尊敬的浪潮存储系统用户：

衷心感谢您选用了浪潮存储系统！

本手册介绍了浪潮存储系统的技术特性与系统的安装、设置和使用，有助于您更详细地了解和便捷地使用本款存储系统。

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，造福人类。

本手册中的内容如有变动恕不另行通知。

如果您对本手册有疑问或建议，请向浪潮电子信息产业股份有限公司有限公司垂询。

浪潮电子信息产业股份有限公司

2013 年 8 月

声　　明

在您正式使用本存储系统之前，请您先阅读以下声明。只有您阅读了以下声明并且同意以下各条款后，方可正式开始使用本存储系统；如果您对以下条款有任何疑问，请和您的供货商联系或直接与我们联系。如您未向我们就以下条款提出疑问并开始使用本系统，则是默认您已经同意了以下各条款。

1、我们提醒您特别注意：在任何时候，除了我们提示您可以修改的参数以外，您不要修改本存储系统主板 BIOS 中的任何其他参数。

2、在您使用的存储系统出现任何硬件故障或您希望对硬件进行任何升级时，请您将机器的详细硬件配置反映给我们的客户服务中心；您不要自行拆卸存储系统机箱及机箱内任何硬件设备。

3、本存储系统的内存、CPU、CPU 散热片、风扇、硬盘托架、硬盘等都是特殊规格的，请您不要将它们和任何其他型号机器的相应设备混用。

4、您在使用存储系统过程中遇到的任何软件问题，我们希望您首先和相应软件的提供商联系，由他和我们联系，以方便我们沟通、共同解决您碰到的问题。对于如数据库、网络管理软件或其他网络产品等的安装、运行问题，我们尤其希望您能够这样处理。

5、如果上架安装本存储系统，请先仔细阅读相关产品手册中的快速安装指南。浪潮致力于产品功能和性能的持续提升，这可能导致部分功能及操作与手册描述有所差异，但不会影响使用，如果您有任何使用疑难问题，请与我们的客户服务中心联系。

6、我们特别提醒您：在使用过程中，注意对您的数据进行必要的备份。

7、此为A级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，可能需要用户对其干扰采取切实可行的措施。

8、请仔细阅读并遵守本手册的安全细则。

9、本手册中涉及的各软、硬件产品的标识、名称版权归产品的相应公司拥有。

10、以上声明中，“我们”指代浪潮电子信息产业股份有限公司有限公司；浪潮电子信息产业股份有限公司有限公司拥有对以上声明的最终解释权。

安全细则

1、本系统中的电源设备可能会产生高电压和危险电能，从而导致人身伤害。请勿自行卸下主机盖,并拆装、更换系统内部的任何组件，除非另外得到浪潮的通知，否则只有经过浪潮培训的维修技术人员才有权拆开主机盖及拆装、更换内部组件。

2、请将设备连接到适当的电源，仅可使用额定输入标签上指明的外部电源类型为设备供电，为保护您的设备免受电压瞬间升高或降低所导致的损坏，请使用相关的稳压设备或不间断电源设备。

3、如果必须使用延长电缆，请使用配有正确接地插头的三芯电缆，并查看延长电缆的额定值，确保插入延长电缆的所有产品的额定电流总和不超过延长电缆额定电流限制的百分之八十。

4、请务必使用随机配备的供电组件如电源线、电源插座（如果随机配备）等，为了设备及使用者的安全，不要随意更换电源电缆或插头。

5、为防止系统漏电造成电击危险，务必将系统和外围设备的电源电缆插入已正确接地的电源插座。请将 3 芯电源线插头插入接地良好、伸手可及的 3 芯交流电源插座中，务必使用电源线的接地插脚，不要使用转接插头或拔下电缆的接地插脚，在未安装接地导线及不确定是否已有适当接地保护的情况下，请勿操作使用本设备，可与电工联系咨询。

6、切勿将任何物体塞入系统的开孔处。如果塞入物体，可能会导致内部组件短路而引起火灾或电击。

7、请将系统置于远离散热片和有热源的地方，切勿堵塞通风孔。

8、切勿让食物或液体散落在系统内部或其它组件上，不要在高潮湿、高灰尘的环境中使用产品。

9、用错误型号的电池更换会有爆炸危险，需要更换电池时，请先向制造商咨询并使用制造商推荐型号相同或相近的电池，切勿拆开、挤压、刺戳电池或使其外部接点短路，不要将其丢入火中或水中，也不要暴露在温度超过 60 摄氏度的环境中，请勿尝试打开或维修电池，务必合理处置用完的电池，不要将用完的电池及可能包含电池的电路板及其它组件与其它废品放在一起，有关电池回收请与当地废品回收处理机构联系。

关于本手册

第一章 系统介绍

介绍浪潮存储系统的配置规格、功能及系统特性。使您对浪潮存储系统有一个大致的了解。

第二章 配置准备

详细介绍浪潮海量存储系统软件的操作之前的准备工作，使您能够迅速熟悉该系统的操作，充分利用该系统的强大功能。

第三章 系统配置

对浪潮存储系统在使用过程的具体操作步骤给予明确说明。以便您在使用过程中遇到任何问题能够方便快速的查找解决。

我们建议您在使用浪潮存储系统之前仔细阅读本手册，以避免您在操作中出现失误。

手册中难免存在细节上的不足，希望您能够包涵，并及时给我们批评指正。

技术服务电话： 86-531-88546554

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有限公司

邮　　编： 250101

目　录

第一章　系统介绍

浪潮集群存储系统 AS13000 是一款高度模块化，面向中高端存储应用需求的存储平台，它具有高可靠性、高可扩展性、高性能以及简洁易用的管理界面，可以满足高性能计算、数据库、网站、文件服务、流媒体、数字化视频监控、文件备份等领域的应用。

1.1 系统特点

1．异构兼容性

系统支持多种业界标准传输协议，包括 NFS、CIFS 以及 HTTP 等，提供对 Windows、Linux、UNIX 等所有主流操作系统的兼容性支持，并保证不同平台和操作系统数据视图的一致性，这极大的简化了用户的管理和维护操作，使用户无需再关心复杂的存储连接、访问权限设计。

2．高性能

优化的网络传输协议保证系统取得高带宽利用率及线性性能增长。

系统的各个数据节点通过 InfiniBand/10GbE 网络互连，采用标准的 TCP/IP 网络传输协议，实现端到端的吞吐性能提升，QDR IB 网络环境下吞吐量可达 1GB/s。

系统的分布式架构确保标准存储节点和元数据节点无瓶颈，文件多节点并发访问。

3．可扩展性

系统具备灵活的扩展能力可很好应对增长型企业业务持续增长带来的数据和性能的增长。主要从下面两方面进行扩展：

1）系统存储容量可扩展

系统可在线将存储单元数据扩充至 256 台以上，最大容量达到 64PB。数据存储节点支持 SSD、SAS 及 SATA 等目前主流磁盘，单盘容量可达 4TB。

2）系统 IO 带宽可扩展

系统可以通过扩展数据节点，为大量的用户提供并发服务，通过数据节点的在线扩展可以有效的提高系统 IO 带宽。系统性能可随集群数据节点数量增加成线形增长，可以在线平滑、快速地扩展集群节点来提高系统的性能，同时也支持单个数据节点的硬件升级扩展（CPU，内存），不同层面来提升整个集群系统的整体处理能力。

每个数据节点配置一个 QDR IB 主机接口，单数据节点可提供 GB 以上吞吐带宽。随着系统中节点的增加，性能呈线性增长，可达数十 GB 吞吐带宽，满足成千上万客户端并发访

问需求。

4．可用性和高可靠性

集群系统所有节点均处于工作状态，任务被平均的分配到各个节点上；当一台节点发生故障，任务被自动平滑的分配到其他节点上。在无故障时可用性为100%；即使有一台节点发生故障，可用性仍为N-1/N。

系统从硬件和软件两个方面进行高可用设计，从而保证系统全年 7*24 小时无间断运行，关键数据不丢失。

1）硬件全冗余设计，整个存储系统中没有任何单点故障，任何部件都可以热更换，确保存储系统7*24连续运行。

2) 数据节点间采用机架感知的技术，同一副本不会存放于不同的机架上，当一个数据节点或机柜因故障不能正常工作时，不会丢失数据，从而保证整个文件系统运行的连续性，不影响主机业务运行。

3) 数据节点支持多副本存储数据，可以实现数据及时快速恢复，支持不同副本跨机架存储，有效避免某一机架故障或掉电引起数据丢失，提高数据可靠性。同时支持针对目录颗粒度的副本策略。

5．可维护性和易用性

系统的控制界面采用了友好的全中文 Web 显示方式；管理员不需经过长期专业培训，仅几个小时就可以熟练掌握系统的操作流程；提供对存储系统状态监控、事件管理、存储系统故障Email报警机制。

