调用库函数malloc时，记录时间为T1，函数完成时，记录时间为T2，内存申请耗时 delta = T2-T1。

测试时间：一小时。

**（4）内存释放耗时**

在调用库函数free时，记录时间为T1，函数完成时，记录时间为T2，内存申请耗时delta = T2-T1。

测试时间：一小时。

**（5）内存碎片率**

通过/proc/buddyinfo中的数据分析内存碎片率，评测内存碎片率。

推荐测试方案：

1. 内存运行带宽，使用stream；
2. 内存延迟，运行lmbench中的内存延迟测试工具lat\_mem\_rd；
3. 其他需要编写程序或者专用工具。
   1. 通信性能技术要求与测试方法
      1. I/O性能

系统根据I/O请求将数据读出或写入目标磁盘的过程涉及多个组件，其处理效率与硬件配置、内存管理、I/O栈、设备驱动和磁盘特性密切关系。I/O性能可通过每秒操作数、操作延迟和吞吐量三个通用指标来衡量。

**（1）每秒操作数 operations/second**

应用程序可并发地通过各种文件接口向系统提交各类io请求, 如文件读写、目录读写等，文件系统为保证接口在posix要求范围内的一致性，对数据和元数据的读写做了限制。根据数据元数据的布局及碎片情况，以及目标媒介的访问特性，特定类型的文件系统在有限系统资源下单位时间内对各类请求的处理个数存在上限。每秒操作数可体现系统对I/O操作请求的响应能力，影响IO密集型任务的并发能力。

**（2）操作延迟 return\_time - invoke\_time**

与每秒操作数相对应，不同的操作所需要的处理不同，如文件写入或删除需要对元数据进行修改，需要从空闲块中做分配或回收动作，同时根据可靠性要求还可以伴随着日志写入，因此完成处理需要的时间也不相同。操作延迟对应于操作请求的响应速度，影响I/O密集型任务的程序响应速度。

**（3）数据吞吐量 bytes/second**

接收到I/O请求后，根据请求类型、数据大小、文件大小、访问模式、一致性要求、缓存策略和文件系统数据组织特点，并同时受I/O调度、队列深度等各因素影响，最终单位时间内从文件系统读出或写入的数据量代表了系统对I/O请求的服务能力。数据吞吐量影响I/O密集型任务的业务吞吐量。

7.1.1 技术要求

测试应覆盖4B、8B、16B、32B、64B、128B、256B、512B、1KB、2KB、4KB、8KB和16KB等大小数据块，顺序读写，随机读写等操作的IOPS（每秒操作数）、latency（延迟）和throughput（吞吐量）。硬件和测试场景等因素影响大，不对数值做具体要求。

7.1.2 测试方法

**（1）I/O每秒操作数**

通过使用不同大小的记录来读写不同大小的临时文件，并使用不同的访问顺序，如初次读写、重复读写、反向读写、随机读写和跳跃读取来操作文件，最终输出不同记录长度下的每秒操作次数。

推荐测试方案：**通过iozone进行IOPS测试**

iozone -a -O

-a 自动进行多种访问类型的操作，如顺序读写，反向读写等。

-O 指定输出结果为每秒操作数。

**（2）I/O延迟测试**

通过使用不同大小的记录来读写不同大小的临时文件，并使用不同的访问顺序 ，如顺序读、顺序写、随机读、随机写和混合读写I**/**O等来操作文件，最终输出包含I**/**O延迟信息(请求从提交到完成)的统计数据。

推荐测试方案：**通过fio进行io延迟测试**

fio -filename=/tmp/test\_randread -direct=1 -iodepth 1 -thread -rw=randread -ioengine=psync -bs=16k -size=200M -numjobs=2 -runtime=60 -group\_reporting -name=mytest

-filename 测试文件名称。

-direct 读写操作是否绕过系统缓存。

-iodepth 同时提交请求的个数上限。

-thread 并发测试时使用多个进程还是多个线程。

-rw 读写类型，可以为顺序读，顺序写，随机读，随机写，混合读写。

-ioengine 提交请求的方式，sync, psync, libaio, io\_uring, mmap等。

-bs 每次请求时的数据块大小。

-size 文件大小。

-numjobs 同时进行测试的线程个数。

-runtime 执行时间。

该示例输出中的lat行可以给出当前操作模式下请求从提交到完成的时间分布。

**（3）I/O数据吞吐量**

通过使用不同大小的记录来读写不同大小的临时文件，并使用不同的访问顺序，如初次读写、重复读写、反向读写、随机读写和跳跃读取来操作文件，最终输出每个线程的每秒读写数据量。

