VoiceXML 语音平台中预取方案的研究

王文林　廖建新　朱晓民　王 纯
(北京邮电大学网络与交换技术国家重点实验室　北京　100876)

摘　要： 该文在分析目前主要预取算法优劣的基础上，根据 VoiceXML 语音平台与基于 HTML 的 WWW 之间的区别，认为在 VoiceXML 语音平台中应该预取其引用的语音资源，提出一种自适应的多用户共享的 Markov 预测模型，统一预测所有在线用户下一步所需的资源及其访问概率，有助于提高预测的准确率。最后，该文还提出抢占式优先级模型来调度预取任务，将资源的访问概率映射为优先级。仿真研究表明，与单用户预测算法和循环调度模型比较，该预取算法和调度模型都能很好地减少用户请求的访问延迟，提高响应速度。

关键词：语音平台；VoiceXML；预取；预测；Markov 模型；调度；抢占式优先级
中图分类号：TP393　　**文献标识码：**A　　**文章编号：**1009-5896(2007)11-2574-06

The Study on Prefetch Schema for the VoiceXML-Based Voice Platform

Wang Wen-lin　Liao Jian-xin　Zhu Xiao-min　Wang Chun
(*State Key Laboratory of Networking and Switching Technology*,
Beijing University of Posts and Telecommunications, *Beijing* 100876, *China*)

Abstract: By analyzing the present mainly prefetch algorithms' advantages and disadvantages, ccording to the difference between VoiceXML-based voice platform and HTML-based World Wide Web, it is proposed that the voice resource should be prefetched in VoiceXML-based voice platform. An adaptive multi-user shared predict Markov model is presented to predict the probability of the forthcoming required resource of all the online users, which help to improve the veracity of the prediction. Finally, a preemptive priority model is designed to schedule the prefetch tasks, which mapped the resource access probability to the task priority. The simulation research shows that the predict algorithm and the schedule model can reduce delay of a user's request and improve response speed better than that of single-user predict Markov model and Round-Robin (RR) schedule model.

Key words: Voice platform; VoiceXML; Prefetch; Predict; Markov model; Schedule; Preemptive priority

1　引言

VoiceXML(Voice eXtensible Makeup Language)[1, 2]是由VoiceXML论坛制定的一种用于编写支持语音交互作用的网页的可扩展标志性语言。它与HTML不同在于后者用于设计可视网页，强调视图的布局与外观，缺乏对用户与应用之间的交互控制；前者则提供了对用户与应用之间语音对话的完全控制，利用它可以通过电话和语音访问网站上的信息和服务。使用VoiceXML开发业务应用可以完全参照成熟的Web应用技术，将话音业务与数据业务相融合，将业务表现与业务逻辑相分离，更加有利于快速开发语音业务，将开发人员从繁琐的底层编程和资源管理中解放出来。

图 1 所示是一种基于 VoiceXML 的语音平台的体系架构。用户通过电话接入后，语音平台启动一个 VoiceXML 解释器实例访问 Web 服务器以获得 VoiceXML 文档，经解释器解释后由执行平台执行，下载语音资源并与用户进行语音交流，然后根据交互的结果再次访问服务器取得下一个文档。显然，从用户拨打电话到听到第一个语音的时间间隔，即用户的等待时间，将是语音平台启动获取文档的时间间隔，从 Web 服务器上获取 VoiceXML 文档及语音资源的时间与解释执行此文档时间之和。在这 3 个时间中，从 Web 服务器上获取文档及语音资源的时间是最长，也是最不可控的。

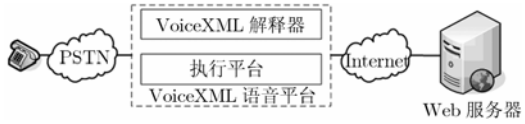

图 1　一种基于 VoiceXML 的语音平台的体系架构

通过语音接入的用户所期待的响应时间远远小于通过 Web 浏览器接入的用户，因此通过语音接入的客户更易因对服务质量(QoS)不满意而流失。所以，VoiceXML 论坛建议：