- 支持性能、容量负载均衡；
- 图形化性能及状态监控；
- 灵活的故障告警监控机制，提供邮件、日志等告警方式；
- 支持系统离线方式整体升级、在线滚动升级；

1.2 系统优势

◆ **先进的系统架构设计**

- 完全对等的横向可扩展集群架构；
- 支持异构形式的存储资源；
- 基于硬件性能的动态数据平衡；

◆ **高可靠，高可用性**

- 支持多副本存储数据，可以实现数据及时快速恢复；
- 支持不同副本跨机架存储，有效避免某一机架故障或掉电引起数据丢失，提高数据可

靠性；

- 支持针对目录颗粒度的副本策略；
- 系统全冗余，无单点故障；
- 控制器之间采用高可用配置。当元数据控制器出现故障时，可以自动切换到备用元数据控制器。任何一个控制器出现故障，不影响数据的正常访问；
- 监控集群通过特定协议进行相互之间的协调，保证整个监控集群就目前整个集群的状态达成一致，避免了由于集群中节点对于集群状态的认知不一致而导致的数据异常现象；

◆ **高性能、高扩展**

- 分布式架构确保标准存储节点和元数据节点无瓶颈，文件多节点并发访问；
- 元数据的缓存机制。引入租约机制，客户端具备了元数据缓存能力，避免了每次元数据请求都需要与元数据服务器进行通讯，较大幅度提高系统的性能；
- 支持基于容量的负载均衡，根据容量进行数据迁移，保持系统容量均衡；
- 支持在线同时增加单个或多个数据节点，系统容量将随之增加；
- 支持在线每次缩减单个数据节点，系统容量将随之减少；
- 支持在系统增加或删除节点时，容量自动均衡，实现系统弹性高效扩展，实现真正的在线扩容；
- 系统支持 64PB 以上的存储容量，整体支持十亿级的文件数量，单目录可以有效支持千万级的文件数量。

◆ **多协议支持**

- 支持 SMB、NFS、CIFS、FTP 多种协议；

◆ **便捷管理**

- 实时监控系统运行状态，
- 全方位的故障诊断机制，极大的降低管理难度；
- 采用副本机制保证数据冗余，降低使用阵列的成本与管理复杂度。

1.3 应用领域

AS13000 的高性能、高扩展、大容量数据存储是其显著特点。通过强大的数据处理能力，AS13000 在保证数据存储性能表现的基础上，利用自身创新型的体系架构，最大可扩展至 PB 级的数据存储容量，这对诸如航空航天、动漫渲染、地质勘探、高性能计算等行业应用客户，能够充分满足其数据剧增的存储容量需求，解决客户不断增长的存储支出的难题。

- 数据密集型领域：互联网、教育、数字图书馆、档案等
- 高性能计算领域：石油、勘探、地震、高能物理、空间信息处理等
- 数字媒体领域：视频点播、网络直播、数字电视、视频监控等

第二章　配置准备

2.1 节点规划

系统包含四种不同角色的节点，分别为监控节点（mon）、元数据节点（mds）、存储节点（osd）和接口节点。这四种不同角色的节点的规划如下。

1、 监控节点（mon）

机器类型：AS13000

主机命名规则：建议命名为 mon_0、mon_1、mon_2

角色：负责整个文件系统的监控，并对外提供挂载点。

2、 元数据节点（mds）

机器类型：AS13000

主机命名规则：建议命名为 mds_0、mds_1

角色：负责元数据操作。由于元数据存放于 osd 集群中，所以 mds 实际上是 osd 的客户端之一。

3、 存储节点（osd）

机器类型：AS13000

主机命名规则：建议命名为 osd_0、osd_1、osd_2、osd_3、...、osd_n（n 为 osd 的数量-1）。

角色：负责数据文件和元数据文件的存储，提供对象访问接口以及副本、故障恢复等功能。

在系统搭建过程中，一般 mon_0 使用一台物理机，mds_0 与 mon_1 共用一台物理机，mds_1 与 mon_2 共用一台物理机，一个 osd 对应物理机的一块硬盘，根据机器类型，决定一个物理机创建多少 osd。

4、 接口节点

机器类型：AS13000

主机命名规则：建议命名为 Inode0、Inode1、Inode2、...、Inoden（n 为 osd 的数量-1）。

角色：负责客户机与文件系统之间的数据交互和通信。

在系统搭建过程中，一般搭建两个或两个以上的接口节点，以满足接口节点的链路冗余，保证在接口节点单节点故障时，集群文件系统的可用性。

2.2 网络规划

系统网络架构包括管理网络、数据网络和监控网络。管理网络采用千兆以太网，用于

Web 系统管理；数据网络支持采用 IB 网络或是 10GE 网络，用于数据传输，提供数据服务；监控网络采用数据网络，用于监控各节点的硬件状态信息等。

1.管理网络

集群节点默认用户名为“root”，密码为“000000”。如果要修改此密码，建议将各个节点密码修改为统一密码。

管理网络采用千兆以太网，用于 Web 系统管理。

所有集群节点都需要配置管理网络，管理网络配置示例：

```
[root@mon_0 ~]# /icfs/sh/setup.sh
hostname:mon_0
eth0 ip(192.168.1.2):12.0.4.25
eth0 gateway(192.168.1.1):12.0.4.1
eth0 netmask(255.255.255.0):255.255.255.0
```

注意

管理网络默认采用设备 eth0。

2.数据网络

数据网络支持采用 IB 网络或是 10GE 网络，用于数据传输，提供数据服务，

所有节点都需要配置数据网络，以配置 IB 为数据网络示例：

```
[root@mon_0 ~]# /icfs/sh/setup.sh
hostname:mon_0
eth0 ip(192.168.1.2):12.0.4.25
eth0 gateway(192.168.1.1):12.0.4.1
eth0 netmask(255.255.255.0):255.255.255.0
ib0 ip(192.168.1.2):192.168.4.25
ib0 netmask(255.255.255.0):255.255.0.0
```

第三章 系统配置

本部分共包含 7 个步骤，详细的介绍的配置和使用系统的所有步骤，通过认真阅读本章节，管理员将能够熟练配置和使用文件系统。

3.1 步骤 1——配置文件管理

本部分介绍文件系统的主配置文件的配置方法。

关键术语

文件系统的主配置文件是存放当前集群中所有节点的配置信息。

集群系统的主配置文件为/etc/icfs/icfs.conf，里面配置了集群运行的一些重要参数，以及各个模块的工作方式、路径、角色分配等。icfs.conf 中主要由【global】、【mon】、【mds】和【osd】这四个模块组成。

【global】模块中存放的全局参数，该设置对文件系统中的所有节点有效。

【mon】模块中存放对所有 mon 节点有效的一些系统参数以及 mon 节点信息。

【mds】模块中存放了对所有 mds 节点有效的一些系统参数以及 mds 节点配置信息。

【osd】模块中存放了对所有 osd 节点有效的一些系统参数以及 osd 节点配置信息。

配置文件配置

1. 全局参数设置

全局参数存放于主配置文件【global】模块中，该模块主要设置的参数为文件系统的认证方式、能够打开的最大文件数、以及 log 的存放目录和文件名、进程 pid 信息存放目录和文件名等。如下所示

```
auth supported = none            //是否开启认证，若为 none，则不开启
max open files = 131072          //允许打开的最大文件数
log file = /var/log/icfs/$name.log     //日志信息存放地址
 pid file = /var/run/icfs/$name.pid        //进程信息存放地址
```

2. mon 节点设置

【mon】模块主要是针对所有 mon 节点进行设置，并指定每个 mon 的地址和运行方式，如下例所示

```
[mon]                                    //mon 总体设置
```

```
        mon_debug_dump_transactions = false      //关闭日志转储的调试功能
        mon data = /data/$name                   // mon 数据存放目录
      [mon.0]                                    //mon.0 的设置
        host = mon_0                             //设置主机名
        mon addr = 192.168.6.4:6789              //指定主机 IP 地址和端口号
      [mon.1]
        host = mds_0
        mon addr = 192.168.6.3:6789
```

3. mds 节点设置

【mds】模块主要是针对所有 mds 节点进行设置，并指定单个 mds 的地址，默认情况向不对 mds 做任何设置。示例如下

```
  [mds]          //mds 参数设置
  [mds.0]                //mds.0 参数设置，默认为主 mds
    host = mds_0        //指定主机名为 mds_0 的主机作为 mds.0
  [mds.1]                //mds.1 参数设置，默认为备用 mds
      host = mds_1        //指定主机名 mds_1 的主机作为 mds.1
```

4. osd 节点设置

【osd】模块主要针对所有 osd 节点进行参数设置，并指定每个 osd 所对应的主机地址和盘符。

```
  [osd]         //osd 全局参数设置
    osd mkfs type = xfs                //指定本地文件系统为 xfs，若不指定，则为 btrfs
    osd data = /data/$name             //指定 osd 的数据盘挂载地址
    osd journal = /data/osd.$id/journal$id  //指定 osd 日志写入的存放地址
    osd journal size = 50000                //指定 osd 日志写入的大小，单位为 MB
  [osd.0]                          osd.0 参数设置
```

```
host = osd_0              //指定主机名为 osd_0 的主机作为 osd.0
devs = /dev/sda           //指定存储设备的盘符为/dev/sda
```

