

# 凌华科技 aTCA-9300 处理器刀片搭配 Intel® Media SDK,大幅提升视频转码性能

凌华科技技术白皮书

平台集成与验证部

嵌入式计算产品事业处

凌华科技

张奇智 资深系统集成工程师

李 想 系统集成工程师

林宗平 PIV 部门经理

2013 年 3 月 29 日



# 目录

<b>前言</b> .....	<b>3</b>
凌华科技 aTCA-9300 .....	4
Intel® Media SDK .....	5
<b>测试方法</b> .....	<b>6</b>
<b>测试结果</b> .....	<b>9</b>
测试 1: 1080P 高清转码测试 .....	9
测试 2: 1080p 转 480p 分辨率下降转码测试 .....	9
测试 3: D1 转 CIF 分辨率下降转码测试 .....	10
<b>分析</b> .....	<b>10</b>
<b>结论</b> .....	<b>12</b>
<b>附录</b> .....	<b>12</b>
<b>References 参考资料</b> .....	<b>14</b>

## 前言

近年来，随着市场和技术的发展，越来越多的集群网络架构开始转向基于模块化计算平台的架构，用以支持和提供多样化的网络单元和丰富的功能，如应用处理、控制处理、包处理、信号处理等。除了节约成本和缩短产品上市时间之外，在机架式系统和不同尺寸的网络设备上，此架构还可以提供模块化架构的灵活性以及按需而定的系统组件独立升级能力。

COTS 产品在许多领域都非常普遍，如电信、网络和国防等行业。相比定制化解决方案而言，COTS 产品能够提供标准的功能、缩短产品上市时间，同时还可以降低成本。为了适应这一趋势的发展，凌华科技发布了 ATCA ARIP (Application-Ready Intelligent Platform) 产品，以满足不同用户的各种需求。

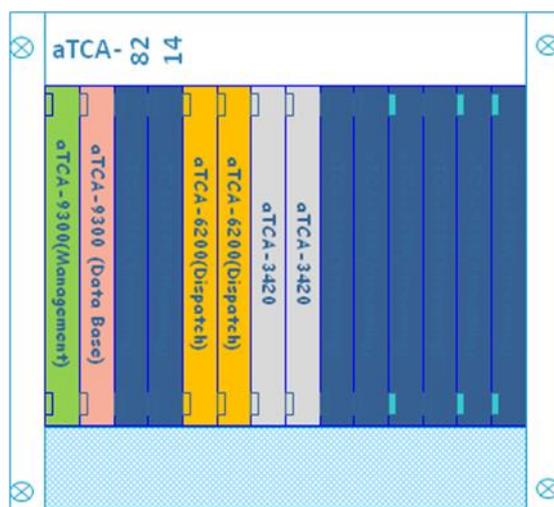


图 1：凌华科技视讯流媒体 ARIP 平台

凌华科技视讯流媒体 ARIP 平台可以根据视频流的负载均衡策略（图 1），将视频流数据转移至服务刀片，从而支持更高密度的 3D 视讯流媒体服务。通过整合 Intel® Media SDK，凌华科技的 aTCA-9300 刀片可以提供高性能的视频转码，并且不增加 CPU 资源的负担。另外，aTCA-9300 的尾板提供 2 个 mini-SAS 接口，可以为视频内容提供足够的存储空间，2 个 10Gbps 以太网接口可以通过 Internet 将视频内容分发至最终用户。每个刀片最多支持 3 个存储设备，通过启用 RAID 功能和数据库功能，可以用来存储配置数据，软件图像以及最终用户的数据。

通过对运行在 CPU 和 GPU 上的视频转码任务进行比较，我们将验证如何利用 Intel® Media SDK 优化凌华科技 aTCA-9300 处理器刀片，从而分担 CPU 的工作负载，并利用 GPU 让转码处理速度更快、更有效。

## 凌华科技 aTCA-9300

凌华科技 aTCA-9300 是一款高集成度的 AdvancedTCA 处理器刀片，支持灵活的处理器选择(支持 Intel® Xeon® E3-1275V2 和 Core™ i7-3750/i5-3550S/i3-3220 处理器)，Intel® 超线程技术和 4 个 DDR3 UDIMM 内存插槽 (总计可提供 32GB 内存容量)。网络 I/O 接口包括 2 个兼容 PICMG 3.1 option 1/9 Fabric 标准的 10G 以太网端口 (XAUI, 10GBase-KX4)，位于前面板的 6 个千兆以太网 10/100/1000BASE-T 端口，2 个 AdvancedTCA Base 接口通道和 2 个后走线千兆以太网接口。

