



思科互动网络——技术达人“秀” 科技互动 给力分享

了解我们

思科互动网络—技术达人“秀”是一档全新的网络视频节目。它通过网络视频、专家在线互动、网络研讨会，以及各大热点技术专题，给大家带来独特的视听感受。

技术达人“秀”的“每一集”都汇聚思科专家及技术达人于一堂，向观众提供热点技术话题，进行互动讨论。

并在每集视频节目直播中，通过思科专家聊天室提出问题，得到相应解答；每集视频直播节目结束之后，既可以通过视频点播方式进行节目回放，也可以参加相同问题的网络主题研讨会。

思科互动网络—技术达人“秀”将与您分享收获。

了解更多详情，请登录活动官网：

http://www.cisco.com/web/CN/solutions/industry/segment_sol/enterprise/programs/interaction_network.html

思科MDS 下一代存储网络的新基准

1

00:00:03,500 --> 00:00:08,750

[Robb]稍微近一点，再近一点

2

00:00:08,750 --> 00:00:12,040

好的。我们快准备好了。更多精彩马上到来！这里是技术达人秀

3

00:00:26,760 --> 00:00:28,980

[Robb]嗨，大家好！欢迎来到思科技术达人秀。我是Robb Boyd

4

00:00:28,980 --> 00:00:30,980

[Jimmy Ray]我是Jimmy Ray Purser

5

00:00:30,980 --> 00:00:34,480

[Robb]今天的问题是—都是关于存储，虽然是

6

00:00:34,480 --> 00:00:36,310

[Jimmy Ray]老生长谈>>但也很有趣

7

00:00:36,310 --> 00:00:38,700

如果我没记错的话，你被弄得有点猝不及防

8

00:00:38,700 --> 00:00:42,890

让我们回到伦敦一月份的Cisco Live

9

00:00:42,890 --> 00:00:47,800

你当时在跟Nitin交谈，Nitin等会儿会是我们节目开场的嘉宾

10

00:00:47,800 --> 00:00:51,400

在当时—你说了一句话—如果我没记错的话—

11

00:00:51,400 --> 00:00:54,290

你说你不确定思科在存储交换机方面在做了些什么

12

00:00:54,290 --> 00:00:57,090

当时并没有对此谈论太多[Jimmy Ray]我那时对它了解不多

13

00:00:57,090 --> 00:00:59,320

我貌似难住他了—我想问一个问题，伙计，M-Disk还在使用吗？

14

00:00:59,320 --> 00:01:03,960

我们还在那个平台上继续开发吗？>>用MDS？

15

00:01:03,960 --> 00:01:06,090

是的，因为它太棒了！>>嗯

16

00:01:06,090 --> 00:01:08,380

笑声在哪里？在哪里？

17

00:01:08,380 --> 00:01:12,390

第二个问题，我们的竞争对手经常说

18

00:01:12,390 --> 00:01:15,790

思科并没有把光纤通道当回事儿，因为

19

00:01:15,790 --> 00:01:19,230

问题的第二部分是，我们只关注SEOE

20

00:01:19,230 --> 00:01:23,010

这才是我们的大事，如果脱离SEOE，你就不需要思科了

21

00:01:23,010 --> 00:01:26,540

所以，我想知道。这是真的吗？我的意思是，说真的

22

00:01:26,540 --> 00:01:28,850

因为如果真是这样，工程师们需要知道这些

23

00:01:28,850 --> 00:01:32,880

内容来开展设计工作。这个问题非常关键。

24

00:01:32,880 --> 00:01:36,800

我们真的需要知道—当我们设计网络的时候—实际情况是怎样的？

25

00:01:36,800 --> 00:01:40,080

所以今天我们就深入探讨一下，这会是个很不错的问題，因为

26

00:01:40,080 --> 00:01:42,930

对我来说很显然的是，因为在我为这期做的准备的时候，

27

00:01:42,930 --> 00:01:44,930

可以有一点点的伏笔

28

00:01:44,930 --> 00:01:47,340

当时他回答了你的问题。给了你直接的答案

29

00:01:47,340 --> 00:01:50,910

如果大家想看的话，我们有视频，但是当时他并没有提到

30

00:01:50,910 --> 00:01:52,910

他们即将推出的产品。>>是的

31

00:01:52,910 --> 00:01:54,930

但是今天，我们有产品发布

32

00:01:54,930 --> 00:01:57,360

我们今天要展示一些很重要的硬件

33

00:01:57,360 --> 00:02:00,390

他们已经着手交付这些硬件；所以许多他那时讲的东西

34

00:02:00,390 --> 00:02:04,980

很有趣，因为我们今天要讲的其实早在2002年

35

00:02:04,980 --> 00:02:08,479

就已经开始了—10多年以前>>对

36

00:02:08,479 --> 00:02:14,530

我们现在看到的是一个巨大的升级，这很可能需要再有11、12年

37

00:02:14,530 --> 00:02:16,890

才能轻松掌握。你同意吗？

38

00:02:16,890 --> 00:02:23,990

恩，同意。但是我真的很想，你知道，很想同Mark等人讨论一下

39

00:02:23,990 --> 00:02:27,450

去了解最真实的内容

40

00:02:27,450 --> 00:02:30,220

你们还做了什么呢？>>换句话说，你们做了什么改变呢？

41

00:02:30,220 --> 00:02:32,220

对！>>比如，为什么要升级呢？>>老实说，我的意思是，好吧

42

00:02:32,220 --> 00:02:36,140

你想要更多端口，更多容量。>>速度

43

00:02:36,140 --> 00:02:40,230

但是它的结构已经很强了，还存在什么问题吗？

44

00:02:40,230 --> 00:02:42,410

这个问题好。>>我真的很想知道

45

00:02:42,410 --> 00:02:45,470

现在仍然是多协议的天下，是吧？>>是的，哦，哥们儿，你搞笑呢吧？

当然

46

00:02:45,470 --> 00:02:48,180

只要我们都想用十进制系统，那就太多了

47

00:02:48,180 --> 00:02:51,000

就是这些。>>因为你不觉得如果你回过头来看

48

00:02:51,000 --> 00:02:55,090

当我们过去以思科的眼光讨论数据存储和交换的时候

49

00:02:55,090 --> 00:02:57,580

如果我们说过：10年之后会是什么样子？

50

00:02:57,580 --> 00:03:00,420

那时我们会说：呃，思科这时候将会

51

00:03:00,420 --> 00:03:02,910

推动单协议市场的发展>>同意

52

00:03:02,910 --> 00:03:05,320

[Robb]那个时候我们真的很希望它会发生>>嗯

53

00:03:05,320 --> 00:03:08,730

不是我们对现在的状况不满，因为

54

00:03:08,730 --> 00:03:11,490

客户网络是这么设计的，我想会继续有更好的东西从中产生的

55

00:03:11,490 --> 00:03:15,210

但是，这很有意思，一边提升这个，一边开始回想

56

00:03:15,210 --> 00:03:19,210

十年之前我们谈论存储的时候是怎样一种心态呢？>>对

57

00:03:19,210 --> 00:03:21,970

我们说过，哦，不，我们推动了FCoE，现在一切都朝着以太网发展

58

00:03:21,970 --> 00:03:24,080

现在是100%的以太网世界了>>恩啊

59

00:03:24,080 --> 00:03:26,800

我敢肯定如果当时，10年之前，你问

60

00:03:26,800 --> 00:03:30,140

多少人还会运行IPv4，可能得到答案是一样的

61

00:03:30,140 --> 00:03:32,510

他们很可能会回答，不，10年之后可能就是v6了

62

00:03:32,510 --> 00:03:34,620

呃，v6一直是这几年来，每年

63

00:03:34,620 --> 00:03:36,830

最重要的新产品发布了，不是吗？>>它不厉害吗？

64

00:03:36,830 --> 00:03:38,950

当然厉害。把这个问题留到下一个节目

65

00:03:38,950 --> 00:03:40,980

哦，它还会在另外一个节目中亮相，是吗？>>当然

66

00:03:40,980 --> 00:03:43,220

这个节目结束之后，我们会赶在夏天结束之前做那个节目

67

00:03:43,220 --> 00:03:46,400

好了，伙计们，不要走开，接下来我要请来一个特殊的嘉宾

68

00:03:46,400 --> 00:03:48,900

我们将要对这次发布做详细地介绍

69

00:03:48,900 --> 00:03:51,940

我们将从思科的视角，了解这一市场

70

00:03:51,940 --> 00:03:54,490

我们还会再次邀请Nitin

71

00:03:54,490 --> 00:03:57,530

不过不是在伦敦的Cisco Live，就是在我们这个节目

72

00:03:57,530 --> 00:04:00,160

给我们大家讲述我们要发布的究竟是什么产品，以及和我们之间的关系

73

00:04:00,160 --> 00:04:02,620

我们该怎么用，然后我们会详细介绍

74

00:04:02,620 --> 00:04:05,220

一些这一全新硬件不可思议的，耀眼的性能

75

00:04:05,220 --> 00:04:07,220

千万不要走开！

76

00:04:07,220 --> 00:04:08,700

思科技术达人秀

77

00:04:08,700 --> 00:04:10,950

你好，Nitin，很高兴在美国再次见到你

78

00:04:10,950 --> 00:04:12,950

[Nitin Garg]见到你也很高兴

79

00:04:12,950 --> 00:04:15,750

[Robb]欢迎来到技术达人秀，欢迎来到演播室，因为

80

00:04:15,750 --> 00:04:19,519

上一次—我在做研究的时候，我想起来了那次谈话

81

00:04:19,519 --> 00:04:22,530

当时我就站在这儿，在镜头外边—你和

82

00:04:22,530 --> 00:04:25,730

Jimmy Ray 在伦敦Cisco Live 预测一些事情的发展

83

00:04:25,730 --> 00:04:28,800

这些事情现在已经很明显了，我那时觉得你们的谈论中没有

84

00:04:28,800 --> 00:04:30,870

漏掉任何东西，但是还是很高兴一些东西又转回来了

85

00:04:30,870 --> 00:04:32,870

所以我对这个节目特别兴奋

86

00:04:32,870 --> 00:04:35,240

让我们来谈谈MDS，思科在做什么

87

00:04:35,240 --> 00:04:37,940

因为我们在这儿对此已经讨论很长时间了

88

00:04:37,940 --> 00:04:40,500

以你来看，有哪些发展趋势需要我们明白呢？

89

00:04:40,500 --> 00:04:44,350

[Nitin Garg]呃，首先，在Cisco Live的交谈很棒

90

00:04:44,350 --> 00:04:48,160

我很高兴能继续这个话题>>哦，太好了

91

00:04:48,160 --> 00:04:52,270

对。其实今天我计划让你做一些介绍

92

00:04:52,270 --> 00:04:55,720

但是，在此之前，我想跟你谈谈我们投资的发展趋势

93

00:04:55,720 --> 00:04:59,550

以及思科是如何做的。>>好的

94

00:04:59,550 --> 00:05:02,690

我们对这些产品的发布真的非常激动

95

00:05:02,690 --> 00:05:04,790

我们先来谈谈发展趋势

96

00:05:04,790 --> 00:05:09,020

我们看到的在数据中心影响存储局域网的存储趋势

97

00:05:09,020 --> 00:05:14,690

首先是，数据中心的服务器虚拟化

98

00:05:14,690 --> 00:05:17,000

正在急剧增长

99

00:05:17,000 --> 00:05:19,250

现在人人都在谈论服务器虚拟化

100

00:05:19,250 --> 00:05:21,480

[Robb]确实是产生了影响>>恩，有影响

101

00:05:21,480 --> 00:05:26,180

它产生的影响是随着越来越多的虚拟机器在服务器上运行

102

00:05:26,180 --> 00:05:29,230

会需要更多的带宽>>对

103

00:05:29,230 --> 00:05:34,710

还需要更大的多协议性能

104

00:05:34,710 --> 00:05:37,820

因为服务器上运行有不同的工作负载量

105

00:05:37,820 --> 00:05:42,600

[Robb]有道理>>对，此外，还需要简化管理

106

00:05:42,600 --> 00:05:49,250

因为客户们习惯了管理单个的服务器

107

00:05:49,250 --> 00:05:52,590

他们可能会理解单个机器上的流量，但是，现在

108

00:05:52,590 --> 00:05:57,560

把所有的虚拟计算机都集中在一个服务器上，这些流量就会变得

109

00:05:57,560 --> 00:06:01,820

不可预测了，所以他们需要管理方案来管理

110

00:06:01,820 --> 00:06:04,310

这些流量。>>有意思。>>对

111

00:06:04,310 --> 00:06:07,990

这是第一个影响—服务器虚拟化

112

00:06:07,990 --> 00:06:09,990

这是服务器虚拟化的影响

113

00:06:09,990 --> 00:06:15,070

第二个趋势是存储或者说是数据—大家都在谈论大数据

114

00:06:15,070 --> 00:06:18,400

巨大的数据。大数据确实实在不断增长

115

00:06:18,400 --> 00:06:24,460

数据正在成倍增长，每两年就翻一番。每两年数据就增加两倍

116

00:06:24,460 --> 00:06:30,110

这一点非常重要，因为客户们—

117

00:06:30,110 --> 00:06:34,960

看到存储的这个增长，他们就对数据中心不那么肯定了>>当然

118

00:06:34,960 --> 00:06:39,600

恩，这也意味着从块存储—传统的文件

119

00:06:39,600 --> 00:06:43,510

到光纤通道存储的发展趋势。这也是FCoE发挥作用的地方

120

00:06:43,510 --> 00:06:46,510

也是NAS起作用的

121

00:06:46,510 --> 00:06:52,240

所以客户想要多协议连接和能实现与同在一个服务器

122

00:06:52,240 --> 00:06:57,990

或他们购买的同一网络上的

123

00:06:57,990 --> 00:07:02,630

其它多协议存储环境进行无缝连接的性能

124

00:07:02,630 --> 00:07:05,150

这是第二个趋势

125

00:07:05,150 --> 00:07:10,200

第三个趋势就是固态硬盘。>>所有的存储器类型，是的>>正是

126

00:07:10,200 --> 00:07:15,530

低延迟。所以他们想要高带宽来应对低延迟环境

127

00:07:15,530 --> 00:07:20,020

所以他们从传统的存储区域转向了固态区域>>真的吗？

128

00:07:20,020 --> 00:07:23,860

有意思。>>第四个趋势是，他们

129

00:07:23,860 --> 00:07:27,170

正在将这一切带到超级数据中心或大数据中心

130

00:07:27,170 --> 00:07:33,440

现在的状况是，有了这些大型的数据中心，客户们

131

00:07:33,440 --> 00:07:38,000

可以将他们不同的工作负载量合并到

132

00:07:38,000 --> 00:07:41,100

同一导向器的不同服务器上>>太有趣了

133

00:07:41,100 --> 00:07:43,480

这样，所有的鸡蛋都放到同一个篮子。>>正是

134

00:07:43,480 --> 00:07:45,480

所以可靠度就尤其重要了。>>当然

135

00:07:45,480 --> 00:07:48,110

所以高可用性很重要

136

00:07:48,110 --> 00:07:50,800

并不只是应用程序减少了

137

00:07:50,800 --> 00:07:56,610

现在你还得确定该性能没有影响到这些应用程序

138

00:07:56,610 --> 00:08:00,050

这太疯狂了！所以这是我们前所未闻的规模

139

00:08:00,050 --> 00:08:06,750

虚拟化产生了这种不稳定性，同时也使我们不能

140

00:08:06,750 --> 00:08:09,210

及时地对某一时间发生的状况有一个真实确切的感受

141

00:08:09,210 --> 00:08:12,230

你不想到时候手足无措。因为它比你想象的要发展的快

142

00:08:12,230 --> 00:08:15,490

所以当你对你的交换设施

143

00:08:15,490 --> 00:08:18,670

网络设计做某一决定时—你必须得考虑到

144

00:08:18,670 --> 00:08:21,960

你真的不能够知道你将要为什么做准备

145

00:08:21,960 --> 00:08:24,730

因为这些不单单是只是决定—人们不可能改变

146

00:08:24,730 --> 00:08:26,730

