IIP 中的 ASR 功能的优化①

Optimization of ASR Function in IIP

石 岳　廖建新　王 纯　朱晓民
（北京邮电大学网络与交换技术国家重点实验室　北京 100876）
（东信北邮信息技术有限公司　北京 100083）

摘　要：IIP（Independent Intelligent Peripheral，独立智能外设）是智能网中重要的功能实体之一，它能够完成智能网的特殊资源功能（Special Resource Function，SRF）。ASR（Automatic Speech Recognition，自动语音识别）是 IIP 系统开展语音业务所需的重要的媒体资源功能。本文首先简要介绍了 ASR 功能在 IIP 中的实现以及存在的问题；随后重点从语法加载方式、系统 I/O 和集成开发方式三个方面对 IIP 中 ASR 功能进行了优化。

关键词：IIP　ASR　优化

1　IIP 中的 ASR 功能

IIP（Independent Intelligent Peripheral，独立智能外设[1]）是智能网[2]中重要的功能实体之一，它能够完成智能网的特殊资源功能，为智能业务提供各种专用资源。随着电信增值业务的不断发展和移动用户对业务内容需求的增大，传统的使用固定语音内容提供服务的方式已经不能满足需求。作为 IIP 系统开展语音业务所需的极为重要的媒体资源功能，ASR（Automatic Speech Recognition，自动语音识别）的加入使得用户使用 IIP 提供的 IVR（Interactive Voice Response，交互式语音应答）业务的体验感得到了大幅度的改善，提高了服务质量，并且促使了新的业务形式的出现[3]。

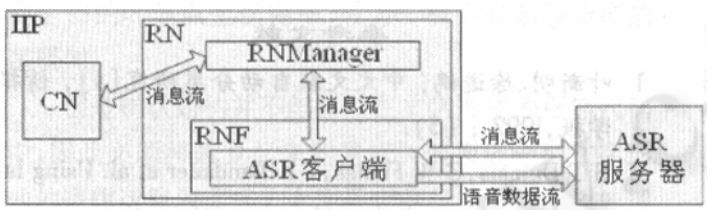

图 1　IIP 中 ASR 功能的实现

图 1 描述了 ASR 功能在 IIP 中的具体实现方式。IIP 由 CN（Control Node，控制结点）和 RN（Resource Node，资源节点）组成。其中 RN 又包括 RNManager 和 RNF（Resource Node Function，资源节点功能）两个子功能实体。RNManager 负责在 CN 和 RNF 之间进行消息路由和转发；RNF 是 RN 的核心模块，基于硬件语音板卡的 API（Application Programming Interface，应用编程接口）实现。它负责对语音卡的资源进行控制和管理，为语音信道提供媒体资源。IIP 中的 ASR 功能是通过在 RNF 中开发 ASR 客户端的方式实现的。ASR 客户端中集成了语音板卡和 ASR 引擎提供的 API，用来完成语音数据的采集传输和与 ASR 服务器的控制消息交互。

ASR 功能的引入使得 IIP 提供智能业务的能力增强，但也带来了以下两个方面的问题：

· IIP 系统性能的下降。ASR 的语法（Grammar）编译加载以及识别运算需要消耗大量的内存空间和

① * 基金项目：国家杰出青年科学基金（No. 60525110）；国家 973 计划项目（No. 2007CB307100，2007CB307103）；新世纪优秀人才支持计划（No. NCET-04-0111）；电子信息产业发展基金项目（基于 3G 的移动业务应用系统）；电子信息产业发展基金重点项目（下一代网络核心业务平台）；电子信息产业发展基金项目（基于内容的综合通信网络计费平台）；国家高技术产业化信息化装备专项项目（支持数据增值业务的移动智能网系统）

CPU 时间。ASR 的语音数据的采集过程是一个多路并发的数据传输和编解码的过程，在服务器 – 客户端的应用模式下，这一过程会占用大量 CPU 时间、总线带宽和网络 I/O 资源。这些因素都会导致 IIP 系统整体性能的下降。

· IIP 系统复杂度的提升，可重用性和可维护性的下降。目前，各个 ASR 引擎提供商都提供“引擎 + API”形式的产品给应用集成商。集成商通过在自己的应用中调用 API 的方式来控制引擎完成 ASR 功能。由于各个厂商引擎产品的在集成方式上存在着较大的差异性，这就造成了引擎与被集成的系统是紧密耦合的。集成开发人员需要针对不同的引擎开发不同的应用程序。在这种开发方式下，应用程序的可重用性和兼容性大大降低，而相应的开发复杂度、出错概率和维护的成本却大幅度提高。

本文以下的内容将针对以上两个方面的问题，从不同角度研究对于 IIP 中 ASR 功能的优化方法[4]。需要说明的是：以下优化方法虽然是在现有 IIP 系统中实施的，但实际上其中的大部分方法以及其优化思想可以推广到更为广泛的 ASR 应用领域中去。