凌华科技 aTCA-9300 处理器刀片专为电信级安全和通信应用而设计，可以在网络基础架构中被用来作为 IMS 服务器，媒体网关，包检测服务器，流量管理服务服务器和 WLAN/小蜂窝接入点控制器。aTCA-9300 详细的功能架构如图 2 所示。

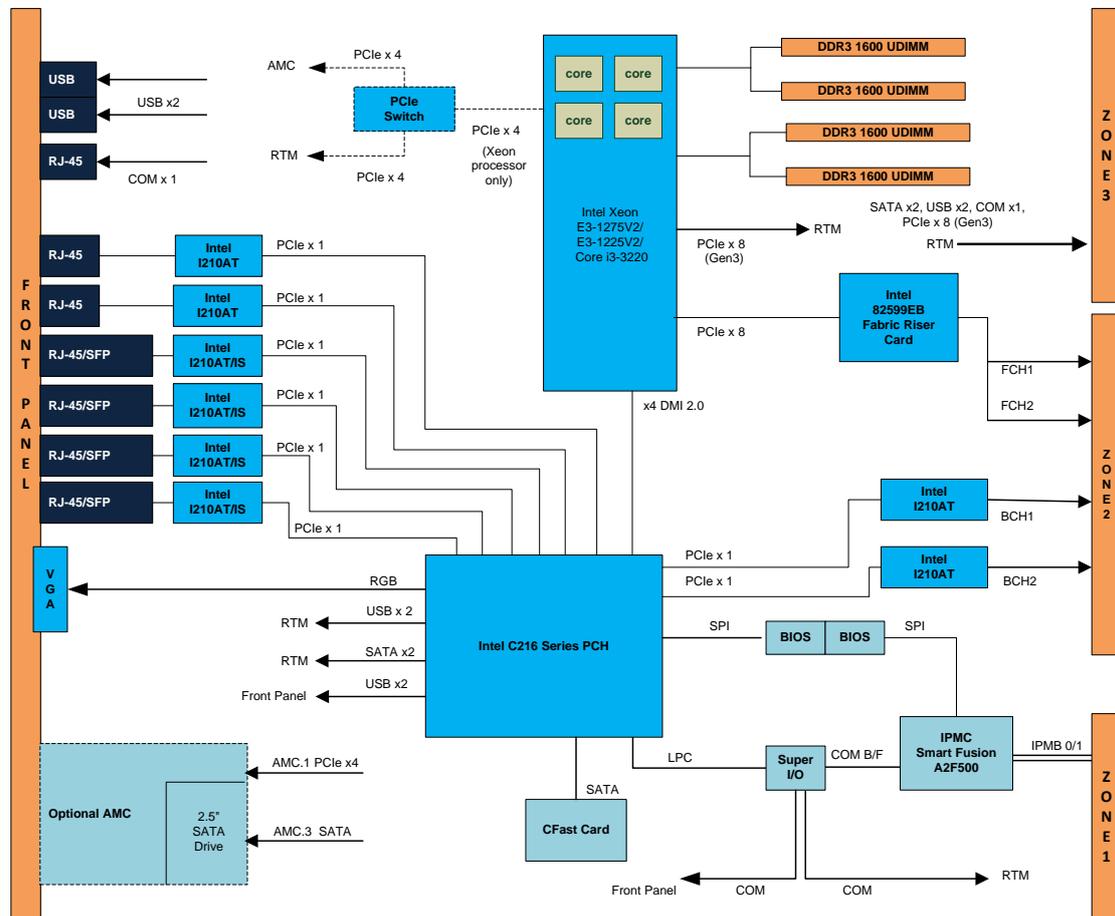


图 2: aTCA-9300 功能示意图

在传统的网络架构中，处理器模块提供应用层和控制层的功能，DSP 模块提供定制化的信号层或媒体处理功能，包处理模块提供数据层的功能，交换模块提供 in-band 和 out-of-band 系统模块之间的路由交换。通过使用 Intel® 媒体软件开发工具 (Intel® Media SDK)，基于 Intel® 架构的处理器模块不仅可以实现