我们每年谈论的东西，对吧？>>是的

147

00:08:26,730 --> 00:08:29,940

所以你不能轻易在中途做出更改。这是个重大的决定

148

00:08:29,940 --> 00:08:34,299

我们有客户一直在这上面部署了5—8年—

149

00:08:34,299 --> 00:08:41,580

一直不停地在数据中心运行。MDS已经运行了相当长一段时间了

150

00:08:41,580 --> 00:08:45,230

那么，思科对这件事是怎么看的呢？

151

00:08:45,230 --> 00:08:47,350

问得好。我们说过了趋势

152

00:08:47,350 --> 00:08:54,050

思科打算对数据中心进行整体的投资组合

153

00:08:54,050 --> 00:08:57,830

你听说过UCS的统一计算吧？>>当然听过

154

00:08:57,830 --> 00:09:03,070

思科的投资中重点在存储网络

155

00:09:03,070 --> 00:09:07,510

局域网—数据中心传统的以太网环境

156

00:09:07,510 --> 00:09:10,050

还有统一管理上

157

00:09:10,050 --> 00:09:14,420

所以这就意味着，客户可以降低

158

00:09:14,420 --> 00:09:18,300

部署应用程序的成本，不需要运行不同的存储器

159

00:09:18,300 --> 00:09:22,250

他们可以在整个组合内使用同样的管理

160

00:09:22,250 --> 00:09:24,280

他们可以在整个组合内使用同样的操作系统

161

00:09:24,280 --> 00:09:26,990

我觉得，在数据中心合并中，这应该也很有效果

162

00:09:26,990 --> 00:09:30,060

绝对的>>因为，你不会充分从我们

163

00:09:30,060 --> 00:09:32,670

目前能做的产品中受益，如果你不合并的话

164

00:09:32,670 --> 00:09:34,670

好吧，好的。这些东西聚在一起，真的很好

165

00:09:34,670 --> 00:09:38,020

所以可能这是一个自然的引入，因为你不能正确地进行你的工作

166

00:09:38,020 --> 00:09:40,150

如果你不像我认为的那样把它建立起来的话

167

00:09:40,150 --> 00:09:42,150

我们开发了新的产品>>对

168

00:09:42,150 --> 00:09:44,180

它们在其中运行的好吗？还是根本就不行？>>对

169

00:09:44,180 --> 00:09:47,840

对！所以思科目前正在投资开发MDS，我刚刚谈了数据中心

170

00:09:47,840 --> 00:09:51,060

思科正在开发最好的产品

171

00:09:51,060 --> 00:09:53,710

MDS就是其中之一

172

00:09:53,710 --> 00:09:56,020

我们在2002年发布了MDS

173

00:09:56,020 --> 00:09:58,610

它当时就像是一个市场破坏者

174

00:09:58,610 --> 00:10:01,720

那个时候，市场上还有很多存储局域网络公司

175

00:10:01,720 --> 00:10:04,840

但是思科进入了市场以后

176

00:10:04,840 --> 00:10:07,450

现在只剩下2个公司了

177

00:10:07,450 --> 00:10:10,040

市场出现了许多合并

178

00:10:10,040 --> 00:10:16,530

所以思科数据中心的MDS做了创新—比如VSAN，安全性，FCoE—

179

00:10:16,530 --> 00:10:20,820

同时还提供了投资保护

180

00:10:20,820 --> 00:10:27,620

带着客户从2G线速率到4G到8G线速率再到FCoE

181

00:10:27,620 --> 00:10:32,160

客户可以在他们之前引进的平台上对这些创新进行部署

182

00:10:32,160 --> 00:10:35,750

还可以得到更高的带宽

183

00:10:35,750 --> 00:10:38,050

就此，我们是怎么看FCoE？

184

00:10:38,050 --> 00:10:40,050

我们对此很支持，很显然。>>对

185

00:10:40,050 --> 00:10:42,870

FCoE是数据中心的一个重要部分

186

00:10:42,870 --> 00:10:45,900

我们还对光纤通道进行了投资

187

00:10:45,900 --> 00:10:50,680

所以我们今天要介绍的是，下一步的飞跃发展

188

00:10:50,680 --> 00:10:54,780

我们通过引进MDS9710和9250i

189

00:10:54,780 --> 00:10:58,980

将把存储网络发展至一个更高的水平

190

00:10:58,980 --> 00:11:01,500

两个平台>>说说怎么会有这么大的作用呢

191

00:11:01,500 --> 00:11:03,660

先说9710吧

192

00:11:03,660 --> 00:11:07,330

9710有三个我们想要着重强调的属性

193

00:11:07,330 --> 00:11:09,470

第一是性能

194

00:11:09,470 --> 00:11:13,260

它是目前市场上导向器的三倍性能>>哇！

195

00:11:13,260 --> 00:11:16,870

目前的市场上的产品是每秒512GB

196

00:11:16,870 --> 00:11:20,970

但是9710要比这快上三倍>>也就是1.5TB每秒

197

00:11:20,970 --> 00:11:23,580

超过三倍>>哇哦！太棒了

198

00:11:23,580 --> 00:11:30,180

我们同时还引进了一个48端口、16GB的线卡

199

00:11:30,180 --> 00:11:33,820

好多端口。那肯定是一个很大的结构吧>>是的

200

00:11:33,820 --> 00:11:38,690

所以48端口、16GB线速率—也就是第一天就是每秒768GB

201

00:11:38,690 --> 00:11:42,920

比某些产品多出50%的线速率端口

202

00:11:42,920 --> 00:11:46,530

现在有了768，可以将它提至1.5TB每秒

203

00:11:46,530 --> 00:11:49,510

这可以让你给客户id提供100%的增长

204

00:11:49,510 --> 00:11:54,350

客户可以在现在配置9710，还可以为未来留有增长空间

205

00:11:54,350 --> 00:11:56,380

太棒了。这貌似就是他们需要的东西>>是的

206

00:11:56,380 --> 00:11:59,630

第二个关键属性是高可用性

207

00:11:59,630 --> 00:12:03,960

我们将9710的高可用性提升至了更高水平

208

00:12:03,960 --> 00:12:09,830

我们正在提供机架内的交换冗余

209

00:12:09,830 --> 00:12:14,560

所以你知道在后面有供交换的结构模块。>>对

210

00:12:14,560 --> 00:12:19,370

客户可以有N+1冗余，所以即使有一个交换模块不行了

211

00:12:19,370 --> 00:12:22,040

对性能并不产生影响

212

00:12:22,040 --> 00:12:25,100

不仅对应用程序不产生可用性影响

213

00:12:25,100 --> 00:12:27,480

还对性能没有任何影响

214

00:12:27,480 --> 00:12:30,800

第三个属性。我们只剩一分钟了。>>第三个属性是灵活性

215

00:12:30,800 --> 00:12:35,820

这是FC、FCoE、FCoE线速率的重要的属性

216

00:12:35,820 --> 00:12:41,160

另外一个我们要介绍的平台是9250i>>哦，这个是什么呢？

217

00:12:41,160 --> 00:12:45,380

9250i 是一个非常酷、非常有创新的一个产品，我们对此很是兴奋

218

00:12:45,380 --> 00:12:49,220

它为统一结构提供存储服务

219

00:12:49,220 --> 00:12:53,760

不管是房子的光纤通道还是FCoE都可以

220

00:12:53,760 --> 00:12:56,520

因为我们既有光纤通道，又有FCoE端口

221

00:12:56,520 --> 00:13:03,820

还有10GB大的FCIP，可以在整个数据库内处理的数据

222

00:13:03,820 --> 00:13:08,530

所以，当你为恢复数据或保持商业业务连续而复制数据时

223

00:13:08,530 --> 00:13:13,770

你可以使用它—这些大的FCIP

224

00:13:13,770 --> 00:13:16,000

为FC和FCoE提供SAN扩展。 >>太好了

225

00:13:16,000 --> 00:13:19,460

你可以运行数据迁移服务

226

00:13:19,460 --> 00:13:25,160

还可以运行I/O加速度服务

227

00:13:25,160 --> 00:13:29,400

使你的数据中心进一步与同步应用程序分离

228

00:13:29,400 --> 00:13:33,320

所以在这个平台上同时为

229

00:13:33,320 --> 00:13:36,700

FC和FCoE 设备提供了关键的存储服务

230

00:13:36,700 --> 00:13:39,420

哇哦！呃，我们的时间刚刚好

231

00:13:39,420 --> 00:13:42,610

今天我们要谈到两件大事>>是的

232

00:13:42,610 --> 00:13:45,620

我们下一步的目标—我想Jimmy Ray马上就会谈到这个

233

00:13:45,620 --> 00:13:49,500

这真是一个漂亮的交换机

234

00:13:49,500 --> 00:13:52,710

我很期待，就是缺了轮子和方向盘把它带到这儿

235

00:13:52,710 --> 00:13:54,270

但是无论如何我们接下来会这样做的。>>谢谢

236

00:13:54,270 --> 00:13:56,540

非常感谢你，Nitin。感谢你的到来>>谢谢你抽出宝贵的时间

237

00:13:58,190 --> 00:14:00,290

Prashant Jain ，欢迎来到技术达人秀，伙计

238

00:14:00,290 --> 00:14:02,410

很高兴来到这里，我很兴奋

239

00:14:02,410 --> 00:14:04,940

你能来我也很高兴，因为带着硬件来的嘉宾

240

00:14:04,940 --> 00:14:07,070

正是我们想邀请的嘉宾>>绝对是

241

00:14:07,070 --> 00:14:09,450

我们今天带来了好多硬件

242

00:14:09,450 --> 00:14:12,370

你带来了像爱荷华级战列舰一样的东西，伙计！

243

00:14:12,370 --> 00:14:14,640

这个真的是不同寻常

244

00:14:14,640 --> 00:14:16,960
我们把它称作什么—9710导向器？

245

00:14:16,960 --> 00:14:19,170
正是，这就是9710导向器

246

00:14:19,170 --> 00:14:22,380
这是MDS9000家族的新成员

247

00:14:22,380 --> 00:14:26,520
增加这位新成员，我们感到非常骄傲，因为

248

00:14:26,520 --> 00:14:30,780
数据中心正经历着许多变化

249

00:14:30,780 --> 00:14:34,020
数据中心发展了许多。存储量、带宽需求

250

00:14:34,020 --> 00:14:36,020

都在突飞猛涨

251

00:14:36,020 --> 00:14:40,740

这是达到数据中心需求的解决方案

252

00:14:40,740 --> 00:14:43,950

不仅是今天、明天，而是10年以后都可以

253

00:14:43,950 --> 00:14:46,900

那，现在，你说的口气不小，所以我要检验一下

254

00:14:46,900 --> 00:14:49,640

你带了幻灯片过来，我们先浏览一下幻灯片

255

00:14:49,640 --> 00:14:52,480

因为你说的其中一件事就是

256

00:14:52,480 --> 00:14:56,830

—你的产品具有最好的性能，最可靠—一口气不小—

257

00:14:56,830 --> 00:15:01,130

无与伦比的灵活性，还有

258

00:15:01,130 --> 00:15:04,560

投资保护这一块是—我觉得熟悉MDS系列的人知道

259

00:15:04,560 --> 00:15:06,560

—它已经存在有一段时间了

260

00:15:06,560 --> 00:15:09,460

而且在不断增长，变得越来越好

261

00:15:09,460 --> 00:15:12,490

这是整个行业可以说是

262

00:15:12,490 --> 00:15:17,660

我们对存储方面做出改变的产品，经过发展——其实它起了带头作用

263

00:15:17,660 --> 00:15:22,550

当然。我们十几年前就开发了9509

264

00:15:22,550 --> 00:15:28,070

现在有一些客户仍然在运行9509——正常运行时间8年——

265

00:15:28,070 --> 00:15:30,570

八年间不曾停过一秒

266

00:15:30,570 --> 00:15:34,220

我之前有过这样的客户。跟这情况一模一样

267

00:15:34,220 --> 00:15:38,890

当然。我们在设计这个产品的时候，利用了这个强大的基础

268

00:15:38,890 --> 00:15:42,630

做出了不可思议的成就

269

00:15:42,630 --> 00:15:48,040

呃，口气不小呢，因为你对它进行增强改进的时候

270

00:15:48,040 --> 00:15:50,040

那就不可思议了

271

00:15:50,040 --> 00:15:53,570

我们从头开始吧。这儿你要讲的是性能

272

00:15:53,570 --> 00:15:57,370

[Prashant]说到性能，我来给你列你几组数据

273

00:15:57,370 --> 00:16:05,170

我们支持24TB背道交换能力，一天

274

00:16:05,170 --> 00:16:08,970

每槽1.5TB交换力

275

00:16:08,970 --> 00:16:15,830

因此你可以在这个线速率上运行384个16GB端口，一天

276

00:16:15,830 --> 00:16:18,590

这只是其中一部分

277

00:16:18,590 --> 00:16:22,420

这个机架可以基本上支持N+1冗余

278

00:16:22,420 --> 00:16:25,490

与9513相比，它有许多附属项目

279

00:16:25,490 --> 00:16:31,900

所以，同时，性能与操作一致

280

00:16:31,900 --> 00:16:37,780

所以习惯管理9513的客户—与9513外观一样，在他们管理9710的时候

感觉也是一样的

281

00:16:37,780 --> 00:16:41,400

所以不需要学习新东西

282

00:16:41,400 --> 00:16:46,640

他们可以扩展到24TB—不需要大幅度升级，这很重要

283

00:16:46,640 --> 00:16:51,220

从8GB转到10GB，再转到不管下一代连通性需要多少GB

284

00:16:51,220 --> 00:16:54,920

他们不需要推倒重来>>哇哦！

285

00:16:54,920 --> 00:16:56,920

这是前所未有的，前所未有

286

00:16:56,920 --> 00:17:01,980

想想都觉得不可思议，在机架，我可以跳到24T

287

00:17:01,980 --> 00:17:05,940

甚至不需要打开这个东西；相信我，与这些一样大的东西

288

00:17:05,940 --> 00:17:07,829

你不会想拎着它们到处跑的，对吧？>>正是

289

00:17:07,829 --> 00:17:10,940

我们可以断言—基本上，你一旦装上了它

290

00:17:10,940 --> 00:17:16,589

它可以满足你未来10年甚至更长时间内的需求

291

00:17:16,589 --> 00:17:20,109

随着数据中心性能和可扩展性的需求

292

00:17:20,109 --> 00:17:22,109

呃，这是我们知道的关于数据中心的一件事情，是吗？

293

00:17:22,109 --> 00:17:24,990

它们就像是孩子，我的意思是说，它们开始很小，后来慢慢长大

294

00:17:24,990 --> 00:17:27,930

它们会一直发展—拥有自己的事情，自己的小癖好

295

00:17:27,930 --> 00:17:30,420

这就是数据中心做的事情

296

00:17:30,420 --> 00:17:33,190

我一直告诉思科技术达人秀的观众朋友—

297

00:17:33,190 --> 00:17:35,660

伙计，你真的在寻找一个事业吗？

298

00:17:35,660 --> 00:17:37,870

数据中心就是就一个好地方，伙计，这是肯定的

299

00:17:37,870 --> 00:17:41,480

那是存储。[Prashant]它一直都能使人振奋激动

300

00:17:41,480 --> 00:17:44,990

总有一些技术出现解决我们的需求问题

301

00:17:44,990 --> 00:17:50,040

但是你需要足够灵活的设备来帮助你转移

302

00:17:50,040 --> 00:17:53,670

慢慢随着你的需求变化发展>>同意