2　语法加载方式的优化

语法在语音识别中是必不可少的。它规定了语音识别的目标范围，对识别的准确率有很大的影响。语法内容一般以 ABNF（Advanced Backus – Naur Formula，扩展巴克斯范式）或 VXML（Voice XML，语音扩展标记语言）文本的形式保存在文件中。文本形式的语法并不能直接被用于 ASR，需要经过编译（文本到二进制）和激活（将语法动态绑定到特定的 ASR 资源中）。整个语法加载的过程包括：编译、激活以及不同形式的语法数据在网络上的传输。设语法加载总时间为 T，则 T 可以用以下表达式描述：（中括号表示可选内容，下同）

$$T = t_a[+t_{tf}][+t_{tu}][+t_{tb}][+t_c]$$

t_a 语法激活时间　　t_{tf} 语法文件网络传输时间

t_{tu} 语法 URI 内容网络传输时间　　t_{tb} 编译后的二进制语法网络传输时间

t_c 语法编译时间　　一般地，$t_c >> t_a > t_{tb} > t_{tf} > t_{tu}$

设语法加载过程中网络数据量为 D，则 D 可以用以下表达式描述：

$$D = 0[+d_f][+d_u][+d_b]$$

d_f 语法文件数据量　　d_u 语法 URI 内容数据量

d_b 编译后的二进制语法数据量　　一般地，$d_b > d_f > d_u$

语法加载方式的优化目的就是在兼顾可维护性和可靠性的同时尽量减小 T 和 D。基于这个目标，IIP 中的 ASR 功能的语法加载方式经过了以下三个阶段的演进：

（1）语法文件保存在客户端。每次使用语法时，客户端将完整的语法文件传递给服务器。语法在服务器端进行编译、加载。

方式 I 实现起来非常简单，但是其缺点也非常明显：语法每次使用前都要被编译一次，这就造成了时间和内存空间的大量消耗，同时也增加了网络上的数据量。语法文件分布在各个客户端主机上，同步比较困难。因此，在方式 I 的基础上可以进行以下优化。

（2）语法文件保存在服务器端，客户端每次以 URI 的方式通知服务器使用的语法，利用服务器的语法缓存来减少语法编译的次数。同时也减少了网络上的数据量。

方式 II 可以大幅度地减小 T，在单一服务器并且语法变化不频繁的情况下是一个很好的解决方案。但是大规模的电信级应用需要避免服务器的单点故障隐患。在多服务器的情况下，各个服务器之间的语法同步是一个不可忽视的问题，为了兼顾可靠性和可维护性，在方式 II 的基础上引入数据库，形成了方式（3）。

（3）将一次性编译后的二进制语法保存在统一的语法数据库中。所有服务器均与此数据库连接，客户端通过 URI 通知服务器使用语法。语法数据库根据服务器的请求，将二进制语法传递给服务器。每当语法发生变化后，数据库中的语法可以异步编译。服务器不须再进行语法编译，直接激活、使用。

通过表 1 中的比较可以看出，从 T 和 D 最小的角度来看，方式 II 是最优的。但是，语法与服务器同在一台主机，大大增加了单点故障的风险。方式 III 中虽然也存在语法数据库主机的单点隐患，但是由于服务器端语法缓存的存在，语音识别在语法数据库主机故障后依然可以继续进行。况且方式 II 和 III 均在第一次使

用后避免了最大的时间消耗 tc，因此在综合考虑效率、可维护性和可靠性的基础上，方式 III 是最优选择。

表 1　不同语法加载方式的时间和网络数据量的比较

	第1次使用		第n次使用(n ≥ 2)		点风险	语法同步
	T	D	T	D		
方式 I	$t_a + t_{ff} + t_c$	d_f	$t_a + t_{ff} + t_c$	d_f	高	困难
方式 II	$t_a + t_{fu} + t_c$	d_u	$t_a + t_{fu}$	d_u	高	容易
方式 III	$t_a + t_{fu} + t_{fb} + t_c$	$d_u + d_b$	$t_a + t_{fu} + t_{fb}$	$d_u + d_b$	低	容易