3.2 步骤 2——时间管理

本章介绍关于集群系统时间的管理设置。

关键术语

Ntpd 是 linux 系统中常用的一个时间服务，通过配置 ntpd 服务可以方便的构造本地时间服务器，并对整个集群进行时间同步。

时间同步是指将整个集群中的所有节点进行时钟同步，所有节点的时间保持一致（误差不能大于 0.05sec）

时间同步配置

1. 设置系统时间

首先在主 mon（mon.0）上设置当前系统时间，设置当前系统时间有多种方法，下面我们选择利用 date 命令来设置当前系统时间

```
[root@mon_1 sh]# date -s "2013-7-23 10:50:20"
Tue Jul 23 10:50:20 CST 2013
[root@mon_1 sh]# hwclock –w
```

第二步调用 hwclock 命令是为了将当前系统时间写入到主板的 rtc 芯片

2. 配置时间服务器

为了保证集群的时间一致，采用 ntpd 服务，首先将主 mon 的节点时间调整到正确的时间，然后修改 ntpd 服务的配置文件/etc/ntp.conf,将下面两行前面的注释去掉，使其可以使用本地时间作为时间服务器，同时需要注释掉以 server 开头的网络时间服务器地址。 这是因为大部分设备在运行过程中无法访问官方的时间服务器。

```
#server 0.centos.pool.ntp.org
#server 1.centos.pool.ntp.org
#server 2.centos.pool.ntp.org
server  127.127.1.0     # local clock
```

```
fudge   127.127.1.0 stratum 10
```

启动 ntpd 服务

```
[root@mon_1 sh]# service ntpd start
```

ntpd 的配置文件 ntp.conf 实例如下

```
driftfile /var/lib/ntp/drift
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
server 127.127.1.0   # local clock
fudge  127.127.1.0 stratum 10
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

3. 配置时间同步客户端

设置完时间同步服务器以后，在集群的其余所有节点上，同时也需要启动 ntpd 服务作为时间同步客户端，并定时与时间同步服务器端进行时间同步。

时间同步客户端的 ntp.conf 中需要将 server 目标指向时间同步服务器的 IP 地址，因此只需修改 server 即可

```
server server_ip
```

启动 ntpd 服务

```
[root@mon_1 sh]# service ntpd start
```

Ntpd 服务的 ntp.conf 配置实例如下：

```
driftfile /var/lib/ntp/drift
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
```

```
server 12.0.4.20                          //指向时间同步服务器的 IP 地址
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

3.3 步骤 3——文件系统

关键术语

监控节点–负责监控和保存整个集群的状态，记录系统中产生的日志，向外提供客户端的挂载。

元数据节点–负责元数据操作。由于元数据存放于 osd 集群中，所以 mds 实际上是 osd 的客户端之一。

存储节点–负责数据文件和元数据文件的存储，提供对象访问接口以及副本、故障恢复等功能。

文件系统操作

集群系统操作包含对集群的操作和接口节点的配置。其中，集群的操作包含集群文件系统初始化、启动、关闭、状态检查等操作；接口节点操作包含接口节点多协议的配置、对应的客户端相关配置以及 NFS 的配置。下面将详细讲述对应的操作命令。

1. 系统初始化

集群文件系统初始化操作用于创建一个空的集群文件系统。

```
[root@mon_0 ~]# mkicfsfs -a -c /etc/icfs/icfs.conf –mkfs
```

在该命令中，-a 表示对集群中所有节点进行操作，-c 表示要使用的集群配置文件，--mkfs 表示 OSD 存储使用的是 xfs 文件系统。

注意

该命令将创建一个空的集群文件系统，若已存在文件系统，该命令将清空原有文件系统的所有数据。

2. 系统启动

集群文件系统启动操作用于启动整个集群文件系统。

```
[root@mon_0 ~]# service icfs –a -c /etc/icfs/icfs.conf start
```

在该命令中，-a 表示集群中所有节点都执行启动操作，-c 表示启动集群需要加载的配置文件。

3. 系统关闭

集群文件系统关闭操作用于关闭整个集群文件系统。

```
[root@mon_0 ~]# service icfs -a stop
```

注意

该命令将关闭整个集群，存储服务将不可用。

4. 系统重启

集群文件系统重启操作用于对整个集群文件系统进行重启。

```
[root@mon_0 ~]# service icfs -a restart
```

注意

该命令将重新启动整个集群，存储服务将暂时不可用。

5. 状态检查

集群文件系统状态检查操作用于查看整个集群系统的运行状态。

```
[root@mon_0 ~]# icfs –s
```

该命令会打印集群系统的状态，包含系统的健康状况以及系统的核心数据状态。

6. 故障修复

当集群发生故障时，可以参考如下步骤进行修复：

1、 查看整个集群系统的运行状态。

```
[root@mon_0 ~]# icfs –s
```

该命令打印当前系统的状态如下：

```
[root@osd_0 ~]# icfs health
HEALTH_WARN 384 pgs degraded; 384 pgs stuck unclean; recovery 21/42 degraded (50.000%)
[root@osd_0 ~]# icfs status
   health HEALTH_WARN 384 pgs degraded; 384 pgs stuck unclean; recovery 21/42 degraded (50.000%)
   monmap e1: 1 mons at {0=127.0.0.1:6789/0}, election epoch 2, quorum 0 0
   osdmap e3: 1 osds: 1 up, 1 in
    pgmap v6: 384 pgs: 384 active+degraded; 9518 bytes data, 1002 MB used, 3895 MB / 5472 MB avail; 0B/s rd, 212B/
s wr, 0op/s; 21/42 degraded (50.000%)
   mdsmap e4: 1/1/1 up {0=0=up:active}
```

2、 在大部分情况下，可以使用下面命令进行修复,例如要修复 mon.2 节点，可执行以下命令：

```
[root@mon_0 ~]# service icfs -a restart mon.2
```

3.4 步骤 4——系统监控

关键术语

集群监控 文件系统 监控项 系统负载

查看系统监控信息

用户可以通过 Web 浏览器访问浪潮分布式存储系统，建议使用 google +1024*768 的方式式浏览。在管理员控制台的浏览器地址栏中输入：http://server ip 即可显示方式观看系统监控主界面。

1. 系统整体状态信息

监控主界面显示集群系统整体的监控状态信息，包括 cpu 、mem、硬盘利用率， I/O 负载、网络流量情况等信息，如图 3-4-1 所示。

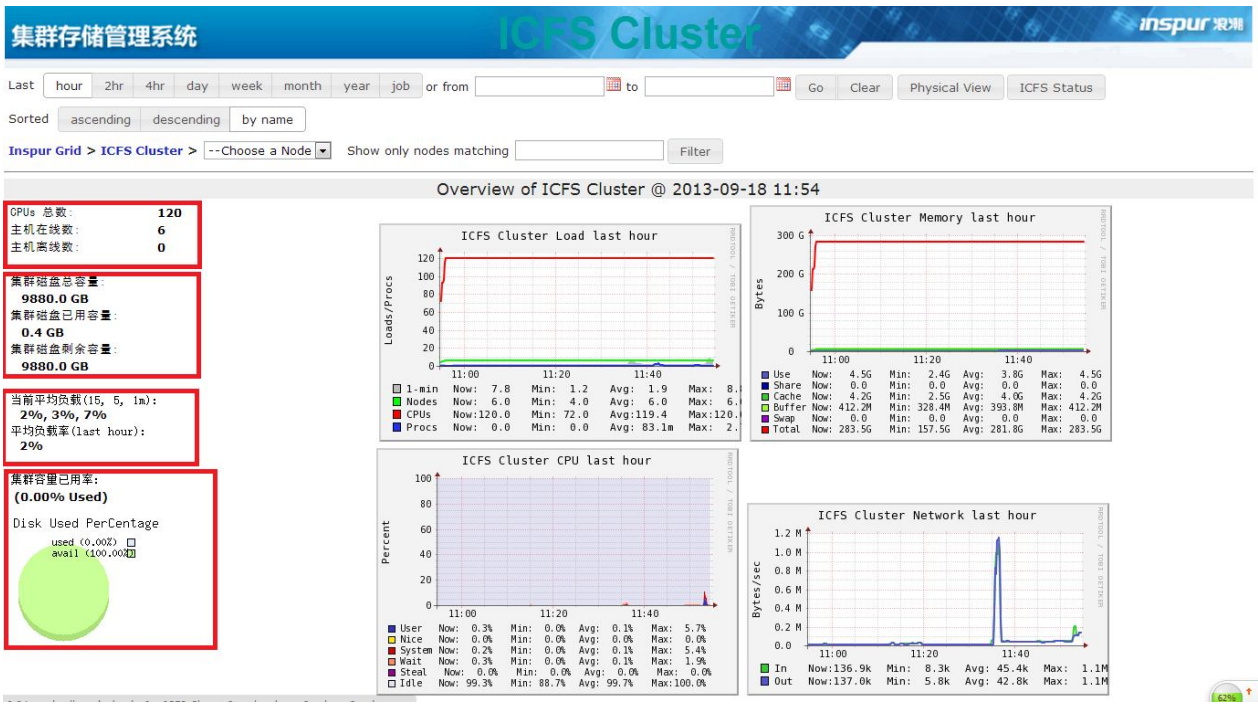


图 3-4-1 主界面监控部分

在主界面的开始部分，我们可以看到若干个时间按钮，“hour”、“2hr”、“4hr”、“day”等，点击这些按钮，监控主界面上会显示最近每小时、每两小时、每四小时、每天、每周、每年的详细监控信息。同时还可以自己选择某个时间段的历史监控信息（点击”from”和”to”之间的按钮，选择相应时间段，将会有该时间段的历史监控信息，监控系统默认是显示每小时的历史监控信息，同时每 60 秒刷新一次最新的监控数据）。注意：在主界面时间列的选项中，选择“job”按钮时，需要选择相应的时间段才会有监控图形出现，同时返回主界面时，应先点击“clear”按钮，取消 job，才能进入默认主界面。

