基于 VoiceXML 的语音平台缓存一致性控制算法

王文林, 廖建新, 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)

摘 要: 在基于 VoiceXML(Voice eXtensible Makeup Language)的语音平台上使用缓存有利于减少网络带宽的消耗, 降低服务器的负荷, 缩短用户的等待时间. 通过分析语音平台的缓存模型, 提出了拟合预测算法, 对随机分布进行参数拟合, 预测一定时间内 VoiceXML 文档修改的概率, 从而确定缓存中的文档是否有效. 仿真研究表明, 该算法能得到低于 1/10000 的文档过期率以满足语音呼叫的需求, 同时能有效提高系统性能, 优于 Alex 算法.

关键词: 缓存一致性; 参数拟合; VoiceXML; 负指数分布; 对数正态分布; Pareto 分布
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2007) 04-0640-07

Consistency Control Algorithm for the Cache of Voice Platform Based on VoiceXML

WANG Wen-lin, LIAO Jian-xin, ZHU Xiao-min
(*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*)

Abstract: Using cache in the voice platform based on VoiceXML (Voice eXtensible Makeup Language) could reduce the cost of network bandwidth, server load and delay of request. Through analyzing the cache model of voice platform, the Fitting & Predicting algorithm is proposed. It estimates the validity of the cached document by performing parameter fits to stochastic distribution and predicting the change probability of VoiceXML documents within a certain period. Simulated research indicates the algorithm surpasses the Alex protocol and can obtain a stale ratio lower than 1/10000 to meet the demand of the voice platform while effectively enhancing the system performance.

Key words: cache consistency; parameter fits; VoiceXML; exponential distribution; lognormal distribution; Pareto distribution

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, 将话音业务和数据业务相融合, 将业务表现和业务逻辑相分离, 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. 用户通过电话接入语音平台, 则启动一个 VoiceXML 解释器实例, 访问文档服务器以获得 VoiceXML 文档, 经解释器解释后由实现平台执行, 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 文档的时间与解释执行此文档时间之和, 在这三个时间中, 从 Web 服务器上获取文档的时间是最长的, 也是最不可控的.

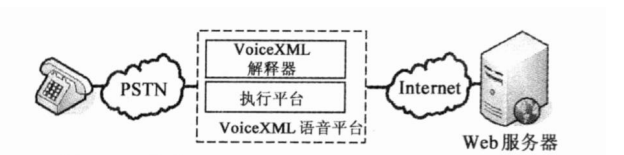


图 1 一种基于 VoiceXML 的语音平台的体系架构

而且, 提供业务的服务器通常都需要实时或定时更新的数据和内容, 以向用户提供个性化的服务, 甚至每一个请求都需要执行相应的脚本或程序生成回应的页面, 如广为使用的 ASP, JSP, PHP 等技术. 当业务访问量增加时, 服务器的负荷随之增加, 对请求的响应时间也随之增加, 而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

收稿日期: 2005-10-24; 修回日期: 2006-09-06
基金项目: 国家杰出青年科学基金(No. 60525110); 国家 973 高技术研究发展规划(No. 2007CB307100, No. 2007CB307103); 新世纪优秀人才支持计划(No. NCET-04-0111); 高等学校博士学科点专项科研基金(No. 20030013006)

入的客户更易因服务质量(QoS)流失. 所以, VoiceXML 论坛建议: (1) 在一个 VoiceXML 文档中包含尽量多的对话, 减少与文档服务器交互次数; (2) 在 VoiceXML 解释器与文档服务器之间设置缓存服务器, 以降低网络负荷, 减轻文档服务器的压力, 减少响应时间.

因为服务器上数据不断更新, VoiceXML 文档也随之更新, 所以必须考虑缓存与服务器之间的同步, 保持缓存与服务器之间的数据一致.

2　相关工作及本文贡献

对缓存的一致性控制一般分成两类: 一类是弱一致性(Weak Consistency)算法, 允许用户访问一定量的过期数据, 在一定约束条件下采取尽力而为策略保证缓存与服务器的数据同步, 主要用于对时延特别敏感的场合; 另一类是强一致性(Strong Consistency)算法, 主要在对数据特别敏感的情况下使用, 确保用户得到最新的数据.