303

00:17:53,670 --> 00:17:56,340

呃，那我们谈谈其中最重要的吧—因为我们在谈数据

304

00:17:56,340 --> 00:17:59,270

数据访问、运行时间、可靠性等等—

305

00:17:59,270 --> 00:18:03,150

这是这个东西中很大很大的一部分，尤其是我们讨论

306

00:18:03,150 --> 00:18:08,500

一串不同的无延迟或无丢失兼容型协议的时候

307

00:18:08,500 --> 00:18:12,790

可靠性，伙计—速度是一个事情，但是可靠性—是完全不同的另一件事情

308

00:18:12,790 --> 00:18:14,790

对我们来说，它是前沿和中心

309

00:18:14,790 --> 00:18:19,510

在数据中心，基本上我们需要零停工期。不停止运作

310

00:18:19,510 --> 00:18:22,430

所以9509—特别可靠

311

00:18:22,430 --> 00:18:26,950

9513已经为许多客户不间断地工作了许多年了

312

00:18:26,950 --> 00:18:29,240

但对我们来说，这还不够好

313

00:18:29,240 --> 00:18:34,200

我们想着这个结构，想如何能使

314

00:18:34,200 --> 00:18:37,380

这个平台的每一个构成部分冗余

315

00:18:37,380 --> 00:18:41,200

我们找到了结构；这是该领域第一个

316

00:18:41,200 --> 00:18:43,780

结构上拥有N+1冗余的平台

317

00:18:43,780 --> 00:18:48,420

我们有冗余的风扇树，所以基本上

318

00:18:48,420 --> 00:18:51,160

每一个这样的导向器都有内置冗余

319

00:18:51,160 --> 00:18:55,210

当你讨论这些的时候，这个单元背后的东西等

320

00:18:55,210 --> 00:18:58,230

因为我们—我们会拍一些漂亮的照片

321

00:18:58,230 --> 00:19:00,260

这块工作由Steve负责，他会给我们拍一些漂亮的照片

322

00:19:00,260 --> 00:19:04,490

我继续，冒昧地拉出了风扇盘

323

00:19:04,490 --> 00:19:07,820

这也是一个重型产品—4个大风扇盘

324

00:19:07,820 --> 00:19:11,550

它们像是你放在喷气式飞机或其他地方的翼扇

325

00:19:11,550 --> 00:19:16,780

很好的风扇，这是它后面的交换结构

326

00:19:16,780 --> 00:19:22,180

交换结构也给我们提供了多余的容量和性能

327

00:19:22,180 --> 00:19:24,790

所以我们只需要很快的开启就行

328

00:19:24,790 --> 00:19:26,890

[Prashant]我稍稍介绍一下风扇

329

00:19:26,890 --> 00:19:29,210

在后面通常有3个风扇盘

330

00:19:29,210 --> 00:19:31,260

每一个风扇盘上有四个风扇

331

00:19:31,260 --> 00:19:34,410

每一个风扇都是冗余的，所以一个风扇盘不运转了—

332

00:19:34,410 --> 00:19:36,470

系统可以照常运行

333

00:19:36,470 --> 00:19:38,720

你有足够的时间进去，替换它

334

00:19:38,720 --> 00:19:42,520

不止如此，每一个风扇盘都有冗余的控制和动力部分

335

00:19:42,520 --> 00:19:45,180

所以即使其中一部分不运作也没有关系>>转到摄像机

336

00:19:45,180 --> 00:19:47,910

我们这里要谈的是这里的两个白色的连接器

337

00:19:47,910 --> 00:19:51,640

[Prashant]其实他们也是冗余的，其中一个也可以出故障

338

00:19:51,640 --> 00:19:54,210

这样风扇盘仍可以继续运作

339

00:19:54,210 --> 00:20:00,260

所以4个风扇，其中一个可以停止作用，而风扇盘仍可以继续工作

340

00:20:00,260 --> 00:20:03,720

所以冗余是多层次的

341

00:20:03,720 --> 00:20:07,600

我们再来看看交换结构—这是我们感到非常骄傲的地方

342

00:20:07,600 --> 00:20:14,440

它可以处理256GB光纤通道流量的交换

343

00:20:14,440 --> 00:20:19,430

每一个交换结构后都有6个交换板卡插槽

344

00:20:19,430 --> 00:20:23,070

所以这也就意味着你只需要三个交换结构

345

00:20:23,070 --> 00:20:28,690

在机架中运行，就可以支撑你在目前的连接性的线速率上运行了

346

00:20:28,690 --> 00:20:31,280

[Jimmy Ray] 你们有6个？>>是，我们有6个，所以这有很大的提升空间

347

00:20:31,280 --> 00:20:34,540

安装4个交换结构，N+1保护

348

00:20:34,540 --> 00:20:43,100

3个交换结构是768 Gbps，6个就是1.5TB，因此提升的空间非常大

349

00:20:43,100 --> 00:20:46,280

>>你不是开玩笑呢吧。这太棒了

350

00:20:46,280 --> 00:20:48,590

我想加快点速度，因为我们的时间不够了

351

00:20:48,590 --> 00:20:52,420

但很显然，MDS迷们还有想知道的事情

352

00:20:52,420 --> 00:20:56,900

软件升级不中断、域故障降低—这是什么？

353

00:20:56,900 --> 00:21:00,860

[Prashant] 这是该平台其中一个重要的结构

354

00:21:00,860 --> 00:21:04,260

域故障降低是因为你有8个线卡

355

00:21:04,260 --> 00:21:08,530

例如，你可以将一个端口通道

356

00:21:08,530 --> 00:21:10,650

分配到这8个线卡上

357

00:21:10,650 --> 00:21:16,030

这样，即使其中一个线卡出现故障了，在极端的情况下，

358

00:21:16,030 --> 00:21:18,280

也不会影响到端口通道

359

00:21:18,280 --> 00:21:21,290

它只影响到该端口通道带宽的1/8

360

00:21:21,290 --> 00:21:23,710

一切都照常运行

361

00:21:23,710 --> 00:21:27,200

类似地，线卡有内置的CRC

362

00:21:27,200 --> 00:21:30,380

一旦数据包进入，就开始运行CRC

363

00:21:30,380 --> 00:21:33,390

一旦它检查过底板，它就返回线卡

364

00:21:33,390 --> 00:21:37,050

以确定数据中心内没有遭到破坏的帧到处游离

365

00:21:37,050 --> 00:21:39,380

>>你们现在仍然在线速率上做这个？

366

00:21:39,380 --> 00:21:41,630

线速率误差校正。>>是的

367

00:21:41,630 --> 00:21:45,940

哇，了不起！[Prashant]我们拿一个线卡出来

368

00:21:45,940 --> 00:21:50,500

[Jimmy Ray] 拿出来，伙计！

369

00:21:50,500 --> 00:21:54,400

[Prashant] 你看，非常复杂精密的设备，里面蕴含着好多技术

370

00:21:54,400 --> 00:21:56,910

许多高水准区块

371

00:21:56,910 --> 00:22:00,890

前面你看到的这些是所有的IO Slice ASICs

372

00:22:00,890 --> 00:22:06,050

它们不仅有MAC的功能，还运行转发、虚拟输出队列

373

00:22:06,050 --> 00:22:12,270

后面两个纵横式交换矩阵用来帮助把768 Gbps 带宽推向底板

374

00:22:12,270 --> 00:22:16,370

也就是位于SUP卡上的中央仲裁器

375

00:22:16,370 --> 00:22:20,200

然后在线卡上的仲裁器确定

376

00:22:20,200 --> 00:22:25,730

所有穿过机架的东西都已受过检查—装配了从入口到出口的部件

377

00:22:25,730 --> 00:22:30,190

嘿，现在，大家看一看这个卡，从工程师

378

00:22:30,190 --> 00:22:32,390

电力工程师的角度看，你要看的最重要东西之一

379

00:22:32,390 --> 00:22:34,900

当你看一个卡的整体布局的时候

380

00:22:34,900 --> 00:22:37,840

就是上面有多少组件，间距是什么类型的

381

00:22:37,840 --> 00:22:41,740

因为间距是散热的主要途径

382

00:22:41,740 --> 00:22:43,990

组件越多

383

00:22:43,990 --> 00:22:47,720

故障点就越多，所以好的设计

384

00:22:47,720 --> 00:22:50,350

都在尽量减少组件的数量，给你留下

385

00:22:50,350 --> 00:22:53,750

尽可能多的绿色空间。你们做得真的太令人钦佩了

386

00:22:53,750 --> 00:22:56,890

这是一款非常非常好的设计

387

00:22:56,890 --> 00:23:00,610

性能如此好，板上又如此干净利落

388

00:23:00,610 --> 00:23:02,970

哇哦，伙计！向这些工程师们致敬！

389

00:23:02,970 --> 00:23:05,590

[Prashant]同意

390

00:23:05,590 --> 00:23:10,490

在这些端口上我们支持许多不同的I/O 连接

391

00:23:10,490 --> 00:23:14,800

这有48个端口—你可以在任意端口、任何地方运行从2GB光纤通道

392

00:23:14,800 --> 00:23:18,840

到16GB光纤通道、10GB光纤通道

393

00:23:18,840 --> 00:23:23,820

所有都有线速率。你可以装配8GB短波、长波、ER镜片

394

00:23:23,820 --> 00:23:28,800

或者16GB短波、长波镜片—许许多多CS可用的镜片

395

00:23:28,800 --> 00:23:31,340

随着我们不断深入，还会有更多>>太酷了

396

00:23:31,340 --> 00:23:33,300

这里还有其他什么卡？

397

00:23:33,300 --> 00:23:40,150

[Prashant]另外一件我想展示的东西是管理器

398

00:23:40,150 --> 00:23:45,760

如果你拿这个同9500平台比较的话

399

00:23:45,760 --> 00:23:52,720

就增加该平台的可用性和可扩展性来说

400

00:23:52,720 --> 00:23:55,260

这个管理器已经在多极水平上消失了

401

00:23:55,260 --> 00:23:58,310

与2GB相比，现在它可以支撑8GB的存储，

402

00:23:58,310 --> 00:24:03,910

这是内存槽—4核CPU，运行速度更快

403

00:24:03,910 --> 00:24:07,360

与32比特相比，支持64比特

404

00:24:07,360 --> 00:24:09,400

[Jimmy Ray]哇！这是个很大的不同啊>>绝对是

405

00:24:09,400 --> 00:24:13,740

同样，这也是不仅仅是为客户短期的需求设计的

406

00:24:13,740 --> 00:24:17,510

同样可以支持一年后的需求—而且还可以支持未来十年的客户需求

407

00:24:17,510 --> 00:24:21,260

所以，所有的都是按照支持客户未来十年的

408

00:24:21,260 --> 00:24:25,650

数据中心转换需求而设计与构造的

409

00:24:25,650 --> 00:24:29,680

>>看这儿—让我迅速把它放到摄像头下

410

00:24:29,680 --> 00:24:34,120

它看起来可能没什么，可是看看这块板的厚度，伙计们

411

00:24:34,120 --> 00:24:39,490

但这确实显示了我们的多层板上的可靠性类型

412

00:24:39,490 --> 00:24:43,830

哇！这是我第一次看见把这个东西拿出来看

413

00:24:43,830 --> 00:24:46,450

我真的很想好好秀给大家，但是我可能把时间用完了

414

00:24:46,450 --> 00:24:49,190

[Prashant] 这真是一个令人兴奋的精密的设计

415

00:24:49,190 --> 00:24:52,140

最后我要展示的是电力供应

416

00:24:52,140 --> 00:24:57,110

现在，看看我们的思科产品家族，很多重心都放在

417

00:24:57,110 --> 00:25:00,030

在不同的平台之间共享组件上

418

00:25:00,030 --> 00:25:02,710

这是一个3000瓦的电力供应系统

419

00:25:02,710 --> 00:25:04,780

它只需要一根电缆

420

00:25:04,780 --> 00:25:09,570

客户可能会问：“那我为什么会有8个电源架呢？”

421

00:25:09,570 --> 00:25:12,230

这样可以允许你进行扩展

422

00:25:12,230 --> 00:25:16,130

第一天机架全负荷工作也只需要3个电源

423

00:25:16,130 --> 00:25:20,350

[Jimmy Ray]哦，是真的吗？>>是的，全部只需要3个电源

424

00:25:20,350 --> 00:25:23,600

我们会装配6个，使客户有极大的冗余

425

00:25:23,600 --> 00:25:27,640

但是，同时，我们同时支持AC和DC电源

426

00:25:27,640 --> 00:25:30,980

所以有极大的灵活性，满足客户的需求

427

00:25:30,980 --> 00:25:33,840

你们能看见这个吗？你能快速看一下吗？

428

00:25:33,840 --> 00:25:36,070

看这这电缆多粗，伙计

429

00:25:36,070 --> 00:25:39,670

你们不要乱弄>>不，可靠性、可用性

430

00:25:39,670 --> 00:25:42,050

[Jimmy Ray]真的非常非常酷，伙计。真的很酷

431

00:25:42,050 --> 00:25:46,810

呃，我知道我们已经超时了，但是很快的，伙计

432

00:25:46,810 --> 00:25:49,220

我想说几件事情

433

00:25:49,220 --> 00:25:52,870

你们开发出了许多出色的解决方案，我们在这里也见到了

434

00:25:52,870 --> 00:25:58,240

是导向器的一部分，但是会很快地，我们已经讲过数据

435

00:25:58,240 --> 00:26:01,300

[Prashant]当然。在线卡上，一般来说，重要的是

436

00:26:01,300 --> 00:26:07,210

有了这些线卡上的缓冲容量，我们可以一直走510千米

437

00:26:07,210 --> 00:26:11,160

所以我们有带有企业许可证的4095缓冲区-缓冲区信用

438

00:26:11,160 --> 00:26:15,070

[Jimmy Ray] 很显然，这一张很重要

439

00:26:15,070 --> 00:26:17,450

就像我刚才说的一样，我知道我们超时了，我向观众朋友们道歉

440

00:26:17,450 --> 00:26:22,560

但是这很显然是一个很好的图，来展示我们的发展是由

441

00:26:22,560 --> 00:26:24,560

我们拥有交换卡的数量决定的

442

00:26:24,560 --> 00:26:27,460

你可以看到你们之前在讨论的不可思议的可扩展性

443

00:26:27,460 --> 00:26:35,040

很快—最后，我们的管理方式是什么，如何比较

444

00:26:35,040 --> 00:26:41,390

多好的工作啊！>>是的>>真的是工程策划和设计的一大进步

445

00:26:41,390 --> 00:26:46,010

呃，这是—我能怎么说呢？这是一个伟大的产品

446

00:26:46,010 --> 00:26:48,570

能邀你来到我们节目，我真是高兴坏了—实际上

447

00:26:48,570 --> 00:26:50,760

我们—关了摄像头，我就要拿一个螺丝刀

448

00:26:50,760 --> 00:26:53,210

再深入研究一下>>来到这里我很高兴

449

00:26:53,210 --> 00:26:55,500

能与这个接触产品我也很兴奋。谢谢你

450

00:26:57,440 --> 00:27:00,760

Mark Allen ，欢迎来到技术达人秀 ，哥们>> Jimmy Ray ，谢谢你邀请我们

451

00:27:00,760 --> 00:27:04,730

我很感谢>>你开玩笑呢吧？开始就是交叉开关的构建图

452

00:27:04,730 --> 00:27:09,370

这一部分肯定很精彩

453

00:27:09,370 --> 00:27:12,300

呃，你能这样说我很高兴，因为这正是我今天要讲的

454

00:27:12,300 --> 00:27:17,170

我们谈过了MDS9710,大家现在都看着我们说

455

00:27:17,170 --> 00:27:19,560

“ 你们要开发出新的导向器了”