主界面左侧统计了集群的节点上下线数量（从图 3-4-1 可以看到目前该集群有六个节点，六个节点都处于在线状态）、集群的磁盘使用情况（包括集群的磁盘总容量、已使用容量、剩余可使用容量以及磁盘已用容量百分比等，并用直观的饼状图展示）以及系统的平均负载情况（每分钟、每五分钟及每十五分钟平均负载情况）。

主界面尾部还有相关的集群节点列表，并且集群节点列表中，每个节点的颜色显示状态标志着节点当前的负载情况，如图 3-4-2 所示，当前状态显示集群负载的状态正常。当某个节点掉电时，监控界面上会有信息，如图 3-4-3 所示。点击某个节点网格，可以进入到该节点的详细监控界面上。对于节点监控页面，我们将在第二部分介绍。

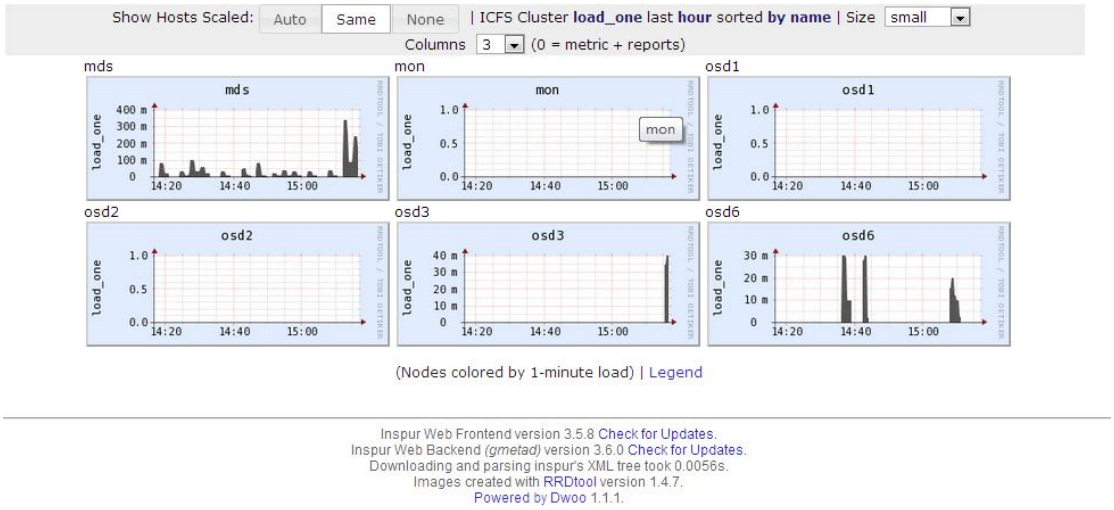


图 3-4-2 主界面节点列表部分

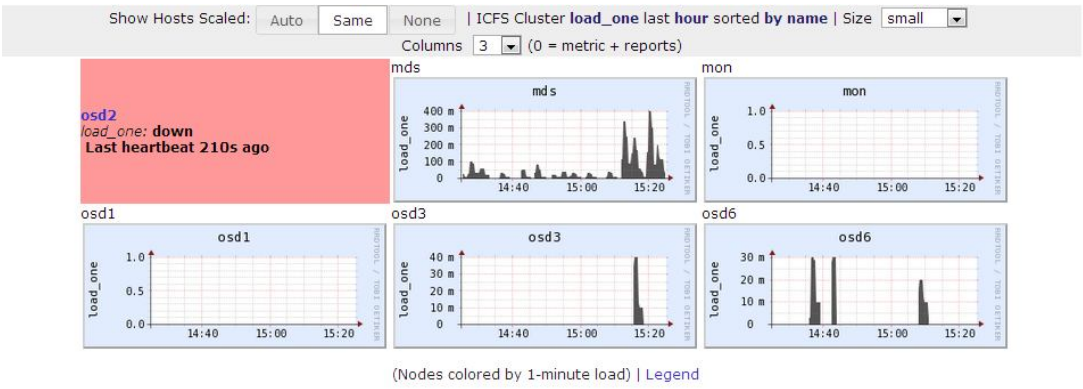


图 3-4-3 某个节点掉电状态图

在主界面点击“Physical View”将会显示集群的所有监控节点的部分物理信息、当前集群的磁盘总容量以及当前集群中磁盘使用率最高的节点，如图 3-4-4 所示，红色圆圈部分显示了当前分布式集群中的磁盘使用率最高的节点信息。

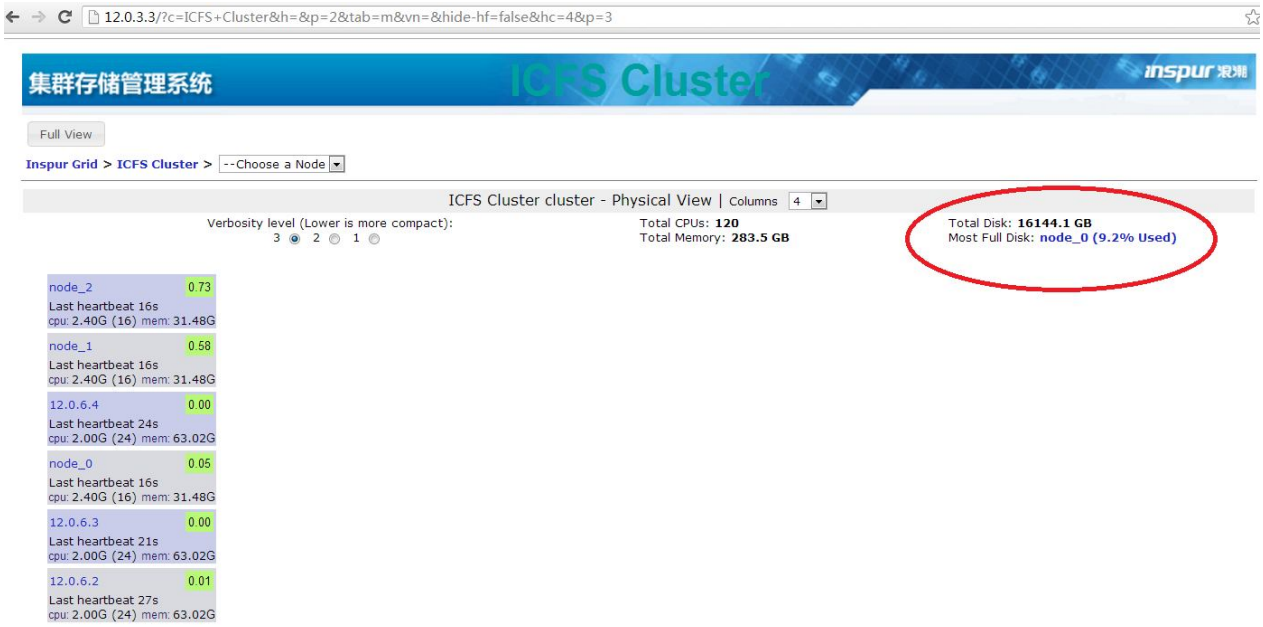


图 3-4-4 集群节点物理信息页面

在主界面点击“ICFS Status”按钮，将会显示当前分布式文件系统的各种状态信息，如图 3-4-5 所示。显示了集群的健康状态信息(HEALTH_OK、HEALTH_WARNING、HEALTH_ERROR)、mon、mds、osd 以及 PG 相关信息，通过该页面可以了解到当前集群文件系统的关键参数信息。