推荐测试方案：**通过iozone进行吞吐量测试**

iozone -t 2 -s 512 -r 4

-t 指定同时测试进行的线程数量,

-s 指定文件大小，

-r 指定记录大小。

* + 1. 网络性能

系统需要为网络拓扑等网络配置信息维护状态，为网络连接建立多元组信息并管理协议状态，其处理效率与硬件配置、调度、内存管理、网络栈、设备驱动和网卡特性密切关系。网络性能可通过数据吞吐量、丢包率、延迟和网络抖动四个通用指标来衡量。

（1）数据吞吐量 packets/second

根据套接字是否面向连接及是否有顺序要求，收发数据的大小、收发数据的内存布局、收发频率、网络状态、以及所有中间节点的软硬件处理能力都影响网络的最终吞吐量，即单位时间内通过本机的入站出站流量。

（2）丢包率 lost\_packets/send\_packets

若接收流量超过节点处理能力或有异常，如并发连接超过上限、内存不足、网络拥塞、路由故障等，系统在尽最大努力交付整机流量原则下，将随机丢弃一部分网络包，因此使用丢包率来表示丢弃包在发送总量所占比例。丢包率达高时将影响服务可用性及用户体验。

（3）延迟 rx\_time - tx\_time

当网络包从发送到接收，经过本地处理、中间节点和对端处理后所经过的总时间。除了正常处理所耗时间外，可能由于本机协议状态发生变化，本机负载变化，本机资源不足导致的重试操作，中间节点的排队处理等各因素影响，会导致延迟在不同情况下发生变化。 当延迟过大时甚至会影响服务可用性。

除此之外，为了提高网络的实时性能，建议禁用TCP协议中的Nagle算法。为了加快网络数据包的处理速度，在保证安全性与可靠性的前提下，推荐使用DPDK与XDP等新兴内核旁路技术。

7.2.1 技术要求

测试应覆盖64B、128B、256B、512B、1KB、2KB、4KB、8KB、16KB和32KB等大小数据包，TCP、UDP传输的latency（延迟）、throughput（吞吐量）和丢包率等。硬件和测试场景等因素影响大，不对数值做具体要求。

7.2.2 测试方法

（1）数据吞吐量

分为TCP协议吞吐量以及UDP协议吞吐量。

使用测试工具或程序在指定的一段时间内，按照设定的速率（如每秒发送数据包的数量或速率比特/秒）连续不断地向网络发送数据包。

在另一端，使用测试工具或接收端设备会捕获这些数据包，并记录接收的数据量。

测试工具或程序会统计发送和接收的数据包数量以及它们的内容大小，并计算在测试时间段内成功传输的总数据量。

（2）丢包率

需要连续测试一小时。

通过测试工具向目标地址发送一定数量的数据包，在接收端记录接收到的数据包数量，用发送数据包总数减去接收成功的数据包数，再除以发送数据包总数，得到丢包率。为了得到更准确的评估结果，通常需要在一段时间内连续发送并监测数据包的接收情况，以获取在不同网络负载条件下的丢包率变化。

（3）延迟

网络延迟包括TCP连接延迟以及TCP往返时间。TCP连接延迟即发送端从三次握手的SYN数据包发出，到响应数据包的时间。TCP往返时间即发送端从发送一个TCP数据包到接收方，并接收到该数据包的确认信息所耗费的时间。

（4）网络抖动

初始测试环境网络负载50%；

通过发送连续的数据包，并在发送端和接收端分别记录每个数据包的发送和接收时间戳。对于每一个接收到的数据包，计算其相对于期望到达时间（基于理想情况下均匀间隔发送的假设）的时间偏差，也就是延迟的变动幅度。统计所有数据包的延迟变化值，计算抖动的平均值、最大值、最小值和方差等统计参数，从而评估网络的抖动特性。

推荐测试方案：

1. 选取两台配置相同设备直连；
2. 运行Iperf、Netperf、ping等网络性能测试工具运行测试，收集结果。
3. 为保证测试一致性，建议在下述测试环境中进行测试：