弱一致性算法主要思想是由服务器(通过 HTTP 协议中 Expires 参数)或客户端给文档设定一个存活时间(TTL: Time-to-Live), 在此时间内认为缓存中的文档是可用的, 超出此时间后则认为其过期, 需要再次访问服务器以得到最新数据. 文献[3]中提出了 Alex 协议以在一个全局文件系统中实现文件同步, 认为较长一段时间不变化的文件在将来较长的一段时间内也不会变化, 所以应该设定较长的TTL, 而较短时间变化的文件则必须设定较短的TTL. 文献[4]将 Alex 协议应用于 Web 环境, 与固定的 TTL 算法进行比较, 得出 Alex 协议效率更好的结论. 文献[5]提出一种基于数据变化量及变化限制的自适应 TTL 算法, 综合考虑了区间限制, 最近数据变化规律等因素, 并用实验进行了验证. 但文献[5]中采用数据变化量作为一个TTL 的衡量依据, 考虑到 Web 环境下文档的变化量无法计算, 故该算法难以应用于 Web 系统中. 文献[6]认为缓存的一致性与其带来的性能提高之间存在类似反比的关系, 提出应该有一个最优的区域可以达到性能与一致性的均衡. 而文献[7, 8]提出了将一个文档的内容分成多块, 每一块独立维护其一致性的方式.

强一致性算法有两种机制, 一是用户对文档的每一次访问都附加一个 HTTP 协议中的 If-Modified-Since 参数再转发给服务器(Polling-Every-Time), 由服务器来判断缓存中的文档是否过期, 若该文档过期, 则返回状态 200 并发送新文档, 否则, 则返回状态 304 指示客户端继续使用缓存中的文档. 第二种是服务器主动失效(Invalidation)算法, 服务器不断检测文档是否发生变化, 以便在文档变化时通知所有的客户删除缓存中的文档. 文献[9]是最早讨论如何维护一致性的文献之一, 它建议服务器通过推(Push)的方式通知缓存数据发生变更. 文献[10]提出了"两级租约扩张失效"协议, 建议服务器在所有发送给客户的文档上附加一个"租约"时间, 保证在"租约"时间内如果数据改变则会发送失效消息, 并与 TTL, Polling-Every-Time 进行比较, 得出该算法效率最好的结论. 文献[11]提出可以建立页面与数据库之间的依赖关系, 通过监视服务器数据库的 Log 以得知文档是否改变. 文献[12]提出"Web 内容分布协议", 通过服务器与客户之间的两段交互支持各种一致性要求.

文献[13]分析了上述各种一致性控制机制在 Web 环境下的应用情况, 并逐一指出了它们的不足: TTL 会导致用户得到过期的信息; Polling-Every-Time 方法虽然能减少不必要的文件传输, 但是却不能有效减少客户端时延; Invalidation 算法需要修改客户与服务器之间的协议, 其主要问题是服务器必须要保存所有缓存的信息, 并与之通信, 代价较大, 另外, 故障处理问题仍未得到完善解决.

本文认为 VoiceXML 文档修改的时间符合某一随机分布, 并根据语音业务的特点猜测其为负指数分布, 对数正态分布及 Pareto 分布之一. 故在缓存中可以通过该随机分布预测服务器端的文档是否改变, 再决定是否访问服务器.

因为 Pareto 分布不能使用矩估计法和极大似然估计法进行参数估计, 故本文先给出 Pareto 分布参数估计方法, 并证明其有效. 最后给出分布拟合及预测算法(Fitting & Predicting), 利用历史数据分别估计三个分布的参数后, 计算服务器端文档修改的概率以决定是否使用缓存中的文档.

3　随机分布拟合预测算法

本文的目的是在不改变现有服务器, 并满足语音呼叫的前提下, 尽量减少用户的等待时间. 故采用 Invalidation 方法的强一致性算法不满足此要求. 而 Polling-Every-Time 虽然能保证用户获取的文档最新, 但是却无法有效减少客户时延[13], 所以不宜在此使用. 考虑到语音呼叫允许万分之一的呼损, 故假设允许同样概率的过期文档, 在这个前提下, 使用 TTL 方法来实现与服务器之间的缓存一致性维护, 尽量减少用户的等待时间.