传统的应用和控制进程任务，还可以高效的处理媒体进程任务（本篇技术白皮书将就此功能展开讨论）。

## Intel® Media SDK

Intel® Media SDK 是一个开发库，基于 Intel® 的平台架构，提供业界领先的媒体加速性能，如编码、解码和转码。API 库通过使用软硬件编解码自适应，可以跨平台、跨操作系统使用，开发者只需简单的编码即可适用今天或者未来的芯片组。开发者可以采用先进的可扩展接口，易于使用的编码示例和文档，可以获得产品在上市时间上的竞争优势，同时为应用提供杰出的动力和性能。建立在 Intel® Media SDK 上的应用提供了一个高度一致的，丰富的优质媒体用户体验。Intel® Media SDK 的功能架构如图 3 所示。

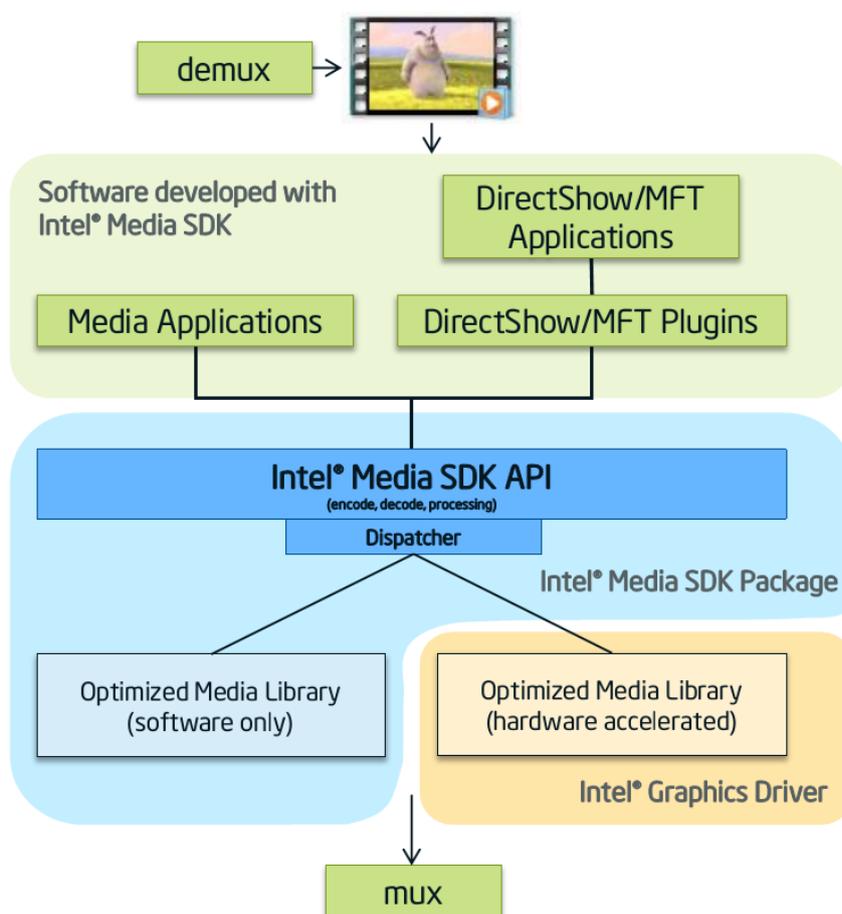


图 3：Media SDK 架构展示了软件和硬件的视频流处理路径

Intel® Media SDK 搭载 Intel® Quick Sync Video 硬件，可以提供最快的性能，既可以应用于一般视频处理程序中，也可以应用于如 Microsoft DirectShow 和 Microsoft MFT ( Media Foundation Transform ) 等插件中，如图 4 所示。

Intel® Media SDK 把与 Intel®图形媒体加速接口交互的复杂任务进行抽象化，并为基本的视频流处理操作提供一个高级的视图。同样的接口，也提供了纯软件实现方式，从而兼容更多的处理器。