环境部分：

•环境温度：23℃±5℃（73.4℉±9℉）

•环境湿度：50±20%

设备连接部分：

准备一台运行Linux系统的测试设备Tester，将智能驾驶域控制器与Tester通过千兆以太网连接，两者处于同一局域网下，保证网络连通性功能正常。在域控制器与Tester上分别安装Iperf3、ping、BCC等性能测试工具。

**（1）数据吞吐量**

采用iperf3进行网络吞吐量测试，主要分为TCP协议吞吐量以及UDP协议吞吐量。具体方法如下：

•Tester作为服务端，运行如下指令：

iperf3 –s

其中，

-s:以Server模式运行

•在测试TCP协议吞吐量时，智能驾驶域控制器中运行如下指令

iperf3 –c <server\_IP\_address> -b <bitrate> -4 –O 5 –i 1 –t 600 -N

其中，

-c: 以Client模式运行

server\_IP\_address：Tester端的IP地址

-b：限制测试带宽,bitrate根据域控制器硬件参数进行定义

-4: 仅适用IPv4协议

-O：忽略前n秒的测试，上述命令为忽略前5秒

-i: 每次报告的时间间隔，上述命令为1秒

-t: 测试时间，上述命令为一小时

-N：屏蔽Nagle算法

•在测试UDP协议吞吐量时，智能驾驶域控制器中运行如下指令

iperf3 –c <server\_IP\_address> -u -b <bitrate> -4 –O 5 –i 1 –t 600

其中，

-c: 以Client模式运行

server\_IP\_address：Tester端的IP地址

-u：采用UDP协议进行传输

-b：限制测试带宽,bitrate根据域控制器硬件参数进行定义

-4: 仅适用IPv4协议

-O：忽略前n秒的测试，上述命令为忽略前5秒

-i: 每次报告的时间间隔，上述命令为1秒

-t: 测试时间，上述命令为一小时

通过上述测试，获取TCP与UDP的吞吐量值，单次测试时间为一小时。

**（2）网络抖动与丢包率**

采用iperf3进行UDP协议网络抖动与丢包率，初始测试环境网络负载50%。采用命令如下：

iperf3 –c <server\_IP\_address> -u -b <value> -4 –O 5 –i 1 –t 600

其中，

-c: 以Client模式运行

server\_IP\_address：测试设备端的IP地址

-u: 采用UDP协议进行传输

-b：限制测试带宽,value值为用户自定义

-4: 仅适用IPv4协议

-O：忽略前n秒的测试，上述命令为忽略前5秒

-i: 每次报告的时间间隔，上述命令为1秒

-t: 测试时间，上述命令为一小时

通过查看输出结果中的Jitter了解网络抖动情况，查看域控制器中iperf3输出结果中的Lost/Total Datagrams了解丢包率。单次测试时间为一小时。

**（3）网络延迟**

网络延迟包括TCP连接延迟以及TCP往返时间。TCP连接延迟即发送端从三次握手的SYN数据包发出，到响应数据包的时间。TCP往返时间即发送端从发送一个TCP数据包到接收方，并接收到该数据包的确认信息所耗费的时间。

TCP连接延迟的具体测试方法如下：

•在智能驾驶域控制器上运行TCP连接建立测试脚本，测试时间一小时。

•在智能驾驶域控制器端进行连接延迟测量。智能驾驶域控制器端发送SYN包时记录时间为T1，域控制器接收到测试设备端发送的SYN-ANK时记录时间为T2，TCP连接延迟时间为delta = T2-T1

测试工具上，可以采用BCC工具中的tcpconnlat，如下：

Tcpconnlat –p <PID> -t

其中，

-p：测试脚本的PID

-t： 在输出数据中添加时间戳

查看工具输出数据中的LAT一栏，即可获取本次TCP连接的连接延迟。

TCP往返时间的具体测试方法如下：

•在智能驾驶域控制器与Tester之间创建持续的TCP数据流量。

•在智能驾驶域控制器端进行TCP往返时间测量。域控制器端发送TCP数据包时记录时间为T1，域控制器接收到该数据包的确认信息时记录时间为T2，TCP往返时间为delta = T2-T1