3.1　算法模型

文献[14, 15]考察了 100000 个 Web 文档的变化规律, 认为网页修改的时间间隔基本满足负指数分布. 根据这一结论, 本文假设 VoiceXML 文档变化时间间隔也基本满足某一随机分布, 记为 $F(x)$.

根据这一随机分布及检测到的 VoiceXML 文档修改的历史数据, 不难拟合其参数, 并预测在一定时间内

VoiceXML 文档的变化概率, 如果变化概率小于某一预先设定的门限值 α, 则认为缓存内的文档还是最新的文档; 而当变化概率大于 α 时, 则认为此时应该再次访问服务器以确定文档是否改变.

故在语音平台中增加一个缓存模块, 如图 2 所示.

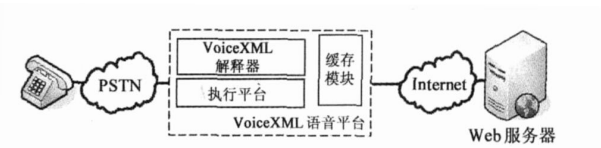


图 2 增加缓存模块的语音增值平台

3.2 参数拟合

根据文献[14, 15]的结论, Web 文档修改的时间间隔满足负指数分布. 考虑当用户使用电话访问语音业务的时候, 需要更长的时间去熟悉语音业务的流程. 所以, VoiceXML 文档必定有以下特点:

(1)VoiceXML 文档的改变频率肯定要比 Web 页面小, 即VoiceXML 文档比较稳定;

(2)一旦 VoiceXML 文档发生改变, 那么它将持续一段较长的时间, 以便语音业务的用户熟悉并使用新的流程.

故不妨假设VoiceXML 文档变化的时间间隔满足负指数分布, 对数正态分布或 Pareto 分布.

参数估计一般使用点估计法, 所谓点估计, 就是在总体 X 的分布函数 $F(x;\theta_1,\theta_2,\cdots,\theta_k)$ 的形式已知的情况下, $X_1, X_2, \cdots, X_n$ 为 X 的一个样本, $x_1, x_2, \cdots, x_n$ 是相应的样本值, 构造适当的统计量 $\theta_1(X_1, X_2, \cdots, X_n)$, $\theta_2(X_1, X_2, \cdots, X_n)$, …, $\theta_k(X_1, X_2, \cdots, X_n)$, 用其观察值, $\theta_1(x_1, x_2, \cdots, x_n)$, $\theta_2(x_1, x_2, \cdots, x_n)$, …, $\theta_k(x_1, x_2, \cdots, x_n)$ 来估计未知参数 $\theta_1, \theta_2, \cdots, \theta_k$ 的方法, 主要有矩估计法和极大似然估计法[16].

对于负指数分布 $F(x)=1-\exp\left(-\frac{x}{\lambda}\right)$, 可以由矩估计法和极大似然估计法得出同样的估计值 $\hat{\lambda}=\frac{1}{n}\sum_{i=1}^{n}x_i$.

对于对数正态分布 $F(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\cdot\exp\left(-\frac{(\ln x-\mu)^2}{2\sigma^2}\right)$, 直接使用矩估计法比较困难, 但根据对数正态分布的定义, 可知当 x 满足对数正态分布时, $\ln x$ 满足同参数的正态分布, 故可针对 $\ln x$ 求估计量. 得: $\hat{\mu}=\frac{1}{n}\sum_{i=1}^{n}\ln x_i$; $\hat{\sigma}^2=\frac{1}{n}\sum_{i=1}^{n}(\ln x_i)^2-\hat{\mu}^2$. 使用极大似然估计法将得到同样的结果.