图 3-4-5 分布式存储文件系统状态信息

在集群监控主界面上，任意点击监控网格图，将会进入该监控项的大图监控信息，

包括该监控项在 1 小时、2 小时、3 小时、4 小时、一天、一周、一月、一年的监控信息，如图 3-4-6 所示为点击“load”监控项之后的页面显示。

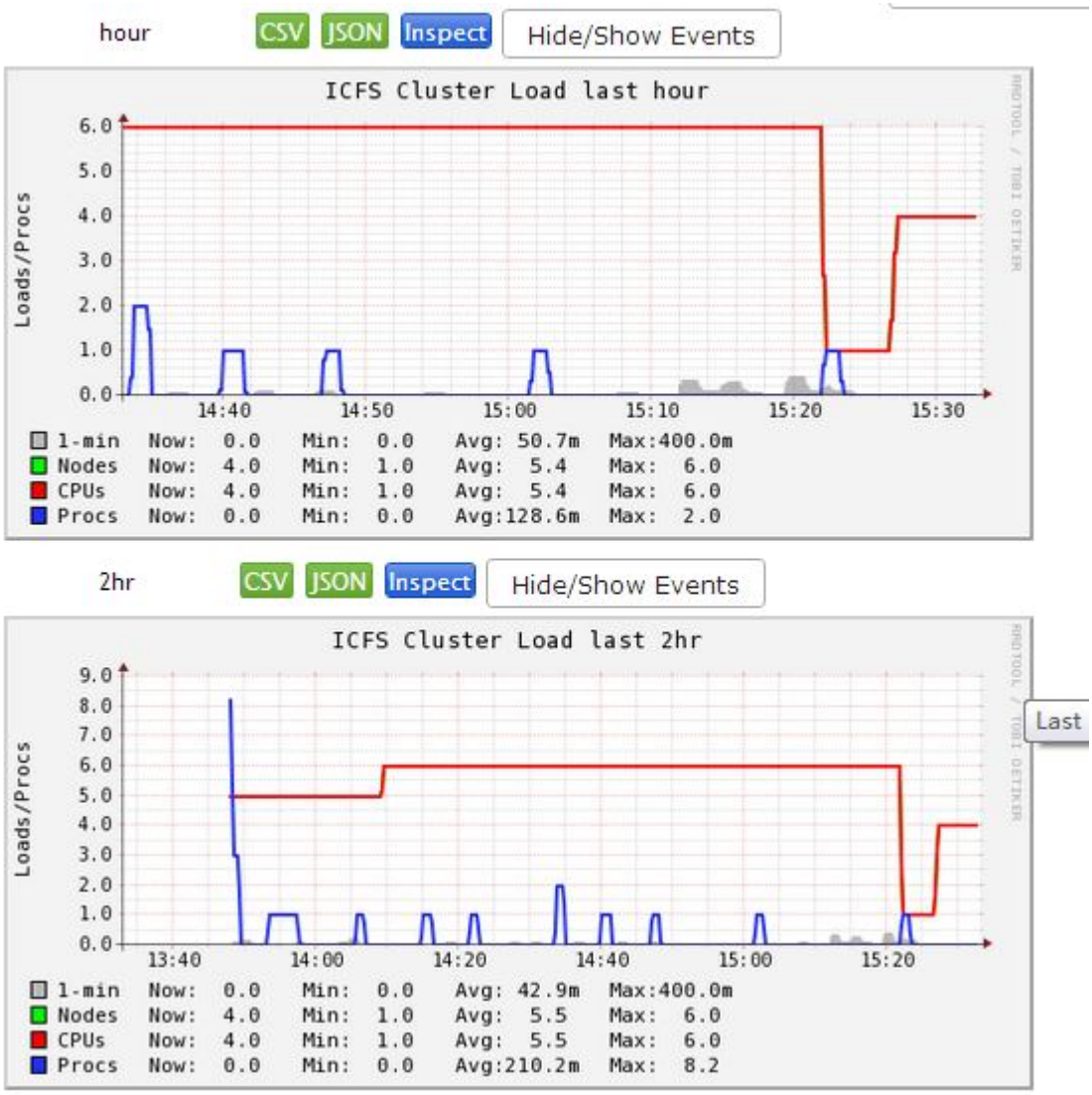


图 3-4-6 集群某监控项大图

2. 系统节点状态信息

在节点监控页面，显示单个节点的监控信息，包括 cpu 、mem、硬盘利用率， I/O 负载、网络流量情况等信息，如图 3-4-7 所示。

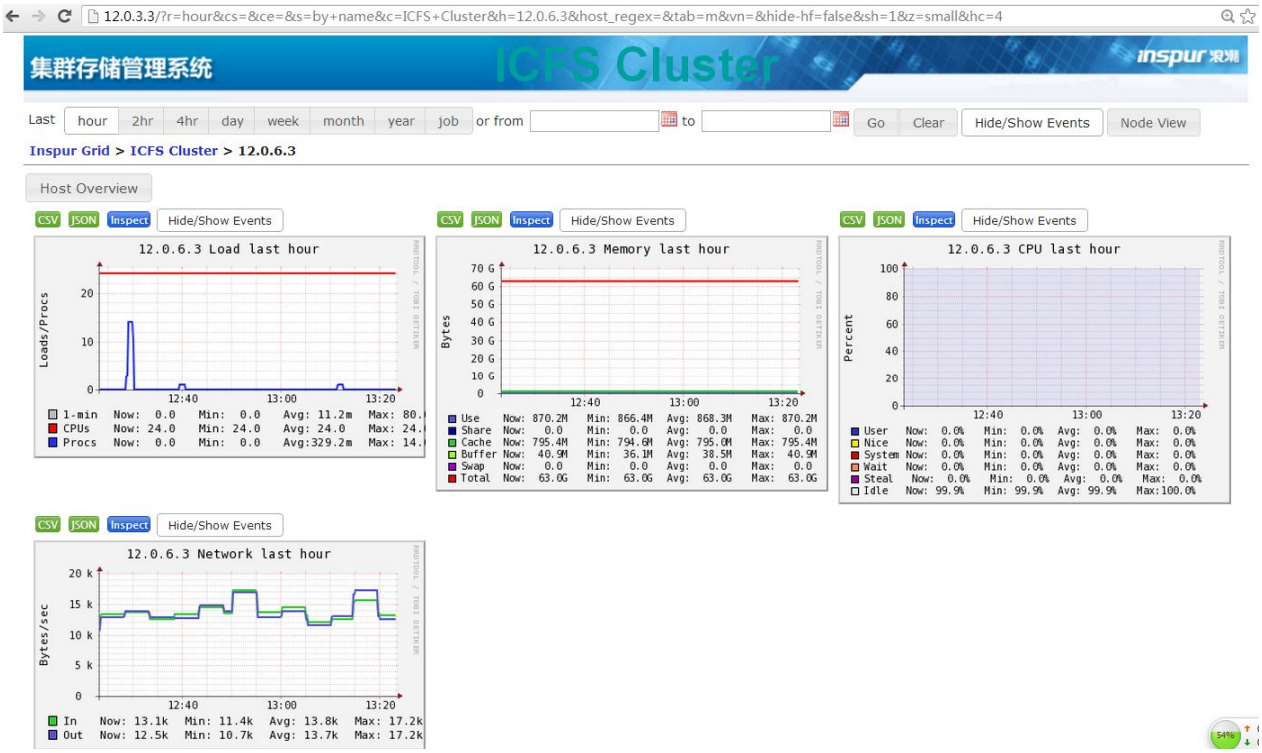


图 3-4-7 单个节点界面监控信息

点击图 5-7 节点界面的时间按钮，同样会显示该节点在 1 小时、2 小时、3 小时、4 小时、一天、一周、一月、一年的监控信息。

在节点界面监控页面的中部可以看到具体的某个监控项的详细信息，包括 cpu、disk、load、memory、network 及 process 等监控汇总信息。图 3-4-8 是 cpu metric 展开后的结果，从图中可以看出 cpu 监控项主要包括 6 个字监控项，分别是”cpu nice”, “cpu wio”, ”cpu user”, ”cpu idle”, ”cpu system”, ”cpu aidle”等。表格 5-1 详细阐述了各个监控项的作用。

监控项	说明	监控值
Load_one	One minute load average 每分钟的系统平均负载	load_one=0.0
Load_five	Five minute load average 每 5 分钟的系统平均负载	load_five=0.0
Load_fifteen	Fifteen minute load average 每 15 分钟的系统平均负载	load_fifteen=0.0
mem_total	Total amount of memory displayed in KBs	mem_total=2075288.0