456

00:27:19,560 --> 00:27:21,560

但是我想强调的一点是，我们不是要推出新的导向器

457

00:27:21,560 --> 00:27:24,030

而可能是一个新的机架

458

00:27:24,030 --> 00:27:28,450

但是从结构上来说，我们并没有与2002年

459

00:27:28,450 --> 00:27:31,680

即11年前，MDS首次面世的时候构建的结构有很大差别

460

00:27:31,680 --> 00:27:34,560

我们开发了我们维修至今的存储交换的一个结构

461

00:27:34,560 --> 00:27:38,000

这是一个基于可靠性的结构

462

00:27:38,000 --> 00:27:40,950

一个基于数据保护的结构

463

00:27:40,950 --> 00:27:45,060

一个基于性能的结构—整个交换结构性能均等

464

00:27:45,060 --> 00:27:48,030

这些才是存储局域网真正有用的东西

465

00:27:48,030 --> 00:27:52,660

一些人想要谈论速度和容量，9710显然是一个

466

00:27:52,660 --> 00:27:56,340

运转速度非常高的导向器，但是它同样有高可靠性和可预测性

467

00:27:56,340 --> 00:27:59,190

这才是我们今天用结构要谈到的方面

468

00:27:59,190 --> 00:28:03,940

我觉得它之所以重要，是因为它——赛车比汽车要快得多

469

00:28:03,940 --> 00:28:07,410

但很显然，运载东西不方便

470

00:28:07,410 --> 00:28:10,680

我需要既能运载东西

471

00:28:10,680 --> 00:28:14,320

又有速度的东西，所以从这儿开始讲挺好

472

00:28:14,320 --> 00:28:18,070

但是你还有这些又大、词儿又多、又复杂的幻灯片，你要给我们

473

00:28:18,070 --> 00:28:23,110

详细的介绍一下MDS系列产品

474

00:28:23,110 --> 00:28:25,460

而不是单单只讲我们现在谈的全新交换机

475

00:28:25,460 --> 00:28:32,190

但是这战略、方法适用于所有MDS。>>对

476

00:28:32,190 --> 00:28:37,040

从9100系列MDS交换机，芯片上的交换机

477

00:28:37,040 --> 00:28:39,620

位于ASIC内部的就是这个结构

478

00:28:39,620 --> 00:28:43,150

如果你看看11年前发布的MDS9500系列

479

00:28:43,150 --> 00:28:48,050

你就知道它是在此结构上开发出来的，9710，同样，也是基于此结构

480

00:28:48,050 --> 00:28:51,950

现在 , 经过这么多年 , 我们也做了改变—这些可能在不同的ASIC中构建

481

00:28:51,950 --> 00:28:56,000

现在它在单个ASIC上就完成了—但是根本上 , 还是同样的功能

482

00:28:56,000 --> 00:28:59,370

而且它已经存在11年了 , 这是开发9710

483

00:28:59,370 --> 00:29:01,600

依照的结构和传统

484

00:29:01,600 --> 00:29:05,680

它是为该领域内需要最大的可靠性和

485

00:29:05,680 --> 00:29:09,390

性能预测性客户设计的

486

00:29:09,390 --> 00:29:12,350

关于这个设计要说的真的很多

487

00:29:12,350 --> 00:29:15,910

我知道我们没有时间坐在这里，谈论工程原理什么的

488

00:29:15,910 --> 00:29:19,930

但是我们可以多谈谈一个非常好的原则

489

00:29:19,930 --> 00:29:25,120

它几乎快成了一个设计规律了—如今一切都是基于此

490

00:29:25,120 --> 00:29:28,560

而且它真的能使你的产品更加成熟、更加先进

491

00:29:28,560 --> 00:29:32,450

真的可以使你继续发展下去，而不是要从零开始

492

00:29:32,450 --> 00:29:36,400

退到最开始建构东西，有些人就是这么做的

493

00:29:36,400 --> 00:29:40,120

我不想打击任何人，但是有了这种设计理念，真的能从中受益很多

494

00:29:40,120 --> 00:29:43,410

所以，总之，你们在说什么—我们怎么开始呢？

495

00:29:43,410 --> 00:29:47,120

我们—这里我们的第一块—我们在看什么呢？

496

00:29:47,120 --> 00:29:49,940

[Mark Allen]谈到数据包流的时候，显然我们要涉及到接口

497

00:29:49,940 --> 00:29:53,390

这些是所有的光接口，所以我们有一它被称为PHY

498

00:29:53,390 --> 00:29:56,540

它是一个光的、电的接口，也是SFP进入的地方

499

00:29:56,540 --> 00:30:00,330

或者你的SFP脉冲。这是光纤出来

500

00:30:00,330 --> 00:30:03,370

并将之转换为电子信号的过程

501

00:30:03,370 --> 00:30:07,230

这是第一个阶段，但是真正的核心在5 MAC层

502

00:30:07,230 --> 00:30:12,150

这是做初始数据包检测的地方

503

00:30:12,150 --> 00:30:14,530

当一个帧进入MDS中时

504

00:30:14,530 --> 00:30:18,400

其实我们对它进行检测，检验CRC，这是第一个功能

505

00:30:18,400 --> 00:30:20,560

为什么这个很重要？

506

00:30:20,560 --> 00:30:23,740

在数据中心，数据是最重要的东西

507

00:30:23,740 --> 00:30:25,920

显然，我们想要尽可能地保护它们

508

00:30:25,920 --> 00:30:29,720

所以如果一个帧进入，它是不好的CRC

509

00:30:29,720 --> 00:30:31,720

那就意味着这个帧被损坏

510

00:30:31,720 --> 00:30:34,400

我们就要丢弃它。我们不想再让它继续穿过交换结构

511

00:30:34,400 --> 00:30:36,520

我们不想让它损耗资源和交换机

512

00:30:36,520 --> 00:30:39,480

我们也不想让损耗接收器上的资源

513

00:30:39,480 --> 00:30:43,000

我们想丢弃它，因为它已经无效了

514

00:30:43,000 --> 00:30:45,900

所以，第一个阶段就是验证CRC

515

00:30:45,900 --> 00:30:48,040

如果是无效的话，那就丢弃

516

00:30:48,040 --> 00:30:52,540

接下来，我们假装EISL是数据头

517

00:30:52,540 --> 00:30:57,530

这个数据头真的包含虚拟结构信息

518

00:30:57,530 --> 00:31:00,780

虚拟结构是VSAN的T11标准

519

00:31:00,780 --> 00:31:06,280

实际上，我们的VSAN变成了T11虚拟结构标准

520

00:31:06,280 --> 00:31:10,520

然后它们就变成手动的了，我们假装VASN在里面

521

00:31:10,520 --> 00:31:13,690

这样我们就知道它进入的是哪一个虚拟结构

522

00:31:13,690 --> 00:31:16,180

这时候我们就可以标记QoS

523

00:31:16,180 --> 00:31:18,870

另一件重要的事情是我们给这个帧打上时间标记

524

00:31:18,870 --> 00:31:21,240

[Jimmy Ray]好的，就是这一点。知道了

525

00:31:21,240 --> 00:31:24,690

[Mark Allen]一旦我们给它打上时间标记，我们就可以对它在交换机里的时间进行追踪

526

00:31:24,690 --> 00:31:29,130

这对光纤通道标准非常重要

527

00:31:29,130 --> 00:31:31,660

因为我们只能在结构里保留一个帧这么长时间—>>是的，对

528

00:31:31,660 --> 00:31:36,630

它很重要，如果结构中出现的问题，就会出现阻塞的情况

529

00:31:36,630 --> 00:31:40,480

运行慢的设备可能会导致帧回退

530

00:31:40,480 --> 00:31:44,110

因为我们有了时间标记，我们就可以辨认出什么时候这些帧

531

00:31:44,110 --> 00:31:49,650

在交换机中呆的时间过长，并提醒操作员潜在的危險

532

00:31:49,650 --> 00:31:53,240

因此，第一阶段这种信息

533

00:31:53,240 --> 00:31:55,480

非常重要 >>好的

534

00:31:55,480 --> 00:31:59,490

从这里，一旦帧被验证有效，也被加上印记以后

535

00:31:59,490 --> 00:32:03,390

我们就进入到了转发决策>>好，现在我们在这里。明白了

536

00:32:03,390 --> 00:32:08,230

在我们的转发决策中，有几件事一定会发生

537

00:32:08,230 --> 00:32:10,260

首先，是安全

538

00:32:10,260 --> 00:32:12,340

在光纤通道中，有一种东西要分区

539

00:32:12,340 --> 00:32:15,150

大家知道这是个什么东西。它是ACL—访问控制列表

540

00:32:15,150 --> 00:32:18,830

这是决定该帧是否可以从启动器到接收器

541

00:32:18,830 --> 00:32:22,000

从源头到目的转发的硬件执行

542

00:32:22,000 --> 00:32:24,230

如果不能，我们就丢弃它

543

00:32:24,230 --> 00:32:26,520

所以，再说一次，它是被阻挡的感冒病毒

544

00:32:26,520 --> 00:32:28,670

你知道吗，还有其它分区的办法

545

00:32:28,670 --> 00:32:30,500

在光纤通道，你可以做名称服务器

546

00:32:30,500 --> 00:32:34,070

你可以不告诉每一个人你可以到那，就可以执行分区

547

00:32:34,070 --> 00:32:36,910

但是你可以证明它。如果你知道某人想要去哪里

548

00:32:36,910 --> 00:32:39,290

无论如何他们都可以经过交换机的>>当然

549

00:32:39,290 --> 00:32:41,560

我们可以通过硬分区阻止它

550

00:32:41,560 --> 00:32:44,070

所以，嘭。作为转发决策的一部分就完成了

551

00:32:44,070 --> 00:32:47,000

我们在VASN经过的地方制定转发决策

552

00:32:47,000 --> 00:32:51,410

这可以是多个VSAN共享的一个设备而来

553

00:32:51,410 --> 00:32:53,560

这是进行中的SAN面试的一部分

554

00:32:53,560 --> 00:32:55,740

在这里你还可以在VSAN之间安排路径

555

00:32:55,740 --> 00:33:01,020

我们还可以做IVR NAT，还可以在这里更改域ID

556

00:33:01,020 --> 00:33:04,830

我们可以在2个交换结构中间穿梭，而只用一个域ID号码

557

00:33:04,830 --> 00:33:07,620

当2个东西合并的时候>>我想说在合并状态时

558

00:33:07,620 --> 00:33:09,810

这是个非常重要的东西。>>正是

559

00:33:09,810 --> 00:33:11,990

你可能没有重新更换整个交换结构的地址的选择

560

00:33:11,990 --> 00:33:15,910

所以在第一个阶段拥有在硬件中做NAT的能力

561

00:33:15,910 --> 00:33:17,990

可以帮你解决这个问题>>你说真的!

562

00:33:17,990 --> 00:33:21,600

一旦我们知道了它要去哪儿，我们就可以继续我们的排队机制

563

00:33:21,600 --> 00:33:25,240

这是MDS中另一个重要的部分

564

00:33:25,240 --> 00:33:27,650

我们有虚拟输出队列

565

00:33:27,650 --> 00:33:32,250

基本上，每一组源头/目的，我们都会建立一个输出队列

566

00:33:32,250 --> 00:33:36,470

所以，如果我有一个主机，在三个不同的接收器上，或者也许是这个主机

567

00:33:36,470 --> 00:33:40,120

在交换机之间经过三个不同的ISL，那么每一个路径

568

00:33:40,120 --> 00:33:42,230

都有一个单独的队列

569

00:33:42,230 --> 00:33:46,010

这就是说，它们可以分别处理，以免出现

570

00:33:46,010 --> 00:33:50,150

一个服务器同时对话好几个磁盘或者如果磁带版本较低、运行和处理都比较慢运行较慢的磁带驱动器

571

00:33:50,150 --> 00:33:53,410

导致的运行较慢的磁带驱动器的情况，这样的话我们就不会被运行较慢的磁带阻塞

572

00:33:53,410 --> 00:33:56,090

或者其他的帧就不会阻塞进入磁盘

573

00:33:56,090 --> 00:33:59,000

同样，这也是通道阻塞的最基本的功能之一

574

00:33:59,000 --> 00:34:03,210

它是在排队机制中完成的

575

00:34:03,210 --> 00:34:07,010

所以当我们最终加工完帧，并将它通过底板和转换结构转换时

576

00:34:07,010 --> 00:34:13,420

我们可以给

577

00:34:13,420 --> 00:34:15,420

可以处理的帧适当的处理和优先权

578

00:34:15,420 --> 00:34:17,880

如果这个帧被拦截，那我们就不对它进行处理。转而转向下一个帧

579

00:34:17,880 --> 00:34:20,010

很好，很好

580

00:34:20,010 --> 00:34:22,010

排队的功能很重要

581

00:34:22,010 --> 00:34:26,010

一旦我们排了队，这就是中央仲裁器在交换机发挥作用的时候了

582

00:34:26,010 --> 00:34:29,670

中央仲裁器—我把它描述成“天眼”

583

00:34:29,670 --> 00:34:33,929

就是在这个时候，交换机可以审视、确认所有在交换机内

584

00:34:33,929 --> 00:34:36,100

帧要往哪里去

585

00:34:36,100 --> 00:34:40,300

所以当我们问—我们给仲裁器发了个请求

586

00:34:40,300 --> 00:34:43,639

说我们需要从这个存储区传输到这个目的地的时候

587

00:34:43,639 --> 00:34:46,139

仲裁器通常会回答：“我给你一个权限，告诉你走哪条路径”