Pareto 分布 $F(x)=1-\left(\frac{b}{x}\right)^a$, 其中 $b>0$, $a>0$, 当 $a>1$ 时其均值存在, 当 $a>2$ 时其方差存在. 一般说来, 文档修改的时间间隔存在均值, 但不一定存在方差, 故 $a>1$ 但不一定 $a>2$. 所以此时无法用矩估计法来估计 Pareto 分布的参数值. 在使用极大似然估计法来估计 Pareto 分布的参数时, 有:

$$L(a,b)=\prod_{i=1}^{n}\frac{ab^a}{x_i^{a+1}}$$

$$\Rightarrow \ln L(a,b)=\sum_{i=1}^{n}(\ln a+a\ln b-(a+1)\ln x_i)$$

令 $\frac{\partial \ln(a,b)}{\partial b}=0$, 则有 $\frac{a}{b}=0$, 这与 Pareto 分布的定义矛盾, 故也无法使用极大似然估计法. 下面证明两个定理以得到 Pareto 分布的参数估计方法.

定理 1 Pareto 分布 $F(x)=1-\left(\frac{b}{x}\right)^a$ 中 a, b 的估计量 $\hat{a}=\frac{n\bar{X}-C}{n(\bar{X}-C)}$, $\hat{b}=\frac{(n-1)\bar{X}C}{n\bar{X}-C}$ 是无偏估计量, 其中 $\bar{X}=\frac{1}{n}\sum_{i=1}^{n}X_i$, $C=\min(X_i)$.

证明: C 的分布为 $F(x)=1-\prod_{i=0}^{n}(1-F_i(x))=1-\left(\frac{b}{x}\right)^{an}$, 故 $E(C)=\frac{nab}{na-1}$, 而 $E(\bar{X})=E(X)=\frac{ab}{a-1}$, 所以:

$$E(\hat{a})=\frac{E(n\bar{X}-C)}{E(n(\bar{X}-C))}=\frac{nE(\bar{X})-E(C)}{n(E(\bar{X})-E(C))}$$

$$=\frac{\frac{nab}{a-1}-\frac{nab}{na-1}}{n\left(\frac{ab}{a-1}-\frac{nab}{na-1}\right)}=\frac{na^2b-ab-a^2b+ab}{na^2b-ab-na^2b+nab}$$

$$=\frac{a^2b(n-1)}{ab(n-1)}=a$$

$$E(\hat{b})=\frac{(n-1)E(X)E(C)}{nE(X)-E(C)}=\frac{(n-1)\frac{na^2b^2}{(a-1)(na-1)}}{\frac{nab}{a-1}-\frac{nab}{na-1}}$$

$$=\frac{(n-1)na^2b^2}{n^2a^2b-nab-na^2b+nab}$$

$$=\frac{(n-1)na^2b^2}{(n-1)na^2b}=b$$

故定理 1 得证.

定理 2 Pareto 分布 $F(x)=1-\left(\frac{b}{x}\right)^a$ 中 a, b 的估计量 $\hat{a}=\frac{n\bar{X}-C}{n(\bar{X}-C)}$, $\hat{b}=\frac{(n-1)\bar{X}C}{n\bar{X}-C}$ 是一致估计量, 其中 $\bar{X}=\frac{1}{n}\sum_{i=1}^{n}X_i$, $C=\min(X_i)$.

证明: 由定理 1 的证明过程可知 $\lim_{n\to\infty}\bar{X}=\frac{ab}{a-1}$, 且 $\lim_{n\to\infty}C=\frac{nab}{na-1}=b$ 所以:

$$\lim_{n\to\infty}\hat{a}=\lim_{n\to\infty}\frac{n\bar{X}-C}{n(\bar{X}-C)}=\lim_{n\to\infty}\frac{\bar{X}-C/n}{\bar{X}-C}$$

$$= \frac{\lim_{n\to\infty}\bar{X} - \lim_{n\to\infty}\frac{C}{n}}{\lim_{n\to\infty}\bar{X} - \lim_{n\to\infty}C} = \frac{\frac{ab}{a-1} - 0}{\frac{ab}{a-1} - b} = a$$

$$\lim_{n\to\infty} b = \lim_{n\to\infty}\frac{(n-1)\bar{X}C}{n\bar{X} - C} = \lim_{n\to\infty}\frac{\frac{n-1}{n}\bar{X}C}{\bar{X} - C/n}$$

$$= \frac{\lim_{n\to\infty}\frac{n-1}{n}\bar{X}C}{\lim_{n\to\infty}\bar{X} - \lim_{n\to\infty}\frac{C}{n}} = \frac{\lim_{n\to\infty}\bar{X}C}{\lim_{n\to\infty}\bar{X}} = b$$

故定理 2 得证.