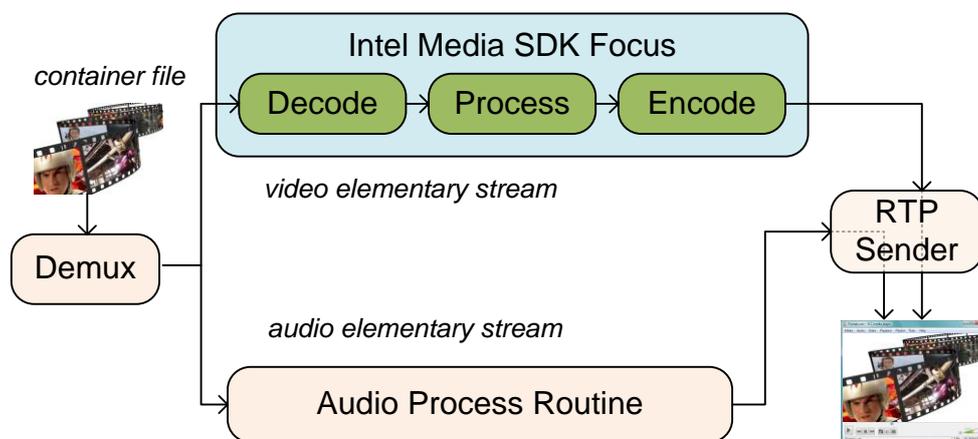


图 4：凌华科技视讯流媒体传输路径图

Intel® Media SDK 针对视频处理，提供了丰富的功能，但是它并不提供针对多媒体流任务的综合性端到端的解决方案。值得注意的例外是音频处理，container 格式操作，以及 RTP 流（如图 4 的亮黄色所示：多路分配器、音频处理程序，以及 RTP 发送器）。但是，凌华科技视讯流媒体 API 为处理这些操作提供了便利，能够协助用户撰写完整功能的流媒体应用，并能处理多种视频格式。

为了理解 Intel® Media SDK 是如何给 HD graphics 加速接口（不是软件接口）提供优势。我们创建了一种测试方法，通过对测试结果进行“apple-to-apple”（一种比较方法，需要两者有一一对应的特点）的比较和分析，得出一些有意义的结论。

## 测试方法

为了评估 Intel® Media SDK 如何增强 aTCA-9300 的媒体处理能力，我们使用了 Intel® Media SDK 的多路转码示例程序，这个程序包含在 Intel® Media SDK 的 Windows 安装中，可以从下面的路径中找到。

```
<install-folder>\samples\sample_multi_transcode.exe
```

Intel® Media SDK 多路转码示例程序演示了如何使用 Intel® Media SDK API 来创建一个控制台应用程序，并且使用这个应用程序进行视频流的转码（解码和编码），即从一个压缩的视频格式转换成其他的格式，并且在重新编码前可选择对解码后的的视频元素进行处理（例如调整大小）。这个应用程序支持多个输入流和输出流，这就意味着它可以同时执行多个转码会话。

这个示例的主要目的是用来证明 CPU/GPU 的平衡能力，并以此为基础在 Intel 硬件加速平台（支持编码）上获得最大的吞吐量。通过一些转码任务的并行运行以及充分挖掘 CPU 和 GPU 的转码性能即可达到这样的验证目的。

可执行文件 *sample\_multi\_transcode.exe* 需要下面的命令行参数切换到正常的运行状态：

<code>-par &lt;par_file&gt;</code>	参数文件是一个具有特殊结构的配置文件。它包含几个命令行，每行对应一个单一的转码会话。
<code>-p &lt;perf_file&gt;</code>	对应于每一个转码会话，将性能统计数据写入到指定的文件中。这样可以统计共用的转码时间。

右边的命令行实例可以用来执行 Intel® Media SDK 转码测试：

```
sample_multi_transcode.exe -par
1.par -p 1.perf
```

下面是两个 *par* parameter 文件内容的实例。

右边命令行中的 *par* 选项适用于单一视频流转码。这个例子是源视频 *input1.mpeg2* (类型: mpeg2) 进行转码，变成目标视频 *output1.h264* (类型: h264, 帧宽: 1920, 帧高: 1920, 帧速率: 30)。由于源视频的相关信息无需写入参数文件，因此转码程序将从视频剪辑中直接提取所需的信息。

```
-<sw|hw> -i::mpeg2 input1.mpeg2
-o::h264 output1.h264 -w 1920 -h
1080 -f 30
```