588

00:34:46,139 --> 00:34:50,610

因为中央仲裁器知道交换机内的每一样东西、每一条路径

589

00:34:50,610 --> 00:34:53,060

不可能会发生阻塞

590

00:34:53,060 --> 00:34:56,429

你一般会有一个确保的路径，所以不会发生

591

00:34:56,429 --> 00:34:58,840

在底板发生拥挤的情况

592

00:34:58,840 --> 00:35:05,090

一些竞争者会说，因为你们在这里有一个中央仲裁器

593

00:35:05,090 --> 00:35:10,330

通过交叉开关做的这件事，他们会说

594

00:35:10,330 --> 00:35:15,650

这种方法已经老了，这是队头。在这里会产生许多队头阻塞的

595

00:35:15,650 --> 00:35:18,790

但是你可以对他们的质疑说不。因为VOQ阻止了这一混乱的发生

596

00:35:18,790 --> 00:35:22,700

加之这里还有那么多人工智能—这根本就不是一个可以争论的问题

597

00:35:22,700 --> 00:35:25,500

这根本就不是一个可以争论的问题，最重要的是，这也是我们为什么

598

00:35:25,500 --> 00:35:30,750

设置不均等性能的原因。不管我是否是从同一端口ASIC的两个相邻端口
开始

599

00:35:30,750 --> 00:35:33,290

不管我是否要到同一线卡还是经过

600

00:35:33,290 --> 00:35:37,730

机架内任何一个线卡，都没有关系，整个机架内的处理都是均衡的

601

00:35:37,730 --> 00:35:41,800

所以，对于一个标准的、想要拥有可预测的性能的管理员来说

602

00:35:41,800 --> 00:35:45,930

不需要担心—呃，如果我插进了这个端口，到那个端口去

603

00:35:45,930 --> 00:35:47,930

要是有其他流量的话，会出现问题吗？

604

00:35:47,930 --> 00:35:51,500

它会消失，因此，再一次，MDS的基础

605

00:35:51,500 --> 00:35:53,800

就是它们可预测的性能

606

00:35:53,800 --> 00:35:56,360

这就是我们如何通过中央仲裁器达到的>>对，非常正确

607

00:35:56,360 --> 00:36:01,430

因此一旦我们恢复了许可，我们把它发往交叉交换结构

608

00:36:01,430 --> 00:36:05,110

现在，那就是在芯片上的交换机了

609

00:36:05,110 --> 00:36:09,350

它可以—其实在这个芯片上的交换机中有两个内置交叉开关

610

00:36:09,350 --> 00:36:13,390

在9500系列中，我们有2个交换机构可以在管理器上

611

00:36:13,390 --> 00:36:16,410

或者在单独的交换结构模块上，现在在9700中

612

00:36:16,410 --> 00:36:18,460

我们所做的就是赋予你们更多

613

00:36:18,460 --> 00:36:21,960

我们现在不是有2个交换结构，而是3个交换结构模块

614

00:36:21,960 --> 00:36:24,520

而这还可以扩展到6个交换结构模块

615

00:36:24,520 --> 00:36:28,520

所以我们扩展性发展地很成功

616

00:36:28,520 --> 00:36:31,270

但我们还有散发数据的能力

617

00:36:31,270 --> 00:36:33,270

Nitin已经讲了N+1

618

00:36:33,270 --> 00:36:37,010

在这里我们实现了。如果我有3个路径，仲裁器将会通过3个路径发送

619

00:36:37,010 --> 00:36:39,220

如果我有4个路径，它就会通过4个路径发送

620

00:36:39,220 --> 00:36:42,150

增加一个，是无缝整合，如果出现故障了

621

00:36:42,150 --> 00:36:45,790

在出故障的组件周围依然可以无缝工作

622

00:36:45,790 --> 00:36:48,100

这是自动恢复>>的确是，它是透明的

623

00:36:48,100 --> 00:36:51,440

它是透明的—其实其中并没有恢复，因为如果这个东西离开了

624

00:36:51,440 --> 00:36:56,160

仲裁器就会知道它不是一个有效路径，也就不会再给这条路径授予权限了

625

00:36:56,160 --> 00:36:59,690

太有意思了。是真的>>对，它就应该是这样子的

626

00:36:59,690 --> 00:37:02,090

无缝操作>>对，对，明白了

627

00:37:02,090 --> 00:37:05,910

当它又回来的时候，我们也回到转发引擎

628

00:37:05,910 --> 00:37:10,460

现在，很有趣，你知道吗，我们再一次做ACL

629

00:37:10,460 --> 00:37:12,700

如果这是从另一台交换机上分离出来的，可能是在交互操作的状况下

630

00:37:12,700 --> 00:37:16,520

我们会再次验证分区，所以我们又有了硬件执行ACL

631

00:37:16,520 --> 00:37:19,320

我们还要重新验证CRC

632

00:37:19,320 --> 00:37:21,860

有人会说这么做太过了，但是

633

00:37:21,860 --> 00:37:23,650

[Jimmy Ray]哦，你在这完成以后，在这里重新验证？

634

00:37:23,650 --> 00:37:26,120

[Mark Allen] 完全正确>>为什么？>>呃，因为我们可能做了变动

635

00:37:26,120 --> 00:37:30,500

如果是IVR NAT，我们改变了帧>>当然

636

00:37:30,500 --> 00:37:33,480

所以我们要确定我们没有破坏它

637

00:37:33,480 --> 00:37:35,650

或者没有组件出现故障

638

00:37:35,650 --> 00:37:40,130

或者其它情况。但是某些会出现故障

639

00:37:40,130 --> 00:37:42,810

在这种远程案例，我们要重新验证

640

00:37:42,810 --> 00:37:45,060

如果它又是有毛病的，那我们就丢弃它

641

00:37:45,060 --> 00:37:51,680

最后，一旦我们验证了硬分区并验证了—我们必须有更多NAT

642

00:37:51,680 --> 00:37:53,850

我们也可以在出口那一面做NAT

643

00:37:53,850 --> 00:37:55,980

我们还有5-MAC可以进行转换呢

644

00:37:55,980 --> 00:37:58,060

我们看着我们的计时器

645

00:37:58,060 --> 00:38:00,710

如果帧在交换机中的时间过长，我们也可以在这丢弃它

646

00:38:00,710 --> 00:38:03,100

否则，我们就可以继续进行，对它进行转换

647

00:38:03,100 --> 00:38:05,100

而且，很棒的事情会发生

648

00:38:05,100 --> 00:38:07,780

我们可以移去内部队头，这样终端设备就不会知道

649

00:38:07,780 --> 00:38:09,780

我们对虚拟交换结构做了什么

650

00:38:09,780 --> 00:38:12,390

如果我们要去另外一个MDS交换机，我们还有ESL队头

651

00:38:12,390 --> 00:38:14,960

它仍然具备VSAN信息、QoS 信息

652

00:38:14,960 --> 00:38:17,040

而且会被继续转换到下一个交换机上

653

00:38:17,040 --> 00:38:19,360

所以，MDS上要进行好多过程

654

00:38:19,360 --> 00:38:23,950

但是所有的这些最终是要将该领域的MDS分开

655

00:38:23,950 --> 00:38:26,540

另外，它也是在线速率上进行的>>全是关于线速率，正是

656

00:38:26,540 --> 00:38:29,340

现在我知道我们真的没有时间了

657

00:38:29,340 --> 00:38:31,380

但是我还想再讲最后一张幻灯片

658

00:38:31,380 --> 00:38:35,480

我保证，我会讲得非常快

659

00:38:35,480 --> 00:38:38,920

我们看的是同一个东西吗？你想让这张幻灯片给我们展示什么呢？

660

00:38:38,920 --> 00:38:41,610

[Mark Allen] 如果你真在看线卡，你想知道帧是如何流动的

661

00:38:41,610 --> 00:38:44,600

我们谈了时髦精美的MAC还有排队机制

662

00:38:44,600 --> 00:38:47,300

这些都是在IO Slice ASIC 这里完成的

663

00:38:47,300 --> 00:38:50,860

你想谈谈仲裁—我们有一个中心点，这个中心点将所有的这些仲裁请求

664

00:38:50,860 --> 00:38:54,200

集中到线卡上，然后再到服务器上进行读取

665

00:38:54,200 --> 00:38:57,370

我们有我们的交换结构模块，如果我们真的进行单步调试

666

00:38:57,370 --> 00:38:59,420

你就可以看到它是如何工作了

667

00:38:59,420 --> 00:39:04,950

我们继续前进，点击它，一个帧就进来了，然后我们向仲裁器发出请求

668

00:39:04,950 --> 00:39:08,940

得到批准。一旦我们知道了路径，可以是任意一条路径

669

00:39:08,940 --> 00:39:11,480

穿越线卡或者交换结构

670

00:39:11,480 --> 00:39:13,770

然后帧就出去了

671

00:39:13,770 --> 00:39:18,270

所以，是的，它是有一点噱头，但是它—>>不，它很酷

672

00:39:18,270 --> 00:39:20,480

这就是它的工作原理。呃，Mark，我告诉你

673

00:39:20,480 --> 00:39:22,480

你和我可以在这一直继续下去，我知道

674

00:39:22,480 --> 00:39:26,110

我们浏览了一下，但是最重要的事情是我们还有

675

00:39:26,110 --> 00:39:30,370

我们在这上面花费了这么长时间，我们已经将它

676

00:39:30,370 --> 00:39:32,370

放在思科网站上好好保存了

677

00:39:32,370 --> 00:39:35,170

你可以找到详细解释该结构的白皮书

678

00:39:35,170 --> 00:39:37,580

最重要的是，伙计，你们把这放到了你们的数据中心

679

00:39:37,580 --> 00:39:41,060

你们真的是马力十足，可靠性很高

680

00:39:41,060 --> 00:39:44,000

Mark，谢谢你兄弟。我很感谢你>>不客气，谢谢邀请我们过来

681

00:39:45,720 --> 00:39:47,750

Mousumi Bose，欢迎来到思科技术达人秀

682

00:39:47,750 --> 00:39:50,270

谢谢，Jimmy。谢谢你邀我来参加节目

683

00:39:50,270 --> 00:39:54,730

你能来真是太好了，尤其是你还带来了全新的9250i系列

684

00:39:54,730 --> 00:39:59,040

是的，我们对它走入市场感到特别兴奋

685

00:39:59,040 --> 00:40:06,770

本质上来说，这是从MDS9200交换机系列发展而来的新一代平台

686

00:40:06,770 --> 00:40:09,860

具有多业务能力

687

00:40:09,860 --> 00:40:20,090

本质上来讲，这是个很整洁的盒子。它是固定的2RU交换机，带有50个
端口

688

00:40:20,090 --> 00:40:23,660

在底部有48个端口

689

00:40:23,660 --> 00:40:29,070

上面有两个端口，这些支持多协议连接

690

00:40:29,070 --> 00:40:37,710

它是个很整齐的盒子，为你的所有SAN需求提供多协议连接

691

00:40:37,710 --> 00:40:45,580

这里我们有16GB光纤通道的40端口，所以这是线速率端口

692

00:40:45,580 --> 00:40:50,710

但是，当然，客户也可以在这些端口上使用8GB镜片

693

00:40:50,710 --> 00:40:53,510

哇，真的吗？太棒了

694

00:40:53,510 --> 00:41:01,450

右边这里，你会看到有8个像10GB线速率FCoE端口的端口

695

00:41:01,450 --> 00:41:08,170

这是一个大的改变，因为这是第一个同光纤通道、FCIP和FCoE结合起来的平台

696

00:41:08,170 --> 00:41:14,890

在左上方这里

697

00:41:14,890 --> 00:41:20,810

我们有2个支持SAN扩展能力的FCIP端口

698

00:41:20,810 --> 00:41:25,540

很棒的是，在这些端口上，客户们可以

699

00:41:25,540 --> 00:41:29,480

根据他们的需求，分别启动1GB或10GB

700

00:41:29,480 --> 00:41:33,600

因为正如我们所知，现在大部分的LAN链接都是1GB的

701

00:41:33,600 --> 00:41:39,010

但是许多客户其实寻求附加容量，使用比如10GB的连接

702

00:41:39,010 --> 00:41:45,370

所以我们在LAN为SAN扩展提供了1GB或10GB连接

703

00:41:45,370 --> 00:41:48,170

具有灵活性

704

00:41:48,170 --> 00:41:50,950

这就像是你把这个盒子放到了整个MDS策略中

705

00:41:50,950 --> 00:41:53,400

它就在原来的基础上进行了发展

706

00:41:53,400 --> 00:41:56,060

而且可以一直使用差不多10年左右

707

00:41:56,060 --> 00:42:01,220

绝对是。我们把MDS传统发展到了一个更高的水平

708

00:42:01,220 --> 00:42:04,500

我们的第一代产品已经很好地服务了客户10多年了

709

00:42:04,500 --> 00:42:08,430

有了这些新一代的产品，我们希望

710

00:42:08,430 --> 00:42:14,270

它们同样可以很好地服务客户，并将性能、可靠性和灵活性

711

00:42:14,270 --> 00:42:17,080

都提升到新的水平>>天啊，我想是这样的，是这样的

712

00:42:17,080 --> 00:42:22,220

把这个东西转个圈，让我们看看后面都有什么

713

00:42:22,220 --> 00:42:24,450

因为许多人想要知道我们是怎么设计这个东西的

714

00:42:24,450 --> 00:42:26,850

我们现在看的是什么？

715

00:42:26,850 --> 00:42:33,150

挺简单的。基本上，我们这里2个冗余风扇

716

00:42:33,150 --> 00:42:38,630

而且，这个产品还有一个很棒的特性，那就是我们在它的内部设置了前-后空气流向

717

00:42:38,630 --> 00:42:43,150

之前的产品要么是边-边或

718

00:42:43,150 --> 00:42:48,040

后-前流向，但是本产品设置了前-后流向

719

00:42:48,040 --> 00:42:51,790

这也是因为它是为同MDS9710导向器机架结合的大部分部署设计的

720

00:42:51,790 --> 00:42:56,880

而MDS9710导向器机架就是前-后流向

721

00:42:56,880 --> 00:43:02,640

所以为了方便数据中心规划，根据所有的冷热通道布局

722

00:43:02,640 --> 00:43:08,450

我们基本上使我们的空气流向设计同时符合这两种产品

723

00:43:08,450 --> 00:43:10,840

它们看起来也很像，真的