2006-03-21 收到，2006-05-23 改回
国家杰出青年科学基金(60525110)，新世纪优秀人才支持计划(NCET-04-0111)，高等学校博士学科点专项科研基金(20030013006)，国家高技术产业化信息化装备专项项目和电子信息产业发展基金资助课题

(1)在一个 VoiceXML 文档中包含尽量多的对话，减少与服务器交互次数；(2)在 VoiceXML 解释器与服务器之间设置缓存服务器，以降低网络负荷，减轻服务器的压力，减少响应时间。

从客户端来看，解决网络时延主要有两种方法，一是采用缓存机制；另一种就是采用预取技术，即在需要某个资源之前就将其从服务器上取过来等待使用。本文主要研究在 VoiceXML 语音平台中的预取技术及其方案。

2　相关工作及本文贡献

预取技术包括了两个关键问题，一是如何尽可能准确地预测用户将需要哪些资源，即预取预测算法；二是采用何种预取的调度机制进行预取操作，即预取调度算法。近年来，众多学者对预取技术进行了研究，主要也集中在上述两个关键问题上。

2.1 预取预测算法

目前研究的预测算法主要有下面几类:

(1) 基于热点的预测　基于热点的预测算法认为每个服务器上都存在若干个最受用户喜爱的页面，它们被访问的次数远远高于其他页面。该算法比较简单，它通过代理服务器聚齐用户的请求[3]，当用户对网页的访问有明显的热点时，该算法的效果非常好，预取的页面有很高的利用率，网络通信量的增加较少。但是该算法只预取访问次数较多的页面，不会顾及访问次数较少的页面。另外，对于单个用户而言，基本不存在热点页面，故该算法不适用于单个用户的预取。而且该算法与具体的访问过程无关，不能根据用户的请求预测某一次访问中的下一个请求，故连续访问将得不到较好的响应。

(2) 基于链接的预测　基于链接的预取认为用户的下一个请求往往来自当前页面的链接，所以可以通过预取当前页面的部分链接[4]以缩短对用户请求的响应时间。因为网页上链接很多，预取所有链接显然会导致很低的预取利用率，极大增加网络通信量，浪费网络带宽，文献[5，6]加入历史信息来帮助选择预取哪些链接。

该算法容易实现，但它同等对待每个链接，预取利用率较低，不适合作为多用户的预取预测手段。文献[7，8]的研究表明，用户在进行浏览时约有 20%可能性采用输入地址或使用书签等手段浏览下一个网页。显然，基于链接的预测无法应对这种情况。

(3) 基于访问序列的预测　基于访问序列的预测认为用户的访问模式相同或相似，故可利用访问历史记录建立用户的访问序列，根据该序列与当前访问序列进行比较，预测用户将要访问的资源。

文献[9，10]从服务器挖掘用户的访问日志得到用户的访问序列并建立Markov链，将此序列与当前访问序列匹配，然后预取匹配序列中其他页面。为了减少复杂度，提高准确性，可以加入设置预取门限概率[11]，联合多个Markov模型[12]，以及采用一定的算法精简模型状态空间[13]。文献[14]从服务器日志中得到用户的访问序列，通过建立访问树并与当前访问树通过兼容性计算进行匹配，预取匹配访问树中没有被访问过的节点。文献[15]在访问树中增加了概率及高度参数。文献[16]则根据用户的访问序列构造权值树，把当前用户的访问序列通过逐步后退编码算法与该权值树匹配，预取匹配最长的序列中的所有元素。

基于访问序列的预测需要比较复杂的模型，并且可以预取用户若干时间以内可能需要的页面，大大减少用户的等待时间，但是该算法准确性不高，预取利用率较低，对网络带宽有较大的浪费。