-<sw|hw>选项表明是选择软件(CPU)还是硬件(GPU,带 Intel® 高清图像媒体加速接口)来执行转码任务。如果“-hw”被选择，Intel® Media SDK 将使用 GPU 上的 Intel® Quick Sync Video 硬件加速转码处理。如果“-sw”被选择或者这个选项没有被定义，那么 CPU 将直接来处理转码。

右边命令行中的 *par* 选项适用于多路视频流转码。这个示例将 5 个源视频文件 *input\*.h264* (类型: h264) 同时转码为 5 个不同的目标视频文件 *output\*.h264* (类型: h264, 帧宽: 848, 帧高: 480, 帧速率: 30)。-<sw|hw>选项具有同样的用处（如同单一视频流转码一样）。

```
-<hw|sw> -i::h264 input1.h264
-o::h264 output1.h264 -w 848 -h
480 -f 30

-<hw|sw> -i::h264 input2.h264
-o::h264 output2.h264 -w 848 -h
480 -f 30

-<hw|sw> -i::h264 input3.h264
-o::h264 output3.h264 -w 848 -h
480 -f 30

-<hw|sw> -i::h264 input4.h264
-o::h264 output4.h264 -w 848 -h
480 -f 30

-<hw|sw> -i::h264 input5.h264
-o::h264 output5.h264 -w 848 -h
480 -f 30
```

当 *sample\_multi\_transcode.exe* 程序完成了转码任务后，它将记录每一个转码过程所需要的时间，并写入到 *perf* 性能日志文件中。通过对此文件的解读，我们能够清楚地了解每个转码流的处理时间以及共同转码时间，这也是完成所有转码流所需花费的时间。

让我们把实例的实时多路转码能力当作测试基准。把一个 10 秒的视频片断的多个复本作为源视频(*input\*.h264*)，可执行文件 *sample\_multi\_transcode.exe* 将所有源视频进行同时转码，变成目标视频(*output\*.h264*)。当转码任务完成后，通过查看 *perf* 文件中的共用时间，我们可以很容易的推断出一个特定的转码任务可以支持多少个实时转码流。如果性能文件中的共用转码时间（最大的消耗时间）大于视频播放的持续时间，那么就没有做到实时处理。相反地，如果共用转码时间小于视频播放的持续时间，我们可以肯定地说实现了实时转码。

如果从 1 个转码流开始并逐渐增加转码流的数量，我们可以测定何时共用转码时间开始变得比视频播放持续时间长。一旦这个时间被确定，我们就可以得出结论，先前的转码流数量就是特定实时转码任务可以支持的最大数量。当作这样的测试时，我们会用到另外一个被称作“Process Explorer”的工具，用来记录大致的 CPU 和 GPU 使用率。下面的伪码说明了当支持实时转码时，如何来获得最大的转码流数量。

```
Preparing the source media;  
Start with one flow in the par parameter file;  
while (TRUE); do  
{  
    Start sample_multi_transcode.exe;  
do {  
    Evaluate the CPU and GPU usage;  
} until the transcoding is completed;  
Retrieve the common transcoding time;  
if (common transcoding time > clip play duration)  
{  
    break;  
}  
    Increase the flows in the parameter file by 1;  
}  
The supported real-time transcoding flows =  
    Number of flows in the last par file - 1;
```

## 测试结果

在这个部分，我们使用了上述的测试方法来测试 3 个不同的转码任务，然后比较各自的转码性能。所使用的基准平台描述如下：

主板	凌华科技 aTCA-9300 处理器刀片
操作系统	Microsoft Windows 7 Professional 64-bit .
CPU	Intel ®Xeon E3-1275V2
内存	Kingstone DDR3 1333MHz 8GB SDRAM
显示	Intel® HD Graphics G4000
Media SDK API	Intel® Media SDK 系统分析器,支持的 API 级别: HW 1.3, SW 1.4