724

00:43:10,840 --> 00:43:13,830

是的，如果从颜色和外观上看的话

725

00:43:13,830 --> 00:43:16,630

所有新一代的产品看起来都很像

726

00:43:16,630 --> 00:43:20,920

现在我们还有3个电源，这里，1、2、3

727

00:43:20,920 --> 00:43:24,980

这些同样也是2+1冗余

728

00:43:24,980 --> 00:43:28,410

基本上它们使用的是300瓦特电力

729

00:43:28,410 --> 00:43:31,250

提供2+1冗余

730

00:43:31,250 --> 00:43:36,350

所以这是一个很简单的产品，这些都像是该领域内

731

00:43:36,350 --> 00:43:39,040

可穿组件和热插拔>>当然，当然

732

00:43:39,040 --> 00:43:42,780

但是，除此之外，这个产品没有很多移动的部分

733

00:43:42,780 --> 00:43:48,800

从本质上来讲，电源和风扇算是唯一移动的部分了

734

00:43:48,800 --> 00:43:54,860

前部是一个布局固定的机架，还有固定的端口

735

00:43:54,860 --> 00:44:03,390

客户使用时可以采用任何顺序，而且还可以灵活地

736

00:44:03,390 --> 00:44:07,500

把本产品当成纯粹的光纤通道交换机来使用

737

00:44:07,500 --> 00:44:11,050

因为它的光纤通道端口密度比较适合。我们有大概40个端口

738

00:44:11,050 --> 00:44:16,780

但是，慢慢地，正如我们所看到的那样，本产品的主要将用于

739

00:44:16,780 --> 00:44:22,930

SAN扩展能力上，这也是它主要的一个存储服务

740

00:44:22,930 --> 00:44:28,370

存储服务可以帮助客户保持业务连续性、恢复丢失的数据

741

00:44:28,370 --> 00:44:35,400

所以，从本质上来讲，它们可以做备份，通过WAN远距离传送将数据复制到SAN

742

00:44:35,400 --> 00:44:39,620

WAN使用的是开放标准、FCIP通道>>真好

743

00:44:39,620 --> 00:44:42,690

所以事实上这是本产品的最大用途

744

00:44:42,690 --> 00:44:49,670

我还想稍微说说开发该产品的主要目的

745

00:44:49,670 --> 00:44:51,700

那我们就换到下一张幻灯片

746

00:44:51,700 --> 00:44:56,510

正如你所见，它们是存储服务的单个平台

747

00:44:56,510 --> 00:45:01,370

如今思科已经有了一系列的存储平台

748

00:45:01,370 --> 00:45:05,600

我的意思是说，MDS是SAN领域最重要的产品

749

00:45:05,600 --> 00:45:09,480

但是我们也提供有其它产品，比如说Nexus 5000

750

00:45:09,480 --> 00:45:14,370

有做光纤通道和COE的统一的端口

751

00:45:14,370 --> 00:45:17,960

我们在Nexus 7K 有COE导向器

752

00:45:17,960 --> 00:45:20,680

为了提供智能存储服务

753

00:45:20,680 --> 00:45:25,570

而不是只看到这些平台有自己支持的服务或线卡

754

00:45:25,570 --> 00:45:30,640

我们开发出了这款产品，它本质上就是为了

755

00:45:30,640 --> 00:45:37,090

提供能够在所有思科存储产品间提供存储服务而设计的—

756

00:45:37,090 --> 00:45:40,560

主要是为MDS以及Nexus产品系列

757

00:45:40,560 --> 00:45:45,780

你可以把本产品同MDS或Nexus并用

758

00:45:45,780 --> 00:45:51,450

让交换机重新将流量导向该产品上，增加服务能力

759

00:45:51,450 --> 00:45:53,450

它们之间互不排斥

760

00:45:53,450 --> 00:45:55,380

其实它们还能联合到一块儿工作

761

00:45:55,380 --> 00:45:57,600

当然，当然

762

00:45:57,600 --> 00:46:01,580

这样，我们也能消除服务设备的无计划扩展

763

00:46:01,580 --> 00:46:05,360

我的意思是，我们不需要好几种设备，只需要

764

00:46:05,360 --> 00:46:08,980

使用本产品将所有的设备连接起来就可以了

765

00:46:08,980 --> 00:46:11,910

还有另外两种服务，我想谈一谈

766

00:46:11,910 --> 00:46:14,240

我已经说过FCIP了

767

00:46:14,240 --> 00:46:19,050

但是该产品还支持IO加速度和数据流动迁移

768

00:46:19,050 --> 00:46:22,900

这些是软件支持的2个特性

769

00:46:22,900 --> 00:46:25,590

现在这些其实是授权特性

770

00:46:25,590 --> 00:46:31,430

IO加速器，这么说吧，可以帮助你更快地通过WAN或MAN恢复数据

771

00:46:31,430 --> 00:46:38,230

和复制数据

772

00:46:38,230 --> 00:46:41,920

它还可以为你的数据和备份进行线路层加密

773

00:46:41,920 --> 00:46:46,250

这使得我们更加容易遵守法规

774

00:46:46,250 --> 00:46:52,160

我们还可以进行数据流动迁移，这是一个干净利落的功能

775

00:46:52,160 --> 00:46:56,320

可以帮助你在LAN或阵列之间迁移数据

776

00:46:56,320 --> 00:46:58,320

现在我们谈的像是异质阵列

777

00:46:58,320 --> 00:47:01,320

它不一定必须全部是光纤通道，你也可以把数据从光纤通道阵列迁移到

778

00:47:01,320 --> 00:47:04,210

FCoE 阵列

779

00:47:04,210 --> 00:47:08,440

还有一些该产品支持的简单整齐的特性

780

00:47:08,440 --> 00:47:12,760

客户可能需要额外的软件许可

781

00:47:12,760 --> 00:47:16,480

呃，是的，但是许可证确实使所有这些

782

00:47:16,480 --> 00:47:20,430

单独的设备 and 检验点消失—其它一切东西。这太不可思议了

783

00:47:20,430 --> 00:47:27,310

通过这两个端口使用，如FCIP和SAN扩展性能

784

00:47:27,310 --> 00:47:29,310

是不需要许可的

785

00:47:29,310 --> 00:47:33,160

基本的光纤通道或FCoE 也不需要许可

786

00:47:33,160 --> 00:47:38,150

我们只有IO加速器和DMM特性在执行中是需要

787

00:47:38,150 --> 00:47:40,860

授权的

788

00:47:40,860 --> 00:47:43,960

呃，如果你用不着的话，就不需要买，对吧？>>当然，当然

789

00:47:43,960 --> 00:47:46,460

我觉得这个盒子是个非常简洁整齐的产品

790

00:47:46,460 --> 00:47:52,860

它用经济高效的多协议连接

791

00:47:52,860 --> 00:47:55,700

将所有的这些存储服务结合到了一起>>我同意，同意

792

00:47:55,700 --> 00:47:58,180

我的意思是说，这也是我们想在本产品上实现的目标

793

00:47:58,180 --> 00:48:06,840

它可以在一个大数据中心与9710或者9500或Nexus存储导向器放在一起

794

00:48:06,840 --> 00:48:11,770

也可以在小的数据中心单独存在

795

00:48:11,770 --> 00:48:15,370

所以，对于小企业和商业客户来说

796

00:48:15,370 --> 00:48:18,070

只需要这样一个小盒子就完全可以了

797

00:48:18,070 --> 00:48:19,680

对，这倒是真的。这是它的好处

798

00:48:19,680 --> 00:48:24,950

它还具有几乎无限的扩展能力

799

00:48:24,950 --> 00:48:27,000

你可以把它放在任何你需要它的地方

800

00:48:27,000 --> 00:48:29,180

你有线速率端口

801

00:48:29,180 --> 00:48:31,390

你还想要什么？

802

00:48:31,390 --> 00:48:35,030

呃，我不能一伙计，我想要，要不你往我在威斯康星的实验室

803

00:48:35,030 --> 00:48:37,200

发送5个这个产品吧。这样就完美了

804

00:48:37,200 --> 00:48:39,400

我们可以试试

805

00:48:39,400 --> 00:48:41,750

我不能给你保证

806

00:48:41,750 --> 00:48:43,830

呃。谢谢你来到思科技术达人秀

807

00:48:43,830 --> 00:48:45,830

非常感谢，Jimmy，邀请我来

808

00:48:47,580 --> 00:48:49,700

Andrew，你能来参加节目我很高兴

809

00:48:49,700 --> 00:48:52,220

谢谢，欢迎>>我很高兴能来这里

810

00:48:52,220 --> 00:48:55,750

你的背景是我为之兴奋的地方，因为你真的

811

00:48:55,750 --> 00:48:59,360

你为正在收看我们节目的许多客户们设身处地地想

812

00:48:59,360 --> 00:49:01,710

就我所知，你在和思科IT

813

00:49:01,710 --> 00:49:05,460

做存储和存储交换机、努力解决早期问题

814

00:49:05,460 --> 00:49:08,130

存储空间的交换有一个传统

815

00:49:08,130 --> 00:49:12,530

也就是我们今天要说的

816

00:49:12,530 --> 00:49:15,410

但是，我想谈谈你的经历以及

817

00:49:15,410 --> 00:49:18,910

DCNM (数据中心网络管理器) 目前的状况，行吗？

818

00:49:18,910 --> 00:49:21,800

好>>屏幕上已经出现了

819

00:49:21,800 --> 00:49:25,840

很显然，这是一个非常可视非常综合的一个东西，对吧？

820

00:49:25,840 --> 00:49:27,840

支持所有事情吗？

821

00:49:27,840 --> 00:49:29,840

显然支持今天我们要谈的事情

822

00:49:29,840 --> 00:49:31,350

要不然我们也不会邀你过来。这合逻辑吗，还是我们没有说过？

823

00:49:31,350 --> 00:49:35,320

很有逻辑，Spock，很好

824

00:49:35,320 --> 00:49:38,550

告诉我，说到管理，现在客户的兴趣是什么？

825

00:49:38,550 --> 00:49:42,190

我觉得客户们各种各样的管理工具，他们都要对其进行管理

826

00:49:42,190 --> 00:49:44,920

这真的很难，因为

827

00:49:44,920 --> 00:49:49,740

如果你要做一个故障根源分析的话，你必须

828

00:49:49,740 --> 00:49:54,350

在一个正常的大环境中进行，很可能要检查10个工具，才能找到问题根源

829

00:49:54,350 --> 00:49:57,150

视问题而定

830

00:49:57,150 --> 00:50:01,280

我们对DCNM做的比较特殊的是，我们把LAN

831

00:50:01,280 --> 00:50:04,800

SAN结合在了一起，还有它们的磁盘工具

832

00:50:04,800 --> 00:50:09,680

事实上，看一下屏幕，你就会看到这些LAN和SAN岛状区域

833

00:50:09,680 --> 00:50:12,950

现在我们的SAN岛状区域也是多协议的

834

00:50:12,950 --> 00:50:16,090

也就是说，无论你在使用

835

00:50:16,090 --> 00:50:18,730

FICON或者iSC SI或者在以太网使用光纤通道

836

00:50:18,730 --> 00:50:21,800

当然，用我们有的光纤通道，都没有关系

837

00:50:21,800 --> 00:50:24,770

我们可以以一个单独路径，对性能

838

00:50:24,770 --> 00:50:27,270

拓扑可视化等等进行监控

839

00:50:27,270 --> 00:50:31,150

全部从这儿开始>>所以那个多协议环境都不能不释放该工具的能力

840

00:50:31,150 --> 00:50:34,970

给你一个—我不知该怎么说才好

841

00:50:34,970 --> 00:50:37,470

一块玻璃？>>你可以说它是一块玻璃

842

00:50:37,470 --> 00:50:40,530

我们经常说。这也是我刚刚为什么那么说>>我们经常说，是因为它是一个真正的营销术语

843

00:50:40,530 --> 00:50:45,070

但是我感觉我们确实给你们提供了一些单个的玻璃

844

00:50:45,070 --> 00:50:49,540

尤其是因为我们与VMware和Hyper-V的结合

845

00:50:49,540 --> 00:50:52,710

我们允许虚拟世界中的这种可视性

846

00:50:52,710 --> 00:50:55,060

然后扩展到存储区域世界

847

00:50:55,060 --> 00:50:59,270

这整个想法，是使你在监控路径时

848

00:50:59,270 --> 00:51:03,920

你不仅仅想要在交换机之间进行监控，还想再连接交换机的主机上进行
监控

849

00:51:03,920 --> 00:51:06,840

>>展示给我看看—客户怎么与它互动？

850

00:51:06,840 --> 00:51:10,650

这有为仿真设计的普通工作流吗？

851

00:51:10,650 --> 00:51:14,610

恩，本质上来讲，当客户登陆时，我们呈现给客户的是

852

00:51:14,610 --> 00:51:18,520

日志界面；日志界面是用来查看

853

00:51:18,520 --> 00:51:22,030

过去24个小时内发生的事情>>有道理

854

00:51:22,030 --> 00:51:25,210

[Andrew]它真的很重要，因为如果有事情发生的话

855

00:51:25,210 --> 00:51:28,560

你会想要对此进行调查，看看是什么问题

856

00:51:28,560 --> 00:51:30,560

马上查看你的优先权>>恩

857

00:51:30,560 --> 00:51:33,500

因此我们是在门户组件周围组织。这不是重要的一个

858

00:51:33,500 --> 00:51:36,500

但是这些重要的门户组件是围绕健康—发生了什么

859

00:51:36,500 --> 00:51:39,340

所以屏幕上的这6个组件是我们要看的吗？>>正是

860

00:51:39,340 --> 00:51:44,000

你可以定制—最小化，如果你—>>有些事情可能是重要的

861

00:51:44,000 --> 00:51:48,010

是的，你也可以重新改组、定义范围

862

00:51:48,010 --> 00:51:50,850

但是这里的范围，我们设定的是整个数据中心

863

00:51:50,850 --> 00:51:52,960

也就是LAN和SAN

864

00:51:52,960 --> 00:51:57,070

现在如果说：“好，那么，我要存储

865

00:51:57,070 --> 00:52:00,260

我想知道这个交换结构”