在具体的测试工具上，可以选择iperf3与BCC中的tcprtt工具，如下：

•在智能驾驶域控制器以及Tester上运行iperf3，创建大量的TCP数据包，操作如下：

Tester端指令如下：

iperf3 –s –p <port>

其中，

-s: 以Server模式运行

-p: 指定服务端端口

智能驾驶域控制器上运行如下指令。

iperf3 –c <server\_IP\_address> -p <port> -4 -t 600 -N

其中，

-c: 以Client模式运行

server\_IP\_address：Tester端的IP地址

-p: 服务端端口

-4: 仅适用IPv4协议

-t: 测试时间

-N：屏蔽Nagle算法

•在智能驾驶域控制器上运行BCC工具中的tcprtt，进行TCP往返时间的测量，如下：

Tcprtt -4 -m –T –e –i 30 –d 5

其中，

-4：仅统计IPv4

-m：往返时间单位为毫秒

-T： 时间戳

-e： 输出平均值

-i： 单次直方图统计时间

-d: 统计次数

查看tcprtt输出数据，即可了解当前环境下的TCP往返时间的直方图分布情况。

* + 1. 时钟

操作系统通过时间来计算系统的调度和延迟，操作系统的时间基于时钟来计算，系统延迟函数精度决定了操作系统调度的精度，可以通过测试延迟函数的精度，来度量系统对于纳秒级延迟操作的精度。

nanosleep的延迟精度受系统调度方式、各种锁关闭时间的影响，可以设置延迟时间为1微秒，nanosleep看系统延迟函数精度。

7.3.1 技术要求

系统延迟函数精度（分辨率）<100纳秒。

7.3.2 测试方法

系统延迟函数精度

（1）获取系统时间；

（2）调用延迟函数nanosleep，设置延迟时间为固定值（1ns，1μs），查看系统延时值是否准确；

（3）延迟结束后，获取系统时间；

（4）计算延迟差值。

推荐测试方案：

通过"/proc/timer\_list "文件中的 ".resolution" 项来确认高精度时钟的同步时间精度。

判读同步时钟精度是否满足要求：分辨率是否在100纳秒内。

* + 1. CAN通讯性能

CAN通讯延迟反映操作系统应用数据到CAN驱动硬件的延迟时间，基于CAN的应用功能对该时间比较敏感。最大延迟时间影响CAN组网报文的实时性，导致报文时间不准确，进而影响CAN总线上层的功能，如网络管理的建环功能、应用报文车的数据更新不及时导致仪表显示延迟等。

CAN通讯延迟最大延迟：应用数据到硬件总线的最大时间，包括描述应用发包到总线的最大时间的上行最大延迟，描述总线包到应用的最大时间的下行最大延迟。

BUSOFF恢复时间，为BUSOFF状态到恢复正常状态的时间，该参数反映操作系统对CAN自恢复的机制支持，CAN组网功能对该参数强需求，且基于CAN的应用功能对该时间比较敏感。BusOFF状态触发源:包含不限于CAN\_H与GND短路、CAN\_L与GND短路、CAN\_H与Batterry短路、CAN\_L与Batterry短路、CAN\_H与CAN\_L短路。BUSOFF恢复时间，如果时间太大，会导致整个基于CAN的应用功能部分时间失效或者延迟较大。

最大网络负载率，操作系统的CAN通信能达到的最大负载，反映操作系统对CAN的处理能力，CAN的应用功能对该时间比较敏感。最大网络负载率如果太小，会出现当整个CAN网络负载大的情况下，报文不能正常收发。

CAN通信最大网络负载率：每秒传输位数/比特率\*100%，包括：

上行负载率：CAN发送的负载率；

下行负载率：CAN接收的负载率。

CAN通信错误帧统计，反映操作系统对CAN的处理能力，基于CAN的应用功能对故障诊断对该参数比较敏感，错误帧统计主要解决硬件故障，通过识别来定位故障的位置，如果不能识别会导致排查故障的时候难以下手。