(4) 基于访问概率的预测　基于访问概率的预测算法认为用户的访问具有一定的规律性，区别于基于访问序列的预测算法，该算法更主要研究下一个资源的预取情况。文献[17−21]定义用户访问资源 A 后访问资源 B 的概率为

$$P(B \mid A) = C_{A,B} / C_A \tag{1}$$

式中 C_A 是资源 A 的计数器，表示资源 A 被访问的次数，$C_{A,B}$ 表示资源 A 到资源 B 的跳转计数器，表示在 A 被访问后，资源 B 随即被访问的次数。故当用户访问 A 时，就可以通过统计数据得到以前访问各后续资源的概率，从而决定是否需要预取。

基于访问概率的预测算法需要进行计数器的维护，会存在大量数值很少的跳转计数器。但是该算法的使用和计算都非常的简单快捷，预测的准确率较高。

(5) 基于兴趣的预测　基于兴趣的预测算法认为用户的相邻的请求具有相关或相近的内容。文献[22]利用数据挖掘技术分析用户访问过的页面，分析其兴趣关联规则作为对用户即将访问的页面进行预取的依据。

该算法预测准确度较高，但是需要积累大量关于某一特定用户的知识，所以并不适合预测多用户的访问，并且当用户的兴趣发生改变时，该算法反应速度较慢，往往需要进行较长时间的调整。

2.2 预取调度算法

目前对预取调度算法的研究还比较单一，主要集中在如何利用调度算法确定预取门限的问题上，文献[17]中就计算了在循环(Round-Robin)处理机共享调度模型下的预取门限。

2.3 本文贡献

根据 VoiceXML 与 HTML 的区别，语音平台与 Web 的不同，本文提出预取的对象不应该是 VoiceXML 页面，而应该是 VoiceXML 中所需要的语音资源。并且本文根据用户访问的历史数据计算各资源之间的转移概率，建立自适应的多用户共享的 Markov 链，由平台统一计算所有在线用户下一步将要被访问的资源及其概率，将多个用户对同一资源的需求进行叠加，正确统计其访问概率，故能得到更好的预测准

确度。本文还提出采用抢占式优先级调度算法对预取任务进行调度，将资源的访问概率作为优先级参与排队进行预取调度，以得到较优的预期用户等待时间。

3 预取算法

VoiceXML 与 HTML 有很大的区别，它们的访问设备，提供平台与执行平台也不一样，如表 1 所示。

所以，VoiceXML 的预取与 Web 的预取也存在很大的差异，在预取的对象，预测的算法等各方面均有不同。

表 1 VoiceXML 与 HTML 访问，提供，执行过程的区别

		VoiceXML	HTML
1	访问	电话访问，语音平台解释执行，多人共享平台	浏览器访问并解释执行，无需共享浏览器
2	跳转	无法任意跳转，只能按照既定顺序选择跳转	可以通过输入地址等手段进行任意跳转
3	交互	通过语音交互，需要按序逐条播放语音	通过文字图像交互，文字图像需要同时显示
4	生成	VoiceXML 注重与用户交互，存在更多动态内容	存在很多一旦生成就不再变化的静态网页
5	在线	平台可以知道用户是否在线， 是否结束浏览	代理及服务器无法获知用户是否在线

3.1 预取对象

一般说来，对 HTML 的预取往往是指页面及其页面元素(包括图片，动画，声音等)的预取，所有的元素必须以页面为单位。而由表 1 中第 4 项可知，预取 VoiceXML 页面有很大的难度，因为它需要根据用户的输入动态的产生，比如目前非常流行的彩铃业务，彩铃排行榜随时都在变化，预取的页面容易因过期而导致无效。而 VoiceXML 中的语音资源文件相对页面本身而言较大，是影响用户等待时间的关键因素，并且改变较少。

而根据表 1 中第 3 项，用户必须逐条地听取语音，所以平台对下一条语音的需求有一定的时间间隔，在这个间隔中，平台可以利用空闲的资源来获取下一条语音。

所以，在 VoiceXML 语音平台中，VoiceXML 页面中引用的语音更加适合作为预取的对象。

3.2 预测算法

在一个 VoiceXML 页面中，往往存在多个对话(Dialog)，比如<form>，<menu>等元素，其中每一个对话都引用一个或者多个语音资源。对话是 VoiceXML 执行跳转的最小单位，是与用户交互的主体，根据 Web 网页的研究经验，用户在对话之间跳转也应具有 Markov 性。故令最小预取单位为一个对话的语音资源，可以建立 Markov 链。