由定理 1, 2 可知, 估计量 $\hat{a} = \frac{n\bar{X} - C}{n(\bar{X} - C)}$, $\hat{b} = \frac{(n-1)\bar{X}C}{n\bar{X} - C}$ 既是无偏估计量, 也是一致估计量, 可以用来估计 Pareto 分布的参数, 其中 $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $C = \min(X_i)$.

3.3　预测算法

令当前缓存中文档的修改时间 T_{doc}, 缓存到服务器的往返时间的一半为 ΔT(这里假设缓存到服务器与服务器到缓存的传输时间相等). 当用户在某一个时刻 T 访问缓存时, 可以假定他想得到 $T + \Delta T$ 时刻时最新的文档. 为了保证用户访问的数据最新, 缓存需要估计 T_{doc}到 $T + \Delta T$ 时刻内文档修改的概率

$$P\{X < T + \Delta T - T_{doc}\} = F(T + \Delta T - T_{doc}) \quad (1)$$

如果 $P\{X < T + \Delta T - T_{doc}\} < \alpha$ 则使用缓存中的文档, 如果 $P\{X < T + \Delta T - T_{doc}\} \geqslant \alpha$ 则使用 HTTP 协议中的 If-Modified-Since 参数访问服务器, 根据服务器返回的状态值(200 或 304)确定更新缓存或继续使用缓存.

但式(1)效率十分低下. 假设用户在 T_1 时刻访问缓存, 计算得到 $P\{X < T_1 + \Delta T - T_{doc}\} \geqslant \alpha$, 故该访问被传送至服务器, 若该文档没有改变, 根据 HTTP 协议的规定, 将返回 304, 指示该缓存并未过期, 故此用户将继续使用该缓存中的文档. 假设此时用户在 $T_2(T_2 > T_1)$时刻再次访问缓存, 而 $P\{X < T_2 + \Delta T - T_{doc}\} > P\{X < T_1 + \Delta T - T_{doc}\} \geqslant \alpha$, 此时必定还将访问服务器, 并没有得益于 T_1 时刻对服务器的访问.

所以必须对式(1)加以改进, 考虑下述两种情况:

(1) $T_2 - T_1 \geqslant 2\Delta T$

如图 3(a)所示, 在这种情况下, 用户在 T_2 时刻访问缓存时, T_1 时刻对服务器访问的结果已经返回. 令上次访问服务器的时间为 T_{fresh}, 可知此时 $T_{fresh} = T_1$. 若服务器返回状态 304, 则可知在$(T_{doc}, T_{fresh} + \Delta T)$内该文档未发生变化; 若服务器返回状态 200, 可知该文档已改变, 设改变时间为 T_m, 必有 $T_m \in (T_{doc}, T_{fresh} + \Delta T)$, 此时必须要更新缓存, 并令 $T_{doc} = T_m$, 可知更新 T_{doc}后在$(T_{doc}, T_{fresh} + \Delta T)$内, 该文档也未发生变化.

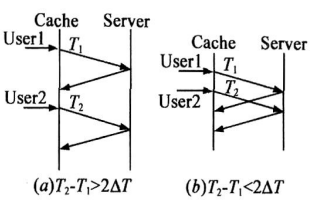


图 3　用户访问序列

所以, 在这种情况下文档改变的概率为

$$P\{X < T_2 + \Delta T - T_{doc} \mid X > T_{fresh} + \Delta T - T_{doc}\} = \frac{F(T_2 + \Delta T - T_{doc}) - F(T_{fresh} + \Delta T - T_{doc})}{1 - F(T_{fresh} + \Delta T - T_{doc})} \quad (2)$$

(2) $T_2 - T_1 < 2\Delta T$

如图 3(b)所示, 在 VoiceXML 的实际应用中, 用户访问缓存的时间间隔非常短, 通常小于缓存到服务器的往返时间. 若此时不再访问服务器, 而等待上次访问服务器的结果, 则在 T_2 时刻得到的文档为 $T_{fresh} + \Delta T$ 时的最新文档, 从而导致无法控制的文档过期.