	物理内存总量（KBs 显示）	
mem_cached	Amount of cached memory 缓存内存大小	mem_cached=470732.0
mem_free	Amount of available memory 空闲内存大小	mem_free=1128860.0
mem_buffers	Amount of buffered memory 内核缓存的内存总量	mem_buffers=353264.0
swap_total	Total amount of swap space displayed in KBs 交换分区总量（KBs 显示）	swap_total=8289532.0
swap_free	Amount of available swap memory 空闲交换分区大小	swap_free=8289532.0
mem_shared	Amount of shared memory 共享内存大小	mem_shared=0.0
proc_run	Total number of running processes 运行的进程总数	proc_run=0.0
proc_total	Total number of processes 进程总数	proc_total=65.0
cpu_idle	Percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request 空闲 CPU 百分比	cpu_idle=99.6
cpu_aidle	Percent of time since boot idle CPU 启动的空闲 CPU 百分比	cpu_aidle=99.8
cpu_user	Percentage of CPU utilization that	cpu_user=0.2

	occurred while executing at the user level 用户空间占用 CPU 百分比	
cpu_nice	Percentage of CPU utilization that occurred while executing at the user level with nice priority 用户进程空间内改变过优先级的进程占用 CPU 百分比	cpu_nice=0.0
cpu_system	Percentage of CPU utilization that occurred while executing at the system level 内核空间占用 CPU 百分比	cpu_system=0.1
cpu_num	Total number of CPUs CPU 线程总数	cpu_num=2.0
cpu_speed	CPU Speed in terms of MHz CPU 速度（MHz）	cpu_speed=2993.0
cpu_wio	Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request Cpu 空闲时的最大 I/O 请求	cpu_wio=0.2
part_max_used	Maximum percent used for all partitions	part_max_used=21.2
disk_total	Total available disk space 磁盘总大小	disk_total=133.991
disk_free	Total free disk space 剩余磁盘空间	disk_free=124.707
boottime	The last time that the system was started	boottime=1285905983.0

	系统启动时间	
pkts_in	Packets in per second 每秒进来的包	pkts_in=10.5
pkts_out	Packets out per second 每秒出去的包	pkts_out=2.85
bytes_in	Number of bytes in per second 每秒进来字节数	bytes_in=769.35
bytes_out	Number of bytes out per second 每秒出去字节数	bytes_out=3674.02
machine_type	System architecture 系统版本（X86 或 64）	machine_type=x86
os_name	系统名字	os_name=Linux
os_release	系统版本	os_release=2.6.18-164.el5PAE
gexec	gexec available	gexec=OFF
disk_free_rootfs	Disk space available on	WARN: ‘ disk_free_rootfs

表格 3-4-1 监控项解析

图 3-4-8 单个节点的 metric 信息

图 3-4-9 所示是节点监控页面目前的汇总监控项列表，每个监控项又有几个子监控项。

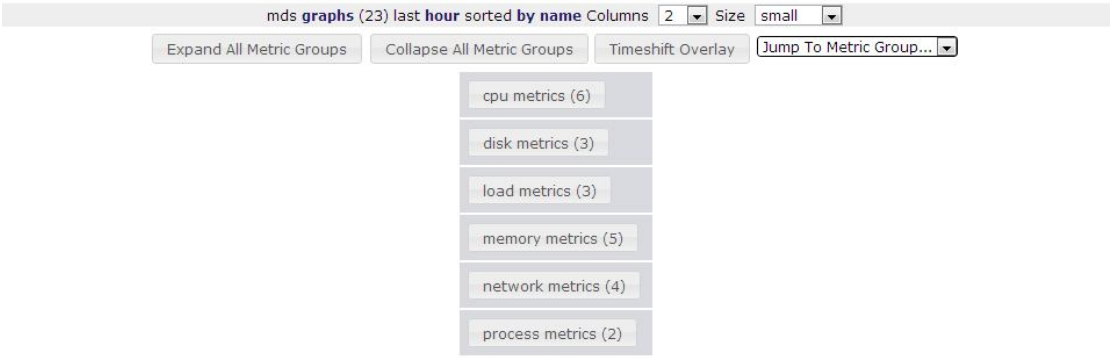


图 3-4-9 metric 列表

图 3-4-10 为在节点监控界面点击“Node View”按钮后进入的节点系统监控信息页面。

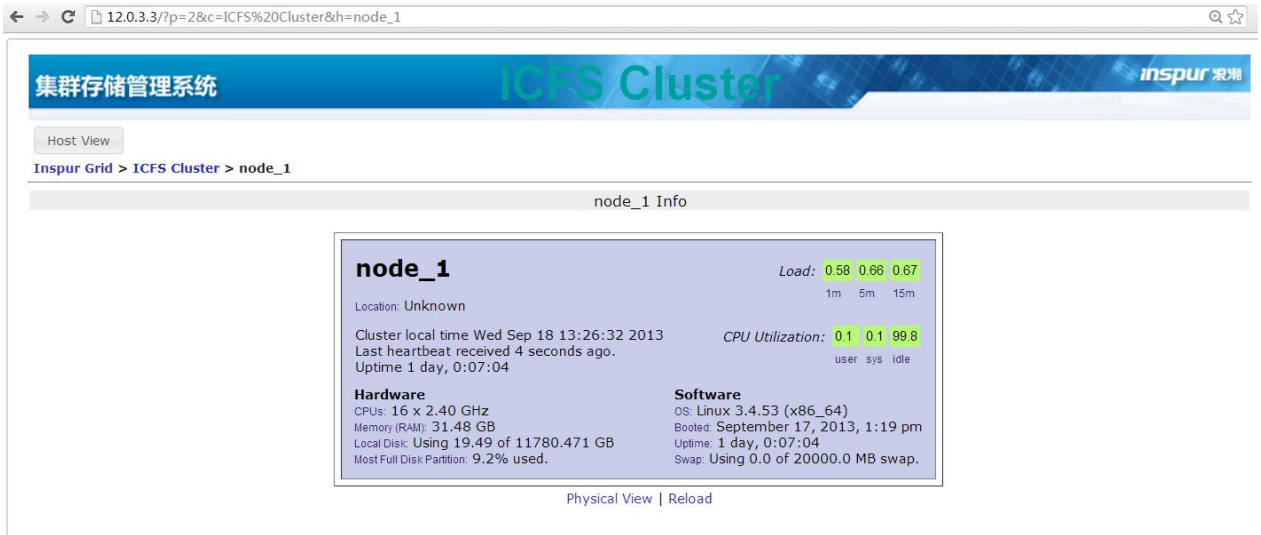


图 3-4-10 单个节点的系统环境页面

图 3-4-11 显示的为集群节点在不同状态下的网格颜色（不同的状态颜色表明该节点的负载状态）。当节点状态颜色处在 Red 时，表明负载利用率已经超过 100%，需要及时调整数据存储方案。