CAN通信错误帧统计：位错误、ack错误、填充错误、crc错误、格式错误5种错误。

CAN帧错误率，反映操作系统对CAN的处理能力，基于CAN的应用功能对该时间比较敏感，错误率太高，会影响上层基于CAN的应用功能。

CAN帧错误率：单位时间错误包数/单位时间帧数\*100%。

7.4.1 技术要求

判断CAN通信性能是否满足要求：Loss < 0.1%， 最大延迟 < 10毫秒；

最大延迟：带有网络管理组网的ecu上行小于5毫秒 ，下行小于5毫秒；

带有uds诊断协议栈的ECU上行小于5毫秒 ，下行小于5毫秒；带应用报文通信的ecu小于1毫秒 ，下行小于1毫秒；

BUSOFF恢复时间：小于300毫秒；

最大网络负载率：上行 100%，下行100%；

错误率：小于0.1%；

错误帧统计：包含不限于位错误、ack错误、填充错误、crc错误、格式错误等5种错误。

7.4.2 测试方法

**（1）最大延迟**

a.上行：使用CAN调试器与设备通过CAN连接，设备向调试器发报文，发报文前读取定时器值T1，调试发送后读取发送状态，如果发送成功再次读取定时器值T2，发送延迟Td = T2-T1，多次发送，统计延迟最大值TdMax。

b.下行：使用CAN调试器与设备通过CAN连接，调试器向设备发报文，接收中断读取定时器值T1，应用读取到报文读取定时器值T2，接收延迟Td = T2-T1，多次发送，统计延迟最大值TdMax。

**（2）BUSOFF恢复时间**

使用CAN调试器与设备通过CAN连接，设备定时发包，使用CAN\_H与GND短路等方式使设备进入CANbusoff，然后看调试器上面定时报文恢复的时间。

**（3）最大负载率**

a.上行：使用CAN调试器与设备通过CAN连接，设备逐渐增大发包频率，看CAN调试器上面显示能达到最大不丢帧的负债率。

b.下行：使用CAN调试器与设备通过CAN连接，CAN调试器逐渐增大发包频率，看设备能显示达到最大不丢帧的负债率。

**（4）错误率**

使用CAN调试器与设备通过CAN连接，设备向调试器，调试器向设备双向定时发报文，统计每秒错误帧数与发送的总帧数x100%。

1. **错误帧统计**

使用CAN调试器与设备通过CAN连接，设备向调试器，调试器向设备双向定时发报文，触发对应的帧错误帧，看设备是否正常识别。

注：使用CAN调试器与设备通过CAN连接如CANoe。CAN调试器上位机自带错误率、错误帧、网络负载率统计，收集结果。

* + 1. UART通信性能

7.5.1 技术指标

单次数据收发量128字节，测试数据收发1000次，数据丢包率为0，且前后收发数据准确、前后一致。

7.5.2 测试方法

推荐测试方案：使用串口工具进行数据收发测试，对比收发前后的数据。测试中选择1~2种项目中使用到的波特率，比如2400-921600bps。

* + 1. SPI通信性能
       1. 技术指标

单次数据读写不小于128字节；

数据传输准确，主从机之间数据收发准确、前后一致。

* + - 1. 测试方法

推荐测试方案：编写测试代码进行master-slave之间的数据收发回环测试，可以对maste模式和slave模式进行同时的数据收发测试。

测试次数不小于1000次；应测试项目所需的速率，建议采用10MHz。

* + 1. PCIe通信性能
       1. 技术要求

通信带宽不低于理论值的70%（如理论值为 1000M/s，实测值不低于700M/s）。

* + - 1. 测试方法

推荐测试方案：编写测试代码，在SOC间进行数据传输测试，SOC1做为ep端，设置为写模式，SOC2做为rc端，设置为读模式。ep端写入指定大小数据SIZE，触发MSI中断，记录时间t1，rp端读取数据，记录时间t2，通信带宽即为：SIZE/(t2-t1)。

* + 1. I2C通信性能
       1. 技术要求

从设备寄存器配置1000次，配置成功率100%；

I2C模式建议配置为fast mode，速率配置为100K或400K。

* + - 1. 测试方法

推荐测试方案，编写测试代码：

1. 确认I2C节点生成，I2C总线初始化成功；
2. 使用I2Cdetect 确认总线上的设备挂载成功；
3. 使用 I2Cset设置从设备寄存器；
4. 使用I2Cget或者I2Cdump检查从设备寄存器的值；
5. 使用示波器测试I2C的时钟引脚频率。
   1. 智驾相关性能技术要求与测试方法
      1. 启动时间