### 测试 1：1080P 高清转码测试

首先,我们会评估一个高清的转码任务。源媒体是一个 1080p MPEG-2 格式的视频剪辑，播放时间为 10 秒，帧速率为 30fps。目标媒体的格式是 H.264，并且具有同样的分辨率。使用硬件方法的测试结果如附件表 1 所示，采用软件方法的测试结果如附件表 2 所示（以红色显示的结果表示转码时间超过了播放时间）。

通过对转码测试结果的比较，很明显，使用硬件的方法 CPU 的使用率非常低（约 9%），而使用软件的方法则因为所有的转码都是由软件编码直接处理的，因此 CPU 使用率极高。

如果考虑实时转码的性能时，使用硬件的方法可以支持 8 个高清的转码流，而软件的方法只能支持 1 个高清转码流，并且当转码流的数量增至 2 个时，则需要长达 23 秒才能完成转码任务，这个处理时间远远大于被测视频剪辑播放持续的时间。

### 测试 2：1080p 转 480p 分辨率下降转码测试

第二步测试我们来评估一下分辨率稍低些（1080p 转 480p）的转码任务。源视频和目标视频都采用 H.264 的格式，源视频为 1080p 视频剪辑，播放持续时间为 10 秒，帧率为 30fps。目标视频的分辨率为 480p。使用硬件方法的测试结

果如附件表 3 所示，使用软件方法的测试结果如附件表 4 所示（红色部分表示转码所需时间超出播放时间）。

通过比较测试结果，很明显使用硬件的方法，CPU 使用率远远再次低于使用软件的方法时 CPU 的使用率。

同样的，对于实时转码性能，硬件的方法可以同时支持 16 个转码流 (H264.1080p 转 H264.480p)，而软件的方法只能支持 3 个转码流。

### 测试 3: D1 转 CIF 分辨率下降转码测试

最后，我们来测试一下低分辨率的转码任务。源视频格式采用 D1 (720x576) H.264 视频剪辑，播放持续时间为 97 秒，帧率为 25fps。目标视频为 CIF(320x288) H.264 格式，与源视频相同的帧率。使用硬件方法的测试结果如附件表 5 所示，附件表 6 为使用软件方法的测试结果（红色部分表示转码所需时间超出播放时间）。由于使用硬件的方法可以支持更多的转码流，因此我们在附件表 5 中只列举有几条有意义的参考数据。

与之前的测试一样，我们可以得出同样的结论：硬件方法的 CPU 使用率远远低于软件方法的 CPU 使用率。H264.D1 转 H264.CIF 的实时转码性能方面，硬件的方法可以同时支持 50 个转码流，而软件的方法最多只能支持 19 个转码流。

## 分析

在本节中，我们对以上的测试结果做进一步的分析。图 5 列举了使用硬件和软件方法时最大实时转码流数量的比较。我们发现，当转码任务越困难时，硬件方法与软件方法的性能差距就越明显。比如 1080p 的转码任务，硬件方法的转码性能是软件方法的 8 倍，而 D1 转 CIF，则只有 3 倍的差距。