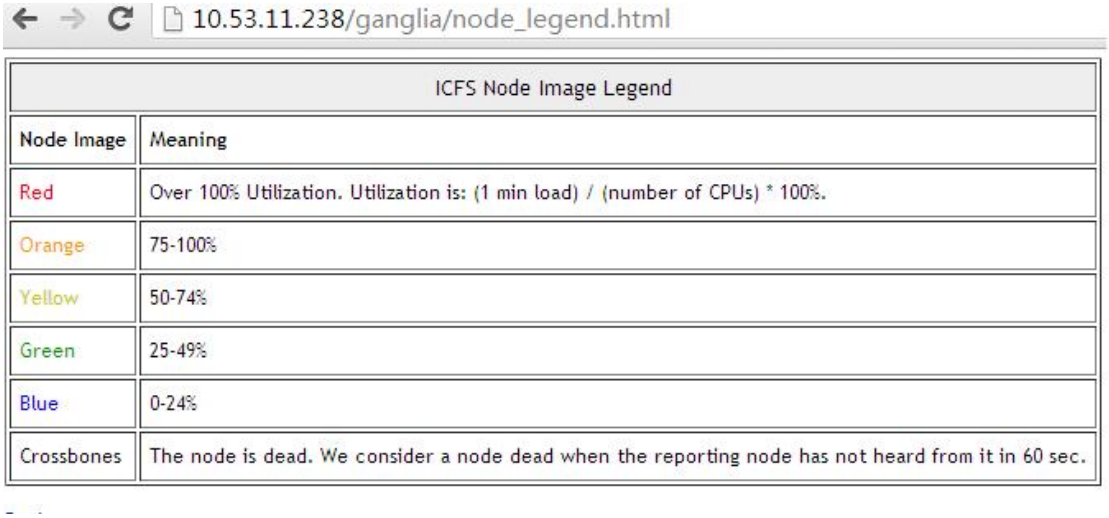


图 3-4-11 集群节点负载在不同状态的颜色

图 3-4-12 显示集群中某个节点的系统版本相关信息，点击“Host Overview”按钮，即可显示，在点击该按钮将隐藏该信息。

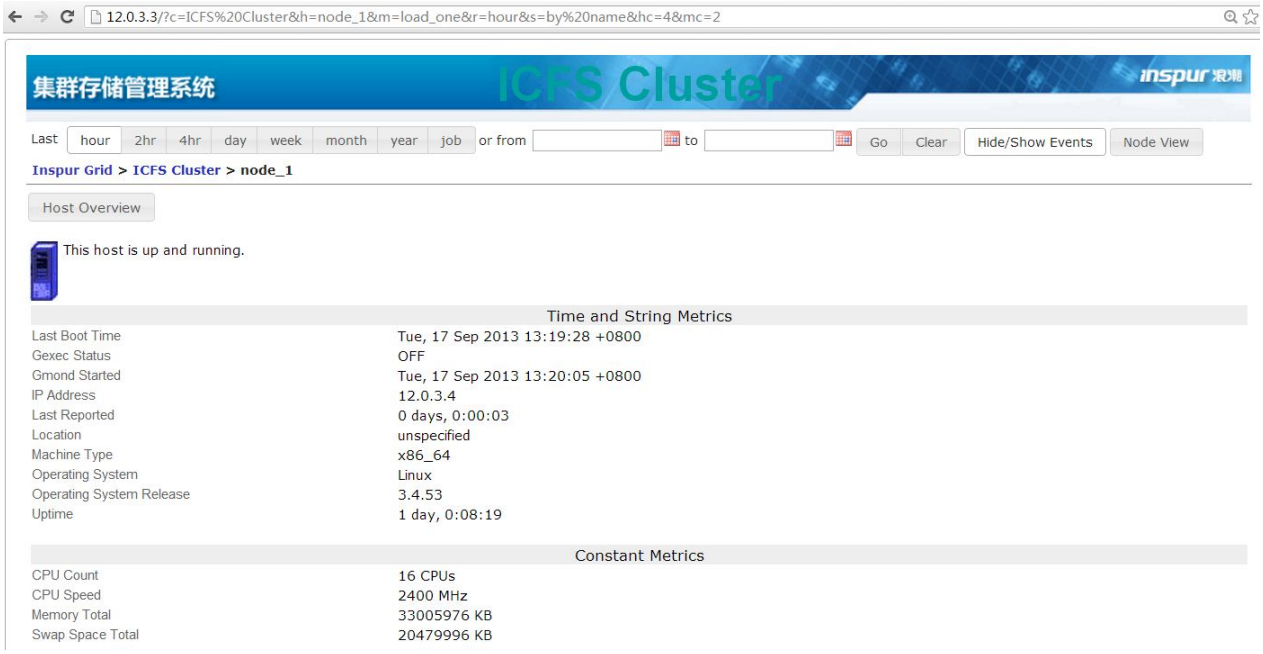


图 3-4-12 节点系统版本信息

3.5 步骤 5——日志管理

关键术语

日志是指集群文件系统中在 mon 节点生成的文件系统运行状态日志，可以从该日志查

看当前系统运行状态，系统出现的异常，以及各个节点的状态信息。

日志管理操作

1. 日志存放位置

集群日志存放在 mon.0（主 mon）节点的/var/log/icfs/目录下，集群整体目录存放在文件 icfs.log 中，其余的各个节点的日志信息都存放于以其角色命名的文件中，后缀名统一为.log。例如：mon.0.log。

集群日志每天将会被打包，并存放在当前目录下。如下所示

```
[root@osd_0 icfs]# ll
total 3960
-rw-r--r-- 1 root root       0 Aug 12 03:28 client.admin.log
-rw-r--r-- 1 root root      20 Aug  7 03:31 client.admin.log-20130807.gz
-rw-r--r-- 1 root root     237 Aug  8 03:15 client.admin.log-20130808.gz
-rw-r--r-- 1 root root    1240 Aug  9 03:34 client.admin.log-20130809
-rw-r--r-- 1 root root     166 Aug 10 03:19 client.admin.log-20130810.gz
-rw-r--r-- 1 root root      20 Aug 11 03:31 client.admin.log-20130811.gz
-rw-r--r-- 1 root root      20 Aug 12 03:28 client.admin.log-20130812.gz
-rw------- 1 root root   34701 Aug 12 15:01 icfs.log
-rw------- 1 root root 1269830 Aug  9 03:34 icfs.log-20130809.gz
-rw------- 1 root root  113807 Aug 10 03:19 icfs.log-20130810.gz
-rw------- 1 root root   95068 Aug 11 03:31 icfs.log-20130811.gz
-rw------- 1 root root   96312 Aug 12 03:28 icfs.log-20130812.gz
-rw-r--r-- 1 root root       0 Aug 12 03:28 mds.0.log
-rw-r--r-- 1 root root      20 Aug  7 03:31 mds.0.log-20130807.gz
-rw-r--r-- 1 root root      20 Aug  8 03:15 mds.0.log-20130808.gz
-rw-r--r-- 1 root root       0 Aug  9 03:34 mds.0.log-20130809
-rw-r--r-- 1 root root      20 Aug 10 03:19 mds.0.log-20130810.gz
-rw-r--r-- 1 root root      20 Aug 11 03:31 mds.0.log-20130811.gz
-rw-r--r-- 1 root root      20 Aug 12 03:28 mds.0.log-20130812.gz
-rw-r--r-- 1 root root       0 Aug 12 03:28 mds.1.log
-rw-r--r-- 1 root root      20 Aug  9 03:34 mds.1.log-20130809.gz
-rw-r--r-- 1 root root      20 Aug 10 03:19 mds.1.log-20130810.gz
-rw-r--r-- 1 root root      20 Aug 11 03:31 mds.1.log-20130811.gz
-rw-r--r-- 1 root root      20 Aug 12 03:28 mds.1.log-20130812.gz
-rw-r--r-- 1 root root  131538 Aug 12 15:05 mon.0.log
-rw-r--r-- 1 root root      20 Aug  7 03:31 mon.0.log-20130807.gz
-rw-r--r-- 1 root root   15963 Aug  8 03:15 mon.0.log-20130808.gz
-rw-r--r-- 1 root root  317585 Aug  9 03:34 mon.0.log-20130809.gz
-rw-r--r-- 1 root root   69608 Aug 10 03:19 mon.0.log-20130810.gz
-rw-r--r-- 1 root root   37036 Aug 11 03:31 mon.0.log-20130811.gz
-rw-r--r-- 1 root root   37503 Aug 12 03:28 mon.0.log-20130812.gz
-rw-r--r-- 1 root root       0 Aug 12 03:28 mon.1.log
-rw-r--r-- 1 root root     157 Aug  9 03:34 mon.1.log-20130809.gz
-rw-r--r-- 1 root root      20 Aug 10 03:19 mon.1.log-20130810.gz
-rw-r--r-- 1 root root      20 Aug 11 03:31 mon.1.log-20130811.gz
-rw-r--r-- 1 root root      20 Aug 12 03:28 mon.1.log-20130812.gz
-rw-r--r-- 1 root root       0 Aug 12 03:28 mon.2.log
-rw-r--r-- 1 root root     158 Aug  9 03:34 mon.2.log-20130809.gz
-rw-r--r-- 1 root root      20 Aug 10 03:19 mon.2.log-20130810.gz
```

```
-rw-r--r-- 1 root root        20 Aug 11 03:31 mon.2.log-20130811.gz
-rw-r--r-- 1 root root        20 Aug 12 03:28 mon.2.log-20130812.gz
-rw-r--r-- 1 root root     65254 Aug 12 15:06 osd.0.log
-rw-r--r-- 1 root root        20 Aug  7 03:31 osd.0.log-20130807.gz
-rw-r--r-- 1 root root      7456 Aug  8 03:15 osd.0.log-20130808.gz
-rw-r--r-- 1 root root    246283 Aug  9 03:34 osd.0.log-20130809.gz
-rw-r--r-- 1 root root     13005 Aug 10 03:19 osd.0.log-20130810.gz
```

2. 日志级别设置

在系统运行过程中，当系统某一部分出现问题时，我们一般会打开日志记录，查看系统的运行信息。记录详细日志需要消耗系统的资源，所以在集群的某一部分出现问题时，只需对该部分开启记录详细日志功能。例如，集群中的 mds 运行正常，但 osd 运行不正常，只需要打开 osd 模块的详细日志记录功能。

在查看单节点日志的过程中，有时我们只需要看简单的日志记录即可，不需要看详细的日志记录，就可以定位到问题，然后我们可以设定该节点日志级别来进行查看。命令格式如下：

```
icfs {daemon-type} tell {daemon id or *} injectargs '--{name} {value} [--{name} {value}]'
```

例如，当对 osd_0 调整日志级别为 5 时，其中，日志级别分为 1-20,20 为最详细级别。命令如下

```
[root@mon_0 icfs]# icfs tell osd.0 injectargs '--debug-osd 5'
```

3. 系统日志查看

查看系统日志或节点日志可以用 tail 命令和 more 命令查看，如下所示

```
[root@osd_0 icfs]# more mon.0.log
2013-08-12  03:28:33.995040  7fbe0c97f700  0  mon.0@0(leader).data_health(40)
update_stats avail 88% total 100791728 used 68
24936 avail 88846792
2013-08-12  03:29:33.995222  7fbe0c97f700  0  mon.0@0(leader).data_health(40)
update_stats avail 88% total 100791728 used 68
24800 avail 88846928
2013-08-12  03:30:33.995398  7fbe0c97f700  0  mon.0@0(leader).data_health(40)
update_stats avail 88% total 100791728 used 68
24812 avail 88846916
2013-08-12  03:31:33.995606  7fbe0c97f700  0  mon.0@0(leader).data_health(40)
update_stats avail 88% total 100791728 used 68
24812 avail 88846916
```