866

00:52:00,260 --> 00:52:03,470

那么我就点击这个交换结构，屏幕上的东西都变了

867

00:52:03,470 --> 00:52:06,500

在这个交换结构背景下

868

00:52:06,500 --> 00:52:11,600

所以，这里，我们就有点忘了我们自己在这里，在屏幕上

869

00:52:11,600 --> 00:52:16,520

但是，本质上来讲，这是你被提示的地方

870

00:52:16,520 --> 00:52:18,620

你会监管你的交换结构

871

00:52:18,620 --> 00:52:21,130

还记得这些最大流量生成者吗？它们还在这里

872

00:52:21,130 --> 00:52:27,130

但是这些最大流量生成者是在你选择的范围环境下>>SAN岛状区域

873

00:52:27,130 --> 00:52:29,350

即交换结构。我们选了这个交换结构

874

00:52:29,350 --> 00:52:32,120

这是过去24小时你的流量分布

875

00:52:32,120 --> 00:52:35,450

从本质上说，该界面上的所有东西都是

876

00:52:35,450 --> 00:52:37,450

你点击就可以进去了

877

00:52:37,450 --> 00:52:40,990

这是可用容量比例的反映吗？

878

00:52:40,990 --> 00:52:43,280

正是>>那，现在我们用的比例很少？

879

00:52:43,280 --> 00:52:45,700

呃，蓝色表示我们几乎没有用

880

00:52:45,700 --> 00:52:47,890

这是实验室环境，我一点也不感到奇怪

881

00:52:47,890 --> 00:52:50,460

但是如果在思科IT，猜猜会怎么样

882

00:52:50,460 --> 00:52:54,280

这个画面会红很多很多>>用的多一点

883

00:52:54,280 --> 00:52:56,730

用的多太多了。>>当然，就是这么设计的。>>是的

884

00:52:56,730 --> 00:53:00,190

这是一个大环境，所以，很显然，它会是一个大拓扑图

885

00:53:00,190 --> 00:53:03,990

这个拓扑图是热点图

886

00:53:03,990 --> 00:53:07,570

它会告诉你过去24小时内什么达到了运行高峰

887

00:53:07,570 --> 00:53:10,470

你就可以在交换机上查看你的CPU使用情况

888

00:53:10,470 --> 00:53:12,850

你还可以查看你的内存使用情况

889

00:53:12,850 --> 00:53:18,370

你可以查看特定的ISL，然后可以用图表查看

890

00:53:18,370 --> 00:53:22,490

我点击这个>>什么东西都可以点击

891

00:53:22,490 --> 00:53:25,490

[Andrew]正是[Robb]它是基于知觉的，就你怎么深入

892

00:53:25,490 --> 00:53:27,780

这个引起了我的注意。我想深入看一下

893

00:53:27,780 --> 00:53:31,310

我能找到故障根源>>正是。>>与筛选非智能

894

00:53:31,310 --> 00:53:36,520

日志不同。>>它只是将纯编码器与GUI工具分开

895

00:53:36,520 --> 00:53:41,340

你还需要做许多解析工作，但是我们已经

896

00:53:41,340 --> 00:53:44,400

为您将这些信息预先打包，并将其以

897

00:53:44,400 --> 00:53:46,400

人类能理解的方式可视化

898

00:53:46,400 --> 00:53:48,610

能不能说这有相当一部分关联性呢？

899

00:53:48,610 --> 00:53:52,750

它是抑制警报，这样一来你就可以解决真正重要的问题？

900

00:53:52,750 --> 00:53:55,180

这个想法好>>可以说，不在方程中迷失？

901

00:53:55,180 --> 00:53:59,300

我们的确有处理容量的事件，我们真正处理事件是在

902

00:53:59,300 --> 00:54:01,620

制定一套规则

903

00:54:01,620 --> 00:54:04,510

因此，比如你进行事件转发

904

00:54:04,510 --> 00:54:11,880

你就有规则提示，之后你可以再这个规则内进行定义

905

00:54:11,880 --> 00:54:18,810

如果你想做系统日志，如果你对系统日志不感兴趣

906

00:54:18,810 --> 00:54:24,060

而对系统日志的描述感兴趣，你可以在这里将它输入进去，猜猜看会发生什么？

907

00:54:24,060 --> 00:54:28,690

它就会被转发到你的邮件，鉴于在你定义的描述基础上的

908

00:54:28,690 --> 00:54:31,170

被困第三方管理应用程序

909

00:54:31,170 --> 00:54:34,930

我有点懂了，告诉我这对不对，就是即使你

910

00:54:34,930 --> 00:54:37,890

更侧重CLI，或者你觉得这背后发生的事并无不妥

911

00:54:37,890 --> 00:54:41,090

因为人们害怕GUI，是它

912

00:54:41,090 --> 00:54:44,200

可以隐藏太多的复杂性，不被你发现

913

00:54:44,200 --> 00:54:48,420

但是，看起来，如果对数据系统日志生成之类的事情熟悉的话

914

00:54:48,420 --> 00:54:52,410

你可以运用这知识，加之你从

915

00:54:52,410 --> 00:54:58,320

GUI接口获得的智能，你就可以得到为你的智能水平定制的东西

916

00:54:58,320 --> 00:55:00,810

这样你就可以按你喜欢的方式打游戏了

917

00:55:00,810 --> 00:55:03,720

我觉得你讲得很到位

918

00:55:03,720 --> 00:55:06,280

恩，做得好

919

00:55:06,280 --> 00:55:13,210

对于这些CLT操作员来说，我们有CLI界面。实际上，就在这儿

920

00:55:13,210 --> 00:55:17,060

点击一下，你就可以在交换机上执行命令了

921

00:55:17,060 --> 00:55:20,560

可以通过DCNM看到ECHO命令

922

00:55:20,560 --> 00:55:22,880

所以你不需要使用你的终端本身

923

00:55:22,880 --> 00:55:25,970

你可以打开多交换机，在所有的交换机上输入一个命令

924

00:55:25,970 --> 00:55:29,620

技术支持不足的例子

925

00:55:29,620 --> 00:55:35,510

将它导入临时文件，然后一天呐，它可以是FTP到DCNM

926

00:55:35,510 --> 00:55:38,770

我们只剩一分钟。告诉我，我们和ESX有一致吗？

927

00:55:38,770 --> 00:55:42,140

数据来源等等—我们摆脱的是怎样一个深度？

928

00:55:42,140 --> 00:55:45,600

很重要的是我们与VMware的结合

929

00:55:45,600 --> 00:55:50,170

我们在将可见性引入到虚拟世界方面

930

00:55:50,170 --> 00:55:52,870

做出了非凡的成就

931

00:55:52,870 --> 00:55:56,490

接下来我给大家展示一个例子

932

00:55:56,490 --> 00:56:00,000

我点击一下ESX服务器

933

00:56:00,000 --> 00:56:04,370

看到一个拓扑视图—所有VM到物理服务器的图

934

00:56:04,370 --> 00:56:06,790

这边有UCS

935

00:56:06,790 --> 00:56:10,650

还有MDS交换机，9216s

936

00:56:10,650 --> 00:56:14,790

想法就是，如果你想知道

937

00:56:14,790 --> 00:56:16,970

VM或者ESX服务器的性能，怎么办？

938

00:56:16,970 --> 00:56:21,290

呃，你就只需点击这其中的图标。你就可以获得延迟信息了

939

00:56:21,290 --> 00:56:24,250

一直到转轴，所以这很伟大

940

00:56:24,250 --> 00:56:30,140

如果你要查看事件，在特定的ESX服务器路径环境下

941

00:56:30,140 --> 00:56:32,990

但是VM呢？VM也在这里

942

00:56:32,990 --> 00:56:35,090

我们要做的就是点击VM

943

00:56:35,090 --> 00:56:37,340

[Robb] 但是，看！你把它带进了一个工具，这样我们就不用

944

00:56:37,340 --> 00:56:40,720

点击或处理在一个完全不同的界面或插件

945

00:56:40,720 --> 00:56:44,130

或者做一些—我们心里想做界面连接

946

00:56:44,130 --> 00:56:46,690

[Andrew]正是>>但是它为我们提了出来，显然

947

00:56:46,690 --> 00:56:51,360

即使是在物理图内，它们可以点击，遵循

948

00:56:51,360 --> 00:56:54,790

同样的原理，你有很多种方法深入到根数据

949

00:56:54,790 --> 00:56:58,280

显示出你要找的东西，对吧？>>对

950

00:56:58,280 --> 00:57:01,320

呃，Andrew，我喜欢这个。数据中心网络管理器

951

00:57:01,320 --> 00:57:03,540

它很有意义

952

00:57:03,540 --> 00:57:05,690

我喜欢用它们的作用为它们命名的这种方式>> Prime , prime

953

00:57:05,690 --> 00:57:09,200

[Robb] Prime , 对 , 很对。它是Prime家族的一员。非常感谢 , 我们稍后再和您谈

954

00:57:11,000 --> 00:57:13,680

很好奇你是怎么看的。我们经历过

955

00:57:13,680 --> 00:57:15,680

我们经历过这些产品 , 经历过这些软件

956

00:57:15,680 --> 00:57:19,650

我们以管理收尾。多协议。存储交换机

957

00:57:19,650 --> 00:57:23,120

思科在这方面 , 活力十足?

958

00:57:23,120 --> 00:57:25,970

马力十足，伙计，绝对的

959

00:57:25,970 --> 00:57:30,680

恩，我们看到的这些产品太让人钦佩了

960

00:57:30,680 --> 00:57:35,120

你知道吗，以多业务形式设计这些产品

961

00:57:35,120 --> 00:57:39,630

我们—这些出击的产品更加全面的多了

962

00:57:39,630 --> 00:57:43,220

在节目开始的时候，你提了个问题，我想在这里说一下

963

00:57:43,220 --> 00:57:48,290

你说“好，也许MDS不是最有吸引力的产品”

964

00:57:48,290 --> 00:57:50,290

我们还没有仔细谈过这个

965

00:57:50,290 --> 00:57:52,910

你是问的Nitin，在伦敦Cisco Live，有点为难他的意思

966

00:57:52,910 --> 00:57:56,370

然后在这儿，我们详细地介绍了

967

00:57:56,370 --> 00:58:00,670