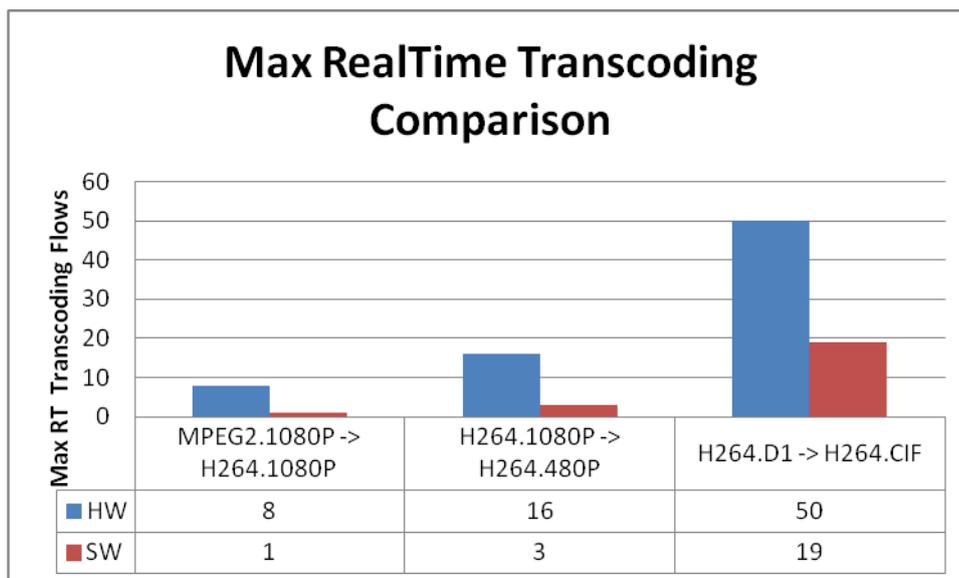


图 5：最大实时转码流数量比较

同样地，在比较实时转码任务中 CPU 的最大使用率时，使用硬件的方法也明显占优。因为使用硬件的方法调用 GPU 硬件来处理转码任务，所以维持了很低的 CPU 使用率。相比之下，软件的方法则完全使用 CPU 的性能来处理转码任务，因才所调用的 CPU 资源就相当的多。（请参看图 6）

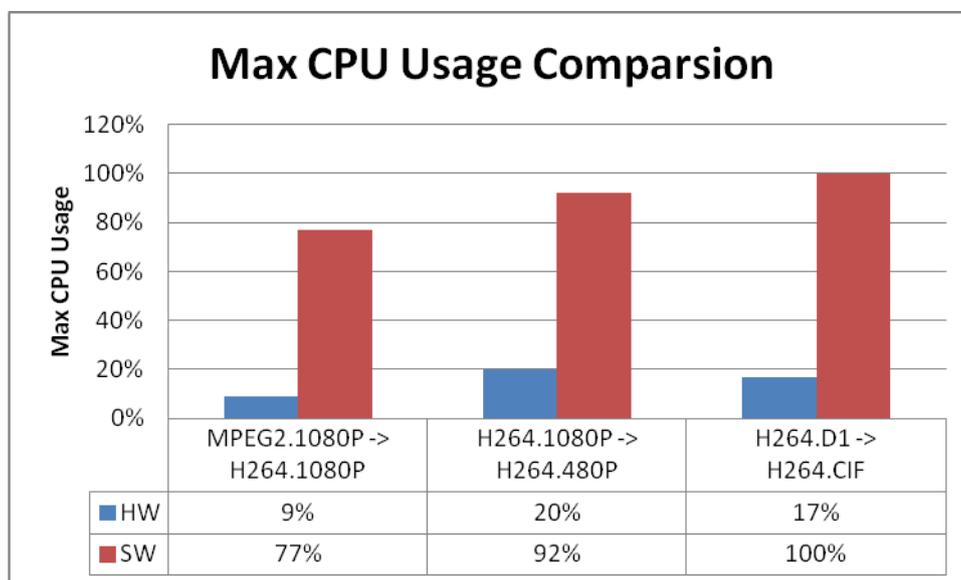


图 6：实时转码时 CPU 的最大使用率

从性能比较的结果来看，很明显 Intel® Media SDK 能够有效利用 GPU 来分担 CPU 处理转码任务的负担，让转码速度变得更快，并且 CPU 资源的占用率更低。如果打算将控制平面和媒体处理整合到一个单一的平台，这种优势便成为一个非常具有吸引力的选择。由于减少了 CPU 编解码的负担，可以让 CPU 集中更多的资源去处理常规的控制任务，并且让这些控制任务的处理变得更加快速和有效。

## 结论

本白皮书通过对比纯软件方法和基于 GPU 和 Media SDK 的硬件方法的转码，展示了凌华科技 aTCA-9300 所具有的精湛媒体处理性能。尤其对于高分比率的转码任务两种方式的转码对比愈加明显，如将 1080p 的 mpeg2 转码至 h264，由 GPU 和 Media SDK 所获得的性能提升是纯 CPU 转码时性能的 8 倍。另一方面，aTCA-9300 在进行硬件转码时还保持了较低的 CPU 利用率 (<10%)。节省下来的 CPU 能力可以用于处理其它应用任务。这使得凌华科技的 aTCA-9300 成为客户寻求以较低成本获得更高吞吐量和性能时的理想平台。