3.6 步骤 6——客户端挂载

本章节介绍客户端挂载的相关概念和相关操作。

关键术语

运行状态–文件系统状态检查健康状况正常。

挂载状态–文件系统在指定客户端是否挂载。

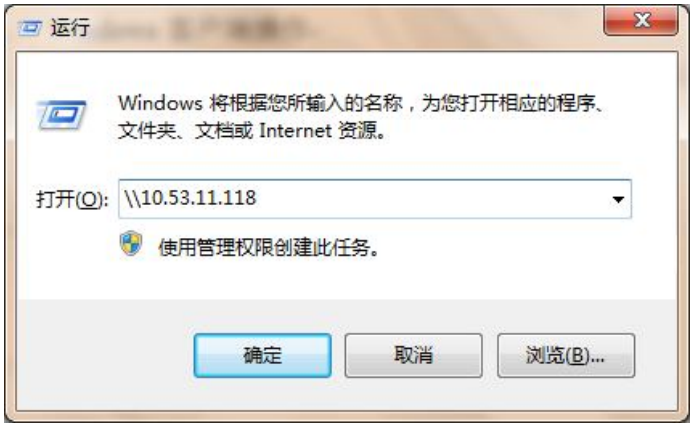
挂载目录–文件系统在客户端的挂载目录，如/mnt/data，在此目录下挂载文件系统后，就可以通过此目录查看文件系统的内容。

客户端挂载操作

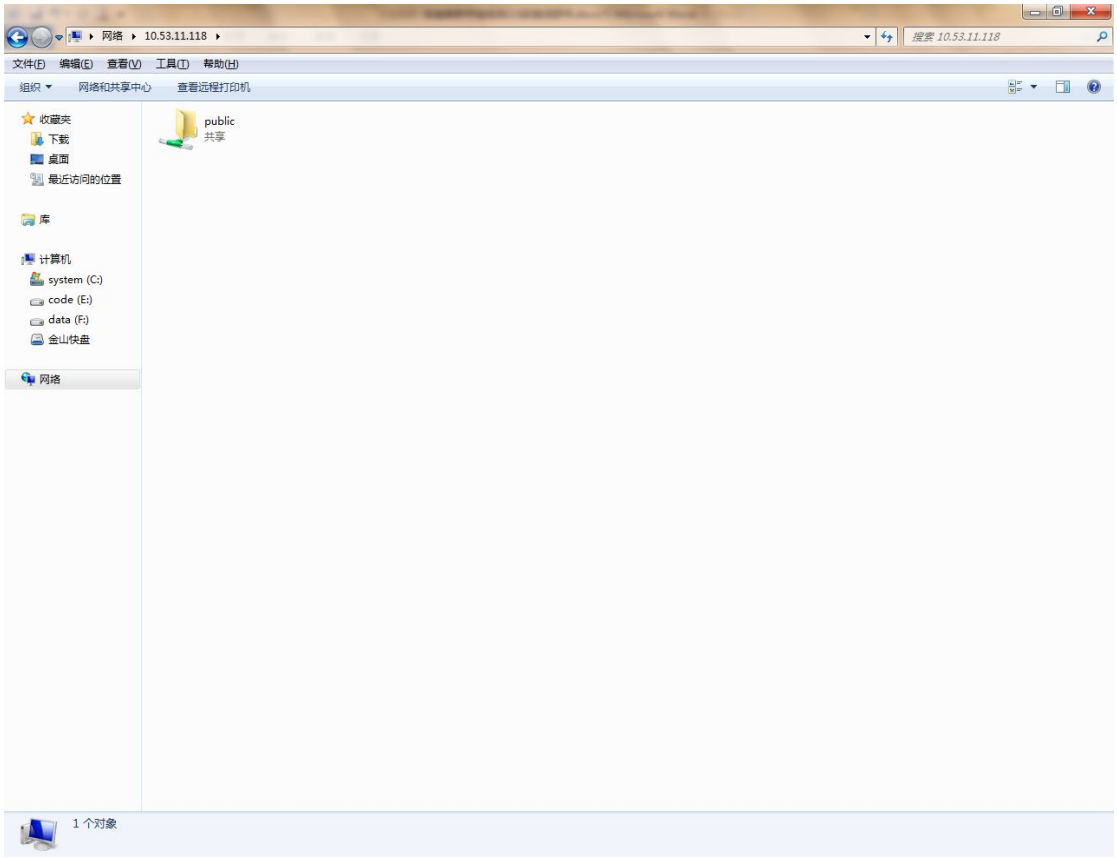
- Windows 客户端挂载

多协议接口节点集群配置完成之后，本节介绍 Windows 客户端上挂载操作。

在接口节点集群的配置中，设置对外开放的 ip 地址为 10.53.11.118 和 10.53.11.119，客户端可以使用任选一个 ip 进行挂载。在 Windows 的运行对话框中输入\\10.53.11.118。



目录界面如下所示：



- Linux 客户端挂载

NFS 服务器端配置完成，可以在 Linux 客户端进行挂载。

```
[root@client1 ~]# mount -t nfs 10.53.11.118:/mnt/icfs /mnt/nfs
```

使用 df 命令查看挂载情况。

```
[root@client1 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdn1             193G   31G  152G  17% /
tmpfs                  16G     0   16G   0% /dev/shm
10.53.11.118:/         33T  2.7T   31T   9% /mnt/nfs
```

3.7 步骤 7——集群动态增减

关键术语

集群动态增减是指增加或减少 osd 节点，以达到增加集群容量或者缩减集群容量的目的。

扩容/缩减技术

1. 自动扩容

随着数据量的增加，集群系统必然面临扩容的问题。我们强烈建议在系统容量到达 85%之前，就要对系统进行扩容。

我们提供了在线扩容的方案，即在集群系统运行状态下，根据扩容的需求增加一定数量的 osd。我们建议新增 osd 时，按照一块硬盘对应一个 osd 的方法进行增加。

具体操作步骤如下：

1、 在新机器上，按照第四章所述安装系统。

2、 在集群中创建一个 osd。

```
[root@mon_0 ~]# icfs osd create
```

该命令在 mon_0 上执行，会打印出所增加的 osd 编号，这个编号将会在后续的步骤中用到。

3、 在新的机器上创建 osd 挂载目录。

```
[root@osd_2 ~]# mkdir -p /data/osd.2
```

4、 在新的机器上格式化设备，然后挂载设备。

```
[root@osd_2 ~]# mkfs -t xfs /dev/sdb
[root@osd_2 ~]# mount -t xfs /dev/sdb /data/osd.2/
```

5、 在 mon_0 上修改集群配置文件/etc/icfs/icfs.conf。增加内容如下：

```
[osd.2]
host = osd_2
devs = /dev/sdb
```

6、 将该配置文件 icfs.conf 向所有 mon、osd、mds 都拷贝一份。

```
[root@mon_0 mon.0]# scp /etc/icfs/icfs.conf mon_1:/etc/icfs/
```

7、 在新增的 osd 上初始化 osd 数据目录。

```
[root@osd_2 ~]# icfs-osd -i 2 --mkfs –mkkey
```

该目录在初始化是必须为空，否则会报错。其中，

8、 注册 osd 的授权 key。

```
[root@osd_2 ~]# icfs auth add osd.2 osd 'allow *' mon 'allow rwx'
-i /data/osd.2/keyring
```

9、 添加 osd 到 CRUSH map 中，这样 osd 就可以开始接受数据。

```
[root@mon_0 mon.0]# icfs osd crush set osd.2 1.000 root=default
```

10、 该 osd 已经添加到集群的运行配置中了，但是该 osd 还没有开始工作。osd 的状态为 down 且 out 的，需要在新增机器上启动 osd 进程。

```
[root@osd_2 ~]# service icfs start osd.2
```

11、 在 mon_0 上查看集群状态。

2. 容量缩减

1、 在 mon_0 上将该 osd.0 从集群系统中移出

```
[root@mon_0 ~]# icfs osd out osd.0
marked out osd.0.
```

2、 在该 osd 所在的主机，关闭该 osd 的进程

```
[root@osd_0 ~]# /icfs/src/icfs/src/init-icfs stop osd.0
=== osd.0 ===
Stopping Icfs osd.0 on osd_0...kill 6266...kill 6266...done
```

3、 在 mon_0 节点，将该 osd 的信息从 crush map 中清除

```
[root@mon_0 ~]# icfs osd crush remove osd.0
removed item id 0 name 'osd.0' from crush map
```

4、 在 mon_0 节点，将该 osd 的认证信息（key）删除

```
[root@mon_0 ~]# icfs auth del osd.0
Updated
```

5、 在 mon_0 节点，将该 osd 删除

```
[root@mon_0 ~]# icfs osd rm osd.0
removed osd.0
```

6、 修改 icfs.conf 配置文件，将该 osd 的信息注释或者删除

```
[root@mon_0 ~]# icfs osd rm osd.0
removed osd.0
vim icfs.conf
#[osd.0]
#       host = osd_0
#       devs = /dev/sda
```

7、 将修改后的 icfs.conf 文件拷贝到其他的 osd 节点上。

```
[root@mon_0 ~]# scp /etc/icfs/icfs.conf
osd_1:/etc/icfs/icfs.conf
```