为了方便描述，下面给出几个定义:

定义 1 对话是 VoiceXML 执行跳转的最小单位， 记为 d。令 $d=\{v_0,v_1,\cdots,v_k,\cdots\}$。其中 v 表示语音资源，$k=0,1,2,\cdots,K$ 表示该对话中包含语音资源的个数，本文将此 K 个资源看成一个整体，故下文中不再区分对话及其资源。

定义 2 状态集合即用户访问过的所有的对话的集合，记为 $D=\{d_1,d_2,\cdots,d_n,\cdots\}$，其中 $n=1,2,\cdots,N$ 表示状态集合中对话的个数。

定义 3 状态转移概率表示用户从某一对话 d_i 转移到另一对话 d_j 的概率，记为 $p_{i,j}=P\{d_j|d_i\}=C_{i,j}/C_i$，其中 C_i 是对话 d_i 的计数器，$C_{i,j}$ 是从 d_i 到 d_j 的转移计数器。状态转移概率组成 $N\times N$ 的一步转移概率矩阵，记为 $\boldsymbol{P}(1)$。

所以，利用历史数据信息很容易构造此 Markov 链。注意到表 1 中的第 2 项，用户不能任意跳转，必须按照预定的顺序访问 VoiceXML 的对话，故必有

$$\sum_{j=1}^{N}p_{i,j}=\sum_{j=1}^{N}P\{d_j\mid d_i\}=\frac{\sum_{j=1}^{N}C_{i,j}}{C_i}\le 1 \tag{2}$$

其中 N 是状态集合中对话的个数。这样也不会统计到一些特殊的，基本不再重复发生的情况，比如因用户输入地址而跳转。

为了减少 Markov 模型的状态空间，并让其能适应用户兴趣和 VoiceXML 页面等的变化情况，可以定时检查一步转移概率矩阵 $\boldsymbol{P}(1)$，将其小于一定门限值的转移概率置为零。

一般地，若某一用户在访问对话 d_i，则他将要访问对话 d_j 的概率是 $p_{i,j}$。对于任意的 $j\in(0,1,2,\cdots,N)$，若 $p_{i,j}$ 大于某一门限值 T，则认为该对话需要预取。这就是单用户的 Markov 预测模型 (Single-user Markov Predict Model, SMPM)。

但是，若有 k 位用户在同时访问同一对话 d_i 时，则此时系统访问 d_j 的概率不再是 $p_{i,j}$，而是 k 位用户中任一位访问 d_j 的概率，即

$$1-(1-p_{i,j})^k \tag{3}$$

在Web环境下，位于代理服务器上的预测系统无法获知用户的在线状态，也无法获知用户是否结束浏览，故无法得知有多少位用户正在访问同一个Web页面。根据表 1 的第 5 项，语音平台可以获知用户是否在线，所以在实际的预取算法的过程中，能够统一计算所有在线用户的预取概率。假设当前时刻系统中有M个用户在线，则每个用户必定正在访问某一对话d。令k_i表示正访问对话d_i的用户数，则令其初始分布为