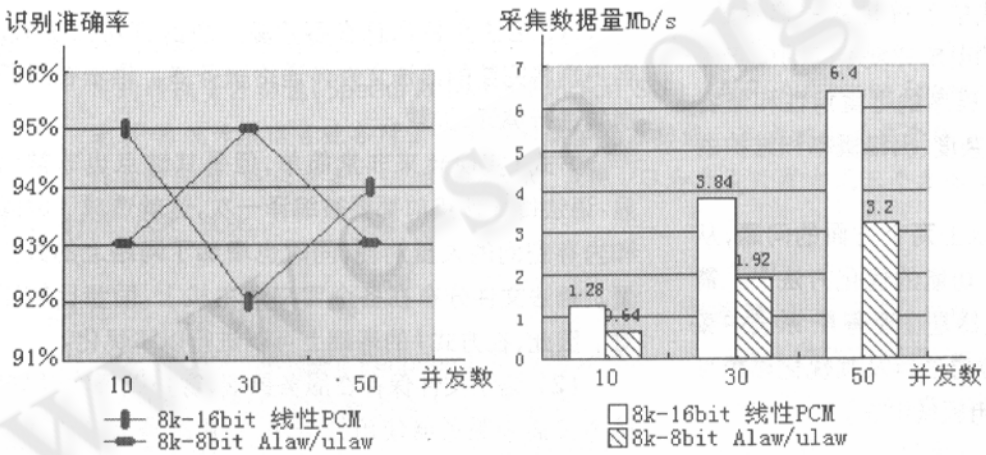

图 2　8k－8bit Alaw/ulaw 和 8k－16bit 线性 PCM 识别准确率和采集数据量比较

HP DL580 主机。实验证明：在其他情况均相同的情况下，8bit 与 16bit 量化数据的识别准确率几乎没有差别，而 8k－8bit Alaw/ulaw 的语音数据流量只有 8k－16bit 线性 PCM 的一半（如图 2 所示）。因此，使用 8k－8bit Alaw/ulaw 作为数据采集的编码格式可以有效地降低数据流量和数据总线的占用率。

需要说明的是差分 PCM（ADPCM）编码算法（如 Dialogic 公司的 vox 格式）可以实现更高的压缩率（如 8k－4bit），但是由于这种算法牺牲了太多的语音特征，失真比较严重，所以一般不能用于语音识别。

3　系统 I/O 方面的优化

要实现 ASR 功能，用户语音输入的采集过程是必不可少的。IIP 是通过 RNF 提供的异步内存录音能力来完成这一过程的。在大规模的电信级应用中，语音采集过程是一个多路并发的内存和网络 I/O 过程。为了降低 I/O 操作对系统整体性能的影响，可以从以下两个方面对语音采集过程进行优化：

3.1　调整语音数据编码格式

用户的语音输入经过语音卡的"抽样－量化－编码"过程[5]后以语音数据的形式被采集到 IIP 系统中。在同样的抽样频率（一般为 8000Hz）下，量化位数越少，编码后的数据量就越小。目前，大多数语音识别引擎都支持 8k－8bit Alaw/ulaw 和 8k－16bit 线性 PCM 两种编码格式。我们设计了一个实验，软硬件环境为：中科信利 TSE 3.4.2 识别引擎，500 条容量的语法和

3.2　调整采集时间片长度

在语音采集过程中，RNF 的异步内存录音功能将用户的语音输入分割成大量的长度相等的语音数据片段，这些数据片段称为时间片（设其长度为 Δt）。时间片越短，语音数据的连续程度越高，识别准确率越高。但与此同时，也带来了频繁的 I/O 操作，导致系统负载过高。时间片越长，单位时间内的 I/O 操作就越少，但是会造成识别准确率下降和响应时间过长导致的系统反映迟钝。因此，采集时间片长度对于 IIP 中 ASR 功能的性能来说是一个非常重要的参数。

我们设计了一个实验，软硬件环境为：Speechworks OSR 3.5.1 识别引擎，1000 条容量的语法和 HP DL380 主机，识别并发数是 1。实验表明：（如图 3 所

示)当 Δt 小于 200ms 时,Δt 继续减小对识别准确率的贡献并不明显,而用于数据采集的 I/O 次数却显著上升。因此,为了在识别准确率和系统负载之间寻求平衡点,Δt =200ms 是一个比较理想的折中数值。