最后，值得一提的是，所有的测试都是基于 Microsoft Windows 平台进行的，Intel®还没有发布 Linux 版本的 Intel® Media SDK。一般来说，大多数媒体客户更喜欢 Linux 环境，所以我们认为 Linux 版本的 Intel® Media SDK 将更容易被一个更大的市场所接受。如果这个在不久的将来变成现实，我们将再来评估 Intel® Media SDK Linux 版本的性能以及其呈现的结果。

## 附录

在上面测试结果章节中描述的各种转码任务使用硬件的方法 (HW) 和软件的方法 (SW) 的测试结果

注意：红色部分表示转码所需时间超出播放时间

表 1: MPEG2.1080p 转 H264.1080p (使用硬件的方法)

转码流	CPU 使用率	GPU 使用率	转码时间
1	7%	55%	2.60s
2	8%	87%	2.79s
3	9%	93%	3.77s
4	9%	94%	5.01s
5	9%	94%	6.32s
6	9%	94%	7.53s
7	9%	94%	8.78s
8	9%	94%	10.14s
9	9%	94%	11.46s
10	9%	94%	12.71s

表 2: MPEG2.1080p 转 H264.1080p (使用软件的方法)

转码流	CPU 使用率	GPU 使用率	转码时间
1	77%	0%	6.79s
2	95%	0%	23.91s

<b>3</b>	<b>99%</b>	<b>0%</b>	<b>54.97s</b>
----------	------------	-----------	---------------

表 3: H264.1080p 转 H264.480p (使用硬件的方法)

转码流	CPU 使用率	GPU 使用率	转码时间
1	8%	50%	0.93s
2	9%	75%	1.27s
3	13%	80%	1.57s
4	16%	83%	2.59s
5	17%	88%	3.20s
6	18%	88%	3.86s
7	18%	88%	4.29s
8	19%	88%	4.61s
9	19%	88%	5.19s
10	19%	89%	5.77s
11	19%	89%	6.26s
12	19%	89%	6.75s
13	20%	89%	7.39s
14	20%	89%	8.37s
15	20%	89%	8.64s
16	20%	89%	9.00s
<b>17</b>	<b>20%</b>	<b>80%</b>	<b>11.20s</b>
<b>18</b>	<b>20%</b>	<b>65%</b>	<b>14.73s</b>

表 4: H264.1080p 转 H264.480p (使用软件的方法)

转码流	CPU 使用率	GPU 使用率	转码时间
1	60%	0%	3.36s
2	83%	0%	5.48s
3	92%	0%	9.00s
<b>4</b>	<b>97%</b>	<b>0%</b>	<b>14.53s</b>

表 5: H264.D1 转 H264.CIF (使用硬件的方法)

转码流	CPU 使用率	GPU 使用率	转码时间
1	8%	45%	5.05s
2	14%	61%	4.82s
5	15%	79%	9.11s
10	16%	79%	18.33s
15	16%	79%	27.50s
20	17%	79%	37.04s
25	17%	79%	46.53s
30	17%	79%	56.14s
35	17%	79%	65.57s
40	17%	79%	74.44s
45	17%	79%	84.71s
50	17%	79%	92.99s
<b>55</b>	<b>17%</b>	<b>79%</b>	<b>101.83s</b>
<b>60</b>	<b>18%</b>	<b>79%</b>	<b>112.37s</b>

表 6: H264.D1 转 H264.CIF (使用软件的方法)

转码流	CPU 使用率	GPU 使用率	转码时间
1	70%	0%	5.49s
2	82%	0%	9.26s
3	90%	0%	13.13s
4	94%	0%	17.27s
5	96%	0%	21.80s
6	98%	0%	25.98s
7	99%	0%	30.07s
8	100%	0%	34.83s
9	100%	0%	38.63s
10	100%	0%	43.99s
11	100%	0%	49.21s
12	100%	0%	56.37s
13	100%	0%	61.18s
14	100%	0%	66.29s
15	100%	0%	70.39s
16	100%	0%	76.20s
17	100%	0%	82.30s
18	100%	0%	88.80s
19	100%	0%	93.29s
20	100%	0%	100.49s

## References 参考资料

1. Intel® Media SDK 2012 Developer's Guide
2. Intel® Media SDK Multi-Transcoding Sample
3. Intel® Media SDK 2012 Product Brief