你在节目开场说你真正要改变的是什么

968

00:58:00,670 --> 00:58:03,050

并不是所有的东西都受到破坏

969

00:58:03,050 --> 00:58:05,470

你现在怎么回答这个问题呢？

970

00:58:05,470 --> 00:58:09,300

呃，这就像是一个古老的工程设计原理，对吧？

971

00:58:09,300 --> 00:58:11,650

“ 如果没有坏，就不要修理”

972

00:58:11,650 --> 00:58:15,720

老实说，MDS并没有多少需要修的地方

973

00:58:15,720 --> 00:58:23,070

它确实有一个不可思议的可靠的结构，开始还做误差修正

974

00:58:23,070 --> 00:58:25,860

我们更改帧的时候它随时在进行误差修正

975

00:58:25,860 --> 00:58:28,610

它在线速率上运行所有事情

976

00:58:28,610 --> 00:58:31,190

它提供了非常好的扩展灵活性

977

00:58:31,190 --> 00:58:34,850

还可以按你的需求处理数据

978

00:58:34,850 --> 00:58:38,430

我们做的唯一一件事就是我们仍然用的是同一种结构

979

00:58:38,430 --> 00:58:42,280

只是改进了技术

980

00:58:42,280 --> 00:58:45,020

我们使它运行的更快

981

00:58:45,020 --> 00:58:47,170

难道不是真的吗？你不能止步不前，对吧？

982

00:58:47,170 --> 00:58:52,580

至少要进行创新，因为你不能原地踏步

983

00:58:52,580 --> 00:58:54,980

如果你原地踏步，那你就是在退步，对吧？

984

00:58:54,980 --> 00:58:58,220

但是你必须实现你的承诺，真正有原理，制作设计

985

00:58:58,220 --> 00:59:04,920

并不只关注与改变它—

986

00:59:04,920 --> 00:59:08,980

你知道，你10多年前设计的东西

987

00:59:08,980 --> 00:59:12,480

在今天仍是最好的解决方案之一，这是事实

988

00:59:12,480 --> 00:59:15,520

许多构建这些的其他人，会说，好吧，我们优化这个

989

00:59:15,520 --> 00:59:18,610

我们建这个，造那个。他们会改变这些结构。那你们得到的是什么呢？

990

00:59:18,610 --> 00:59:21,550

特性不统一。你们得到的是这些撤销重写

991

00:59:21,550 --> 00:59:25,420

你们得到的是这些产品上奇怪的错误—甚至将各自的设备连接

992

00:59:25,420 --> 00:59:28,400

所以当你有这样的投入精神

993

00:59:28,400 --> 00:59:32,490

你会，比如，这样行得通。这是2+2匹配

994

00:59:32,490 --> 00:59:36,010

它运作起来，每天都会这样运作

995

00:59:36,010 --> 00:59:38,360

你可以将它变得越来越好、越来越好

996

00:59:38,360 --> 00:59:41,930

这是整个节目给我印象最深的地方

997

00:59:41,930 --> 00:59:44,110

是的，它们更快。是的，它们更强

998

00:59:44,110 --> 00:59:47,780

这些东西真的很棒。我喜欢管理—这类东西

999

00:59:47,780 --> 00:59:49,780

这些变来变去的东西

1000

00:59:49,780 --> 00:59:54,010

结构保持没变>>这是让人伤心的地方，是吗？

1001

00:59:54,010 --> 00:59:56,640

因为你使在这些东西得到改进，而又不使由此产生的变化

1002

00:59:56,640 --> 00:59:59,100

把你的核心搞得乱七八糟呢？

1003

00:59:59,100 --> 01:00:01,270

所以即使我们现在仍然是多协议世界

1004

01:00:01,270 --> 01:00:04,990

我们已经有了一个单独的、吸引人的、运行很快的操作系统

1005

01:00:04,990 --> 01:00:07,590

仍然在数据中心运行

1006

01:00:07,590 --> 01:00:10,800

我喜欢的一件—就像是终点一样

1007

01:00:10,800 --> 01:00:14,420

你一直在同客户们见面—一直这样—在此之前

1008

01:00:14,420 --> 01:00:17,260

你出差了几个星期—在我们录制节目之前

1009

01:00:17,260 --> 01:00:19,880

询问客户们在做什么，

1010

01:00:19,880 --> 01:00:23,440

对数据中心和存储合并实施升级

1011

01:00:23,440 --> 01:00:28,250

我很好奇。竞争形势是怎么发生改变的，在过去

1012

01:00:28,250 --> 01:00:32,840

不说从2002年到现在，就说过去5到8年或10年的时间里

1013

01:00:32,840 --> 01:00:36,890

使事情转变到—我的意思是说思科肯定不是市场上唯一的竞争者

1014

01:00:36,890 --> 01:00:39,110

但是我们—我们的方法对吗？

1015

01:00:39,110 --> 01:00:41,110

是的，看起来我们的行动是正确的

1016

01:00:41,110 --> 01:00:45,650

许多客户都在寻求这种解决方案—就是简洁整齐的解决方案

1017

01:00:45,650 --> 01:00:51,380

客户使用MDS解决方案，他们会说：“你知道吗

1018

01:00:51,380 --> 01:00:55,850

如果你真的有更好的东西，拿来给我们看看

1019

01:00:55,850 --> 01:00:58,850

给我们看看你的产品为什么更胜一筹”

1020

01:00:58,850 --> 01:01:03,140

他们可以感觉到他们会引进你的产品，而不是我们的

1021

01:01:03,140 --> 01:01:09,570

即使是10年前发布的产品也比新产品性能好

1022

01:01:09,570 --> 01:01:12,560

是的，有些速度更快，这很好，它们得到了新性能

1023

01:01:12,560 --> 01:01:16,260

但是硬件有点老

1024

01:01:16,260 --> 01:01:19,240

现在你引进了新产品，是你的硬件得到了增长

1025

01:01:19,240 --> 01:01:24,360

新产品再等上10年时间依然好用

1026

01:01:24,360 --> 01:01:26,810

10年对任何一个环境任何一个产品—

1027

01:01:26,810 --> 01:01:29,660

看看你现在有的网络产品，你可以说

1028

01:01:29,660 --> 01:01:32,400

可以使用10年，真的很好。这是了不起一句话

1029

01:01:32,400 --> 01:01:34,780

存储开发人员很难改变他们目前在做的工作吗？

1030

01:01:34,780 --> 01:01:38,150

是的。比说说要难，那网络的其他连接部分呢？

1031

01:01:38,150 --> 01:01:40,620

这是我的经验，恩>>有道理，对吧？>>是的

1032

01:01:40,620 --> 01:01:44,870

因为他们面对的是所有的鸡蛋—金蛋

1033

01:01:44,870 --> 01:01:48,090

太对了，因为一旦你安装了，一旦你

1034

01:01:48,090 --> 01:01:51,830

安装了一个运行良好的解决方案，为什么要招惹它呢，伙计？

1035

01:01:51,830 --> 01:01:54,580

所以它肯定受人瞩目>>是的。肯定会非常非常

1036

01:01:54,580 --> 01:01:57,940

但是我向你保证，如果它不工作的话，我向你保证，所有的存储人员

1037

01:01:57,940 --> 01:02:00,370

他们会立马把你赶去的

1038

01:02:00,370 --> 01:02:04,750

他们会很快摆脱掉你，因为他们确定

1039

01:02:04,750 --> 01:02:08,480

他们有100%的传输可靠性

1040

01:02:08,480 --> 01:02:12,130

如果你不说的话—如果你一次也不做，即使他们用了你的产品10年

1041

01:02:12,130 --> 01:02:16,350

你也要消失了>>没有多少误差的空间

1042

01:02:16,350 --> 01:02:18,960

不，你没有>>呃，有道理>>这就是全部内容了

1043

01:02:18,960 --> 01:02:21,370

节目很优秀。干得好！谢谢你>>恩，真的很棒，你做得很好

1044

01:02:21,370 --> 01:02:23,470

感谢大家的收看

1045

01:02:23,470 --> 01:02:25,470

向所有的嘉宾表示深深的感谢

1046

01:02:25,470 --> 01:02:29,540

工程师们花费了大量的时间运送装置、装配设置

1047

01:02:29,540 --> 01:02:33,790

做足了准备工作，以确保呈现在镜头前的东西又闪亮又运行良好

1048

01:02:33,790 --> 01:02:36,410

所以感谢您的收看。您可以同您的朋友分享我们的节目

1049

01:02:36,410 --> 01:02:38,190

记得同我们保持联系哦。欢迎对我们的工作做及时反馈

1050

01:02:38,190 --> 01:02:40,190

方法很多

1051

01:02:40,190 --> 01:02:43,490

当然，你可以在新浪微博 @思科技术达人秀

1052

01:02:43,490 --> 01:02:45,560

你也可以在思科中文官网观看我们的视频

1053

01:02:45,560 --> 01:02:47,810

简单的方法是登陆我们的网址www.cisco.com.cn

1054

01:02:47,810 --> 01:02:50,990

页面底部有入口可以进入思科技术达人秀的页面

1055

01:02:50,990 --> 01:02:53,520

你可以查找我们的新浪微博名称—很可能已在屏幕下方显示了

1056

01:02:53,520 --> 01:02:57,820

但是你也可以在@techwisetv 再次观看我们的节目或者

1057

01:02:57,820 --> 01:02:59,930

在微博上与我们互动

1058

01:02:59,930 --> 01:03:03,410

我们会持续更新并提供节目或图片幕后花絮之类的东西

1059

01:03:03,410 --> 01:03:06,130

Francis已经着手找照片了

1060

01:03:06,130 --> 01:03:08,410

这真的很有意思，因为你们一直在为我们提出问题

1061

01:03:08,410 --> 01:03:10,410

要继续下去啊

1062

01:03:10,410 --> 01:03:12,670

这是我们的Facebook网站，快捷方式是techwise

1063

01:03:12,670 --> 01:03:15,150

因此Facebook.com/techwise 将会把你链接到我们的网站



1064

01:03:15,150 --> 01:03:17,150

下次再见

1065

01:03:17,150 --> 01:03:19,150

非常感谢你抽出时间参加我们的节目，下次再见

1066

01:03:19,150 --> 01:03:21,150

保重

[© 2013 思科和/或其附属公司版权所有。保留所有权利。]