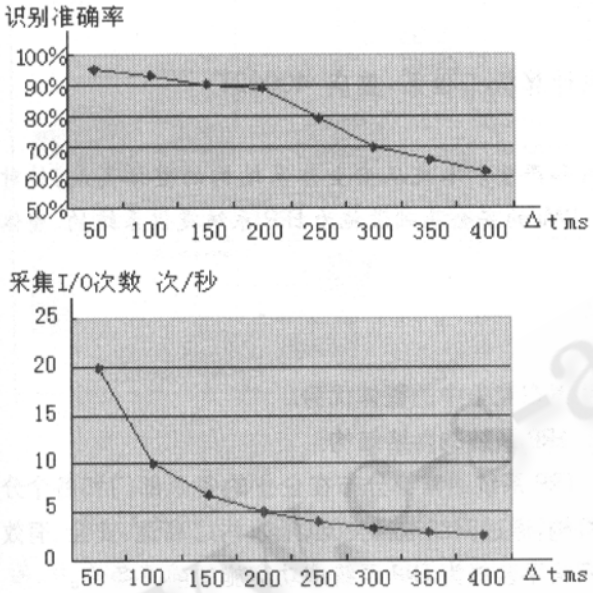

图 3　识别准确度 – Δt　　采集 I/O 次数 – Δt

4　集成开发方式的优化

尽管通过使用语音技术提供商所提供的 API 可以把这些引擎快速地集成到系统中,但是这种集成方式存在明显的缺陷。当需要集成多个不同引擎时,由于不同引擎所提供的 API 之间的较大差异,在每集成一个新的引擎时都需要做新的开发。因此,语音处理行业迫切希望不同的 ASR 引擎能够提供一个标准的接口,MRCP(Media Resource Control Protocol,媒体资源控制协议[6][7])就是在这种条件下产生的。

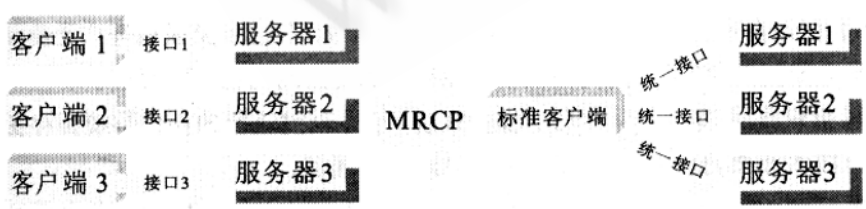

图 4　MRCP 带来的集成开发方式的转变

MRCP 是由 IETF(Internet Engineering Task Force,互联网工程任务组)组织制定的,应用于媒体资源控制方面的协议。尽管 MRCP 并不仅仅是针对语音应用设计的,但是目前其最主要的应用是在语音领域。如图 4 所示,MRCP 的出现使得集成开发人员可以通过一次性开发 MRCP 客户端来与所有支持 MRCP 的 ASR 服务器进行通信,从而使用标准接口来完成 ASR 功能[8]。

表 2　两种实现方式的比较

	集成 API 的实现方式	基于 MRCP 的实现方式
开发工作量	大(针对不同的引擎分别开发)	小(按照统一标准开发)
代码可重用性	低(针对不同的引擎分别开发)	高(一次开发多次使用)
代码维护工作量	大(并行维护多种版本)	小(统一维护标准版本)
开发成本	高(开发、维护成本高,重用性低)	低(开发、维护成本低,重用性高)
实时性	高(专用系统,专门优化)	低(功能实体多,消息控制复杂)
资源利用率	高(专用资源,专门优化)	低(涉及电路 – 分组转换)

从表 2 的比较可以看出,基于 MRCP 的实现方式具有开发工作量、维护工作量小,可重用性高等突出优点。但是,在我们享受 MRCP 带来的方便、高效的同时,还应该注意到 MRCP 复杂的协议体系和电路 – 分组转换所带来的实时性降低和系统资源利用率方面的局限性。

5　结束语

从 IIP 产品在现网实际应用情况来看,成本(包括开发和维护)一直是困绕 ASR 相关业务大规模应用的障碍。通过以上优化,IIP 中 ASR 功能在稳定性,资源利用率、性能表现和可重用性方面有了较大幅度的提升。与此同时,开发的复杂度和维护成本有所下降。相信在各方力量的共同努力下,ASR 将在语音增值业务中充分发挥其潜力,为广大用户提供更为方便、快捷、稳定、高效的服务。

(下转第 40 页)

(上接第35页)

参考文献

1 朱晓民、黄晖、廖建新、沈奇威、郑劲松,移动智能网独立智能外设的设计与实现,北京邮电大学学报,第26卷第4期,2003年12月,pp75-79.

2 廖建新、王晶、郭力等,移动智能网,北京邮电大学出版社,2000年11月.

3 李国翼,ASR与TTS功能在语音增值业务平台中的设计与实现,北京邮电大学硕士研究生学位论文,2006年2月.

4 杨孟辉、廖建新、沈奇威、张奇支,独立智能外设的性能建模与分析,电子信息学报,第28卷第8期,2006年8月,pp1422-1428.

5 樊昌信等编著,通信原理,国防工业出版社,2005年2月.

6 Saravanan Shanmugham, P. Monaco, B. Eberman. IETF RFC 4463, Media Resource Control Protocol (MRCP), April 2006.

7 Saravanan Shanmugham, D. Burnett, Media Resource Control Protocol Version 2 (MRCPv2), draft-ietf-speechsc-mrcpv2-11, September 2006.

8 蒋明哲,独立智能外设中基于MRCP的语音定制业务的设计与实现,北京邮电大学硕士研究生学位论文,2007年2月.