操作系统启动快慢是智驾操作系统的一个重要指标，智驾操作系统期望能够快速的达到控制驾驶系统以及实现与用户交互的效果。

操作系统启动越快，也代表着能和用户尽快的进行交互和控制。

操作系统内核启动时间主要是指从操作系统启动到系统可登陆（命令行出现）的时间。操作系统内核启动过程中，会加载很多驱动，加长整体的启动时间，这部分也算在内，系统可登陆后，系统的关键进程就可以启动。

系统的关键进程启动时间，是指从内核启动开始，到系统关键进程准备好的时间，关键进程和业务相关，不同的系统也不一样，此指标为参考信息，但是对具体的业务有很好的指导意义。

8.1.1 技术要求

内核启动完成时间 < 5秒；

关键进程启动完成时间 < 8秒；

对于关键进程不同的系统会不一样，需要记录测试的关键进程信息。

8.1.2 测试方法

通过串口Log输出估算内核启动完成时间及关键进程启动完成时间。

推荐测试方案：

1. 配置好开发板的串口输出并配置时间戳输出功能；
2. 开始以下操作，并循环10次：
3. 下电后并清除串口内容；
4. 记录上电后确认串口第一条Log对应的时间戳 T(1)；
5. 记录上电后内核启动完成的Log对应的时间戳 T(2)；
6. 记录上电后关键进程启动完成的Log对应的时间戳 T(3)；
7. 计算内核启动完成时间为：△T(内 )= T(2 )- T(1)；
8. 计算关键进程启动完成时间为：△T(关) = T(3) - T(1)；
9. 对以上的△T(核)、△T(关)取平均值即为内核启动完成时间和关键进程启动完成时间。
   * 1. 通信中间件性能

通信中间件性能主要包括SOMEIP性能和DDS性能。

SOMEIP（Scalable service-Oriented MiddlewarE over IP）是一个基于IP的可扩展面向服务的中间件，主要用于汽车电子领域的通信。SOMEIP性能包括通信延迟平均值和最大值。

DDS（Data Distribution Service for Real-Time Systems）是一种针对实时系统设计的数据分发服务规范，它提供了一种可靠的、实时的数据共享机制。DDS性能包括通信延迟平均值和最大值。

8.2.1 技术要求

SOMEIP性能：通信延迟最大值<50毫秒，通信延迟平均值<25毫秒。

DDS性能：通信延迟最大值<100毫秒，通信延迟平均值<50毫秒。

8.2.2 测试方法

SOMEIP测试方法如下：

1. 统计someip client-server（通信数据≤4Mb）的通信时间，包含最大值和平均值。
2. 判断someip性能是否满足要求。

DDS测试方法如下：

1. 统计DDS的pub server-sub client（通信数据≤4Mb，发送周期为50ms）的通信时间，包含最大值和平均值。
2. 判断DDS性能是否满足要求。
   * 1. 摄像头性能

摄像头性能是涵盖了摄像头硬件、操作系统、和应用软件的整体性能，对于操作系统来说，主要是图像的采集与处理和以及传输图像的功能，对于图像处理的效果和硬件的能力关系更大，考虑这些因素，对于操作系统来说摄像头性能主要考虑图像分辨率、帧率和帧传递时间抖动。

图像分辨率和格式决定了每帧图像的大小，图像的帧率是每秒能处理的图像帧数，每秒处理的数据量和图像分辨率大小及帧率成正比。

CPU处理图像期望还是等间隔的，帧传递时间抖动的最大值和平均值反映了处理图像的时间均匀度，也反映了图像的处理能力。

8.3.1 技术要求

测量图像分辨率、帧率和帧传递时间抖动，同时需要记录图像格式、CPU的利用率和内存使用情况。

图像分辨率≥1920×1080；帧率≥20fps；帧传递时间抖动<15毫秒。

8.3.2 测试方法

分辨率、帧率测试方法如下：

1. 通过存图或者配置文件获取每个摄像头的分辨率；
2. 通过log文件或者配置文件获取每个摄像头的帧率；判断分辨率和帧率是否满足要求。
3. 帧传递时间抖动，通过帧率计算每帧的间隔时间t，记录一帧和前一帧的时间差t1，则帧传递时间抖动值为t1-t，记录观测时间内的帧传递时间抖动，然后计算最大值和平均值。