$$p(0)=\{k_1,k_2,\cdots,k_n,\cdots\} \tag{4}$$

其中 $n=1,2,\cdots,N$。故此时可通过一步转移概率矩阵求得 $p(1)$， 如下

$$p(1)=\{p_1^1,p_2^1,\cdots,p_n^1,\cdots\}=p(0)\circ\boldsymbol{P}(1) \tag{5}$$

其中运算符。定义为

$$p_i^1=1-\prod_{j=1}^{N}(1-p_{j,i})^{k_j} \tag{6}$$

其中 $i=1,2,\cdots,N$ 表示对话的个数。

求得 $p(1)$之后，访问概率大于门限值 T 的对话 d 即是需

要预取的对象，将其加入到预取调度队列中等待调度。

如果用户的状态发生了改变，从某一对话 d_i 转移到另一对话 d_j，则需要完成下面的工作：

(1)若新对话 d_j 不在状态集合 D 中，则需要将其加入到 D 中，并扩展一步转移概率矩阵 $\boldsymbol{P}(1)$，初始化其转移概率。

(2)更新计数器 $C_{i,j}$ 与 C_j。

(3)更新一步转移概率矩阵 $\boldsymbol{P}(1)$。

(4)根据式(4)更新 $p(0)$，利用式(5)重新计算 $p(1)$，并根据 $p(1)$中各对话的概率更新预取调度队列。

因为语音平台可以获知用户在线或浏览结束的状态，而统一计算所有用户的预取请求概率，从而减少重复预取的可能，并能将多个用户对某些资源的访问概率进行叠加，这样更有助于提高预测的准确率，减少用户的等待时间。以上预取模型即多用户共享 Markov 预测模型(Multi-user shared Markov Predict Model，MMPM)。

3.3 调度算法

预测算法解决了预取过程中预取哪些对象的问题，调度算法则要解决如何预取的问题。

不妨假设某用户使用电话接入语音平台访问某一对话，平台播放语音并等待用户的输入，在这个交互时间间隔中可以进行预取工作，令其时长为 I; 预取完一个资源 i 的时长设为 F_i; 用户输入完毕，应用开始请求下一个资源，直到该请求被满足的时长为 W，表示用户的等待时间。则它们之间的关系如图 2 所示。其中图 2(a)表示 $F_i \le I$ 的情况，图 2(b)表示 $F_i > I$ 的情况，此时资源 i 的预取并没有完成。

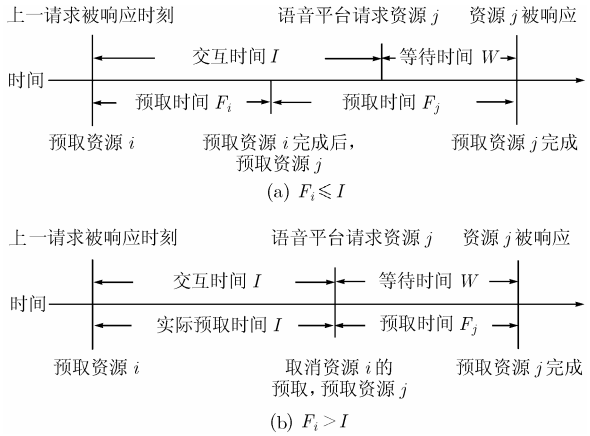

图 2　各时间之间的关系

显然，用户等待时间的长短取决于预测的准确度及资源的预取时间。在循环(Round-Robin，RR)调度模型[17]中，让调度队列中每一个任务都分享相等的处理机时间，忽略了预取任务的概率不同，即忽视了预测的准确度，可能导致用户等待的时间更长，浪费大量的时间取回无用的数据。

定理 1　先预取访问概率较大的资源获得的收益要大于等于先预取访问概率较小的资源。

证明　不妨设预取调度队列中资源 d_i 的访问概率为 p_i,d_j 的访问概率为 p_j，其中 $p_i > p_j$。F_i,F_j 分别是资源 d_i,d_j 的预取时间，F 为访问除资源 d_i,d_j 之外的其他资源时的平均预取时间。用户的交互时间为 I。

预取的收益主要表现为用户等待时间的减少。如果不预取的话，用户等待时间的期望应是

$$E(W) = p_iF_i + p_jF_j + (1 - p_i - p_j)F \tag{7}$$

下面分 4 种情况讨论:

(1) $I \le \min(F_i,F_j)$　如果先预取 d_i，则用户等待时间的期望是

$$E_i(W) = p_i(F_i - I) + p_jF_j + (1 - p_i - p_j)F \tag{8}$$

而先预取 d_j 时用户等待时间的期望为

$$E_j(W) = p_iF_i + p_j(F_j - I) + (1 - p_i - p_j)F \tag{9}$$

式(8)与式(9)相减，可知

$$E_i(W) - E_j(W) = (p_j - p_i)I < 0 \tag{10}$$

(2) $\min(F_i,F_j) < I \le \max(F_i,F_j)$　此时分成两种情况，当$F_j<I\le F_i$时，先预取 d_i 时与先预取 d_j 时用户的等待时间的期望之差是

$$\begin{aligned} E_i(W) - E_j(W) &= p_i(F_i - I) + p_jF_j - p_i(F_i - (I - F_j)) \\ &= (p_j - p_i)F_j < 0 \end{aligned} \tag{11}$$

当 $I > F_i + F_j$ 时，先预取 d_i 时与先预取 d_j 时用户的等待时间的期望之差是

$$\begin{aligned} E_i(W) - E_j(W) &= p_j(F_j - (I - F_i)) - (p_iF_i + p_j(F_j - I)) \\ &= (p_j - p_i)F_i < 0 \end{aligned} \tag{12}$$

(3) $\max(F_i,F_j) < I \le F_i + F_j$　此时先预取d_i时与先预取d_j时用户的等待时间的期望之差是

$$\begin{aligned} E_i(W) - E_j(W) &= p_j(F_j - (I - F_i)) - (p_i(F_i - (I - F_j)) \\ &= (p_j - p_i)(F_i + F_j - I) \le 0 \end{aligned} \tag{13}$$

(4) $I > F_i + F_j$　此时可以在I时间内完成资源 d_i,d_j 的预取，所以收益一致，均为 $(1 - p_i - p_j)F$。

综上，在第(1)，第(2)种情况中，先预取 d_i 时用户的等待时间都要小于先预取 d_j 时用户的等待时间，即前者收益要大于后者；在第(3)种情况下，前者收益不小于后者；第(4)种情况下，两者收益一致。故定理 1 得证。　　证毕

由定理 1 可知，每次都预取访问概率最大的资源才能得到最大的收益。显然，处于等待时间的资源的访问概率为 1，即如果某个用户正在访问对话 d_i，则资源 d_i 访问概率为 1。若此刻 d_i 尚不可用，则获取 d_i 是此刻最迫切的任务。而采用循环调度方法则可能会将最迫切的任务暂停而将时间分配给访问概率较低的任务，反而导致用户等待的时间延长，失去预取的意义。

采用抢占式优先级(Preemptive Priority，PP)调度方法进行调度显然是更好的选择，同时，为了更好的公平调度，不妨将优先级设为 0~10 级，其中 0 表示最高优先级。将资源被访问的概率通过式(14)映射为优先级，其中 p 是资源被

访问的概率，⌈⌉为上取整函数。

$$\mathrm{PRI}=\lceil(1-p)\times 10\rceil \tag{14}$$

将调度队列按优先级分成 11 个队列，从优先级最高的队列开始调度，同一队列中采用循环调度法进行处理机分享，直到优先级为 n 的队列为空时，优先级为 n+1 的队列才可以参与调度。而由式(14)可知，其中 0 级队列中只会存在访问概率为 1 的预取资源，总处于最优先调度的位置。

因为用户的状态会发生变化，所以将会不断有新的预取任务加入到队列中，并且已有的任务的优先级也会不断地发生变化。如果某一个任务的优先级大于目前正在运行任务，则它会抢占当前优先级队列的处理机时间，以确保先完成高优先级的任务。任务完成之后，将其从调度队列中删除。如果某一个任务在队列中的时间超过了一个门限值 L，则认为此预取任务已不再需要，将此任务删除以免调度队列中任务过多。

4 仿真及结果分析

为了对上述的算法进行验证，本文对下面 4 种情况进行了仿真。

(1) SMPM 与 RR 模型配合(SMPM-RR);

(2) SMPM 与 PP 模型配合(SMPM-PP);

(3) MMPM 与 RR 模型配合(MMPM-RR);