令缓存得到关于此文档信息的时刻为 T_{return}, 表示最后从服务器获得文档信息的时间. 则需要猜测 T_1 时刻访问服务器时的返回结果:

如果在 T_1 时刻访问服务器时的返回状态为 304, 表示文档在$(T_{doc}, T_{fresh} + \Delta T)$内没有变化, 此时文档改变的概率应是 $P\{X < T_2 + \Delta T - T_{doc} \mid X > T_{fresh} + \Delta T - T_{doc}\}$.

如果服务器返回状态 200, 表示该文档在$(T_{return} - \Delta T, T_{fresh} + \Delta T)$内已经发生了变化, 但是此刻不能得到文档真正的修改时间, 不妨设其为 $T_m \in (T_{return} - \Delta T, T_{fresh} + \Delta T)$, 则此时文档改变的概率为 $P\{X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m\}$, 下面给出确定 T_m 最佳取值的方法.

解: 考虑到本算法首先考虑文档的过期率, 故应提高缓存访问服务器的概率, 所以欲使 T_m 取值最佳, 必需使式 $P\{X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m\}$ 取值最大.

$$P\{X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m\} = \frac{P\{T_{fresh} + \Delta T - T_m < X < T_2 + \Delta T - T_m\}}{P\{X > T_{fresh} + \Delta T - T_m\}} = \frac{F(T_2 + \Delta T - T_m) - F(T_{fresh} + \Delta T - T_m)}{1 - F(T_{fresh} + \Delta T - T_m)} \quad (3)$$

式(3)对 T_m 求导, 得:

$$\left(\frac{F(T_2 + \Delta T - T_m) - F(T_{fresh} + \Delta T - T_m)}{1 - F(T_{fresh} + \Delta T - T_m)}\right)' = \frac{(1 - F(T_2 + \Delta T - T_m))f(T_{fresh} + \Delta T - T_m)}{(1 - F(T_{fresh} + \Delta T - T_m))^2} - \frac{(1 - F(T_{fresh} + \Delta T - T_m))f(T_2 + \Delta T - T_m)}{(1 - F(T_{fresh} + \Delta T - T_m))^2} \quad (4)$$

其中 $f(x)$ 是该分布的分布密度函数. 由概率的性质可知 $F(x)$ 必不大于 1, 故令

$$(1 - F(T_2 + \Delta T - T_m))f(T_{fresh} + \Delta T - T_m) - (1 - F(T_{fresh} + \Delta T - T_m))f(T_2 + \Delta T - T_m) = 0 \quad (5)$$

解方程(5)可知 T_m 取何值时 $P\{X < T_2 + \Delta T - T_m \mid$

$X > T_{fresh} + \Delta T - T_m$} 得到极值, 再判断其是否最大值, 然后根据 $T_m \in (T_{doc}, T_{fresh} + \Delta T)$ 即可确定 T_m 的取值.

作为特例, 若 VoiceXML 文档的修改间隔时间满足负指数分布, 则式(4)恒为0, 即此时概率 $P\{X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m\}$ 与 T_m 无关. 这与负指数分布无记忆的特性吻合.

综上, 若令 T_1 时刻文档改变的概率为 p, 则在 $T_2 - T_1 < 2\Delta T$ 时, 文档修改的概率不会大于:

$$p \times P\{X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m\} + (1-p) \times P\{X < T_2 + \Delta T - T_{doc} \mid X > T_{fresh} + \Delta T - T_{doc}\} \quad (6)$$

其中 $T_m \in (T_{return} - \Delta T, T_{fresh} + \Delta T)$ 是计算得出文档最佳修改时间. 在实际使用时, 由于 T_m 较难计算, 简单起见, 可以取 $T_m = T_{return} - \Delta T$. 此时式(6)变为:

$$p \times P\{X < T_2 + 2\Delta T - T_{return} \mid X > T_{fresh} + 2\Delta T - T_{return}\} + (1-p) \times P\{X < T_2 + \Delta T - T_{doc} \mid X > T_{fresh} + \Delta T - T_{doc}\} \quad (7)$$

若令式(2)、(7)等于门限值 α, 可求得 T_2 即是上述两种情况下 Time-To-Live 的值. 当一个呼叫到达时间大于 T_2 时, 需要访问服务器是否存在更新的文档, 否则使用缓存中的文档.

4 性能研究

在拟合预测算法中, 如何在尽量保证缓存一致性的同时提高系统的性能是首要考虑的问题. 这里将通过与 Alex 算法比较来考察这种能力.

4.1 仿真说明

本次仿真假设用户拨打 VoiceXML 语音平台访问某一业务满足泊松分布, 平均到达率为每秒 1 个用户. 假设缓存位于 VoiceXML 语音平台上, 即语音平台到缓存的时延为零, 缓存到服务器之间的往返时间为 3 秒, 即 $\Delta T = 1.5$ 秒. 同时假设 VoiceXML 文档平均 1 天(86400 秒)修改一次.

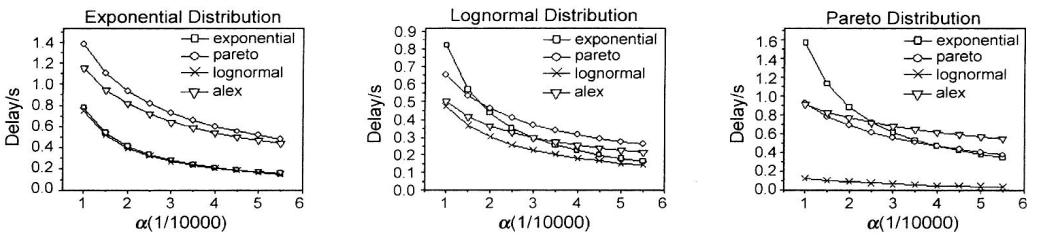
为了更好的衡量本算法的性能, 共做了三组实验, 第一组实验假设文档的修改时间间隔为负指数分布, 参数 $\lambda = 86400$, 第二组实验假设文档的修改时间间隔为对数正态分布, 并在 1 小时(3600 秒)时修改的概率最大, 故 $\mu = \ln(3600)$, 由公式 $E(x) = \exp(\mu + \sigma^2/2)$ 可知 $\sigma = \sqrt{2 \times (\ln(86400) - \mu)}$, 第三组实验假设文档的修改时间间隔为 Pareto 分布, 其中 $b = 3600$, $a = 24/23$.

对每一组实验, 都用负指数分布、对数正态分布、Pareto 分布以及 Alex 协议进行测试, 以比较各自的性能关系. 在每一组实验中, 为了得到万分之一左右的文档过期率, 门限值 α 取值从 0.00010 到 0.00055 共十个数值, 每次递增 0.00005.

4.2 仿真结果及分析

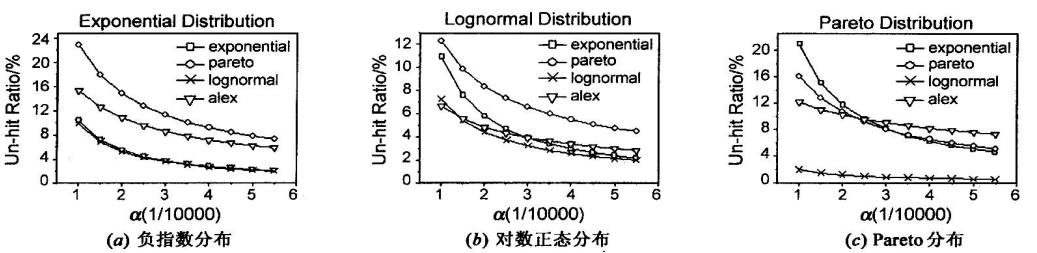
图 4 是三种分布下四种算法分别得到的客户时延, 意味着当用户拨打语音平台后, 到语音平台获取第一个 VoiceXML 文档之间的时间间隔, 显然此时间越少越好. 从图中可知, 客户的时延在 1.4 秒内, 而根据仿真的设置, 不采用缓存时客户时延将为 3 秒. 时延随着门限值的 α 增大而减少.