(4) MMPM 与 PP 模型配合(MMPM-PP)。

假设用户的到达时间间隔满足参数为 λ 的负指数分布，即用户的平均到达时间间隔为 λ；用户交互时间满足负指数分布，平均时间为 3s；对话数目为 100 个，每个对话的下载时间也满足负指数分布，平均时间为 8s；系统中只有一个预取引擎。每次实验都采用同样的缓存策略：若某一时刻系统对某一对话资源没有任何需求(包括预取需求)，则将其从缓存中清除。

如图 3 所示是各算法在不同的门限值 T 下的平均时延，此时用户的平均到达时间间隔 λ =10。显然，在允许资源预取的情况下，用户的平均时延要低于对话的平均下载时间。所有的 4 种情况下，平均时延随着门限值 T 的增加逐步增加。门限值 T 增加意味着每次预取资源的数目减少，导致预取命中率下降，从而增加平均时延。

图 4 是不同用户平均到达时间间隔下的平均时延情况，此时门限值 T 为 0.20。当用户的到达时间间隔不断增加时，采用 SMPM 的算法平均时延变化较少，特别是 SMPM-RR 算法。SMPM-PP 算法不断向 SMPM-RR 靠近是因为预取文件的访问概率不断降低，预取优先级之间的差别不断减少。但是预取优先级的差异不可能消失，所以 SMPM-PP 的平均时延始终要小于 SMPM-RR 的平均时延。

用户平均到达时间间隔对 MMPM 的影响非常大是因为 MMPM 与当前的在线用户数量相关，当系统中只有一个用户在线时 MMPM 将与 SMPM 等价。

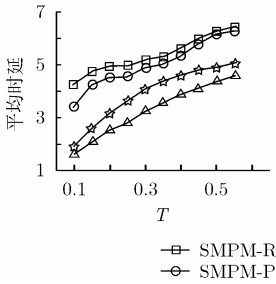

图 3 不同门限值 T 下的平均时延

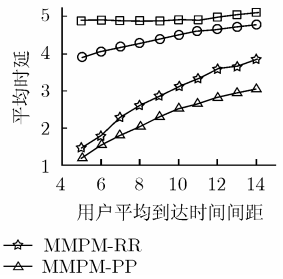

图 4 不同用户平均到达时间间隔下的平均时延

从图 3，图 4 可知，PP 调度模型要优于 RR 调度模型，MMPM 预测模型要优于 SMPM 模型，MMPM 预测模型结合 PP 调度模型能得到更佳的效果。

5 结束语

本文分析了目前预取算法的优劣，并根据 VoiceXML 语音平台与以 HTML 为基础的 WWW 之间的区别，认为在 VoiceXML 语音平台中需要将 VoiceXML 页面中引用语音资源作为预取的对象，并提出一种自适应的多用户共享的 Markov 链，可以统一计算所有在线用户下一步所需的资源及其概率，提高预测的准确率。进一步提出采用抢占式优先级调度算法对预取任务进行调度，将资源被访问的概率作为优先级参与排队进行预取调度。仿真研究表明，多用户共享的 Markov 预测模型的平均用户时延要低于单用户 Markov 预测模型，基于抢占式优先级的调度方法能比循环调度模型更好地减少用户请求的访问延迟，提高响应速度。多用户共享的 Markov 预测算法与抢占式优先级调度模型结合能取得更好的效果。

预取离不开缓存的支持，所以下一步的工作将是研究缓存的一些关键问题，以便能更好地减少用户的等待时间，降低网络的通信量，降低服务器的负荷。

参 考 文 献

[1] Scott M, Daniel C B, and Jerry C, *et al*.. Voice Extensible Markup Language (VoiceXML) Version 2.0. http://www.w3.org/ TR/ voicexml20/, 2004, 3.

[2] 龚双瑾, 刘多. 移动与 IP 智能网. 第一版, 北京: 人民邮电出版社, 2004: 161-169.

[3] Marcatos E P and Chronaki C E, A top-10 approach to prefetching the web. The Eighth Annual Conference of the Internet Society, Geneva, INET, 1998, http://www.isoc.org/inet98/proceedings/1i/1i_2.htm.

[4] Chinen K and Yamaguchi S. An interactive prefetching proxy server for improvement of WWW latency. The Seventh Annual Conference of Internet Society, Kuala Lumpur, INET, 1997: 473-478.

[5] Duchamp D. Prefetching hyperlinks. The Second USENIX Symposium on Internet Technologies and Systems, Boulder, USENIX, 1999: 127–138.

[6] Davison B D. Predicting Web actions from HTML content. The Thirteenth ACM Conference on Hypertext and Hypermedia, Maryland, ACM, 2002: 159–168.

[7] Tauscher L and Greenberg S. How people revisit Web pages: Empirical findings and implication for the design of history systems. *International Journal of Human-Computer Studies*, 1997, 47(1): 97–137.

[8] Davison B D. Web traffic logs: An imperfect resource for evaluation. The Ninth Annual Conference of the Internet Society, San Jose, INET, 1999, http://www.isoc.org/ inet99/ proceedings/4n/4n_1.htm.

[9] Su Z and Yang Q. Whatnext: A prediction system for Web requests using n-gram sequence models. The First International Conference on Web Information System and Engineering Conference, Hong Kong, WISE, 2000: 200–207.

[10] Yang Q, Zhang H H, and Li T Y. Mining Web logs for prediction models in WWW caching and prefetching. The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'01, San Francisco, ACM, 2001, 473–478.

[11] 朱培栋, 卢锡成, 周光铭. 基于客户行为模式的 Web 文档预送. 软件学报, 1999, 11(10): 1142–1147.

[12] Pitkow J and Pirolli P. Mining longest repeating subsequences to predict World Wide Web surfing. The Second USENIX Symposium on Internet Technologies & Systems, Boulder, USENIX, 1999: 139–150.

[13] Deshpande M and Karypis G. Selective Markov models for predicting Web-page accesses. *ACM Trans. on Internet Technology*, 2004, 4(2): 163–184.

[14] Lei H and Duchamp D. An analytical approach to file prefetching. USENIX 1997 Annual Technical Conference, Anaheim, USENIX, 1997: 275–288.

[15] Fan L, Cao P, and Jacobson Q. Web prefetching between low-bandwidth clients and proxies: potential and performance. The ACM SIGMETERICS'99, Atlanta, ACM, 1999: 178–187.

[16] Schechter S, Krishnan M, and Smith M D. Using path profiles to predict HTTP requests. *Computer Networks and ISDN Systems*, 1998, 30(1-7): 457–467.

[17] Jiang Z and Kleinrock L. An adaptive network prefetch scheme. *IEEE Journal on Selected Areas in Communications*, 1998, 16(3): 358–368.

[18] Jiang Z and Kleinrock L. Web prefetching in a mobile environment. *IEEE Personal Communications*, 1998, 5(5): 25–34.

[19] Jiang Z and Kleinrock L. Prefetching links on the WWW. The 1997 IEEE International Conference on Communications, Montreal, IEEE, 1997: 483–489.

[20] Conhen E, Kristhnamurthy B, and Rexford J. Efficient algorithms for predicting requests to Web servers. The IEEE INFOCOM'99, New York, IEEE, 1999: 284–293.

[21] Padmanabhan V N and Mogul J C. Using predictive prefetching to improve World Wide Web latency. *Computer Communication Review*, 1996, 26(3): 22–36.

[22] 徐保文, 张卫丰. 数据挖掘技术在 Web 预取中的应用研究. 计算机学报, 2001, 24(4): 430–436.

王文林：男，1979 年生，博士生，研究方向为多媒体通信、下一代网络增值业务.

廖建新：男，1965 年生，教授，博士生导师，主要研究领域为通信软件、增值业务提供技术.

朱晓民：男，1974 年生，博士，副研究员，主要研究领域为智能网、下一代业务网络、协议工程.

王　纯：男，1970 年生，高级工程师，研究方向为智能网、下一代业务网络、通信软件.