为了更好考察缓存与服务器通信的情况, 图 5 表示了在三种分布下四种算法的未命中率, 即缓存需要访问服务器的次数与总的用户访问缓存次数之比. 可以看到, 只有 20% 左右的访问需要传送到服务器, 80% 左右的访问可以使用缓存中的内容, 相对不采用缓存而言, 性能有很大的提高.



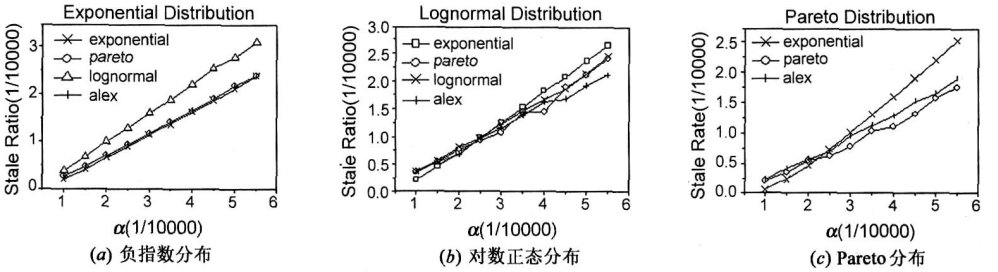
(a) 负指数分布 (b) 对数正态分布 (c) Pareto 分布

图 4 三种分布下的时延



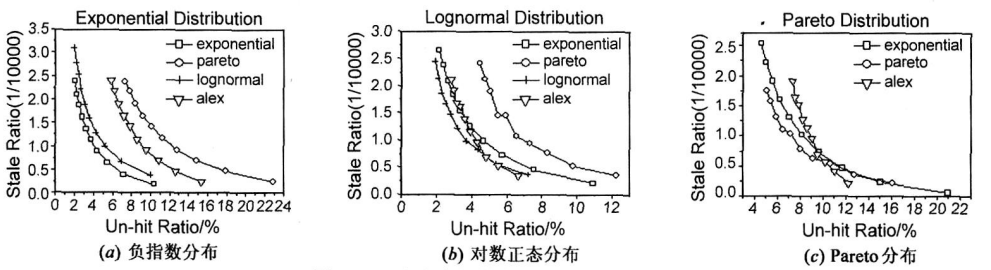
(a) 负指数分布 (b) 对数正态分布 (c) Pareto 分布

图 5 三种分布下的未命中率



(a) 负指数分布　(b) 对数正态分布　(c) Pareto 分布

图6　三种分布下的文档过期率



(a) 负指数分布　(b) 对数正态分布　(c) Pareto 分布

图7　三种分布下的未命中率与文档过期率

对比图 4 可知, 当未命中率较大时, 对应的客户时延也较大, 基本成正比关系, 因为未命中时缓存需要访问服务器, 从而导致客户时延增加. 同样, 服务器与缓存之间的通信量, 服务器的磁盘读写次数, CPU 使用率, 内存使用率也与未命中率成类似正比的关系. 这里不再一一列举其数据.

图6 是四种算法分别在三种分布下的文档过期率, 表示用户得到的过期文档与用户访问缓存的次数之比. 其中图 5*c* 中, 因为用对数正态分布拟合 Pareto 分布所导致的文档过期率达到了 88% 左右, 显然不能满足语音呼叫的需求, 故将其忽略, 只画出其他三种算法的结果.

由图 6 可知, 文档的过期率基本与门限值 α 成正比. 在门限值为 0.0002 左右时, 拟合分布算法能得到小于 0.0001 的文档过期率. 当文档的修改时间间隔满足负指数分布时, 对数正态分布的文档过期率要大于其他三种算法.

好的算法不仅要得到较小的文档过期率, 同时也应该得到较小的时延, 即应该得到较小的未命中率. 从图 4 到图 6 来看, 要评估四种算法的优劣非常困难. 根据文献[6]的结论, 缓存的一致性与其带来的性能提高之间存在类似反比的关系. 故建立一个新的坐标系, 令 X 轴为其性能而 Y 轴为其一致性, 可知某一算法的曲线更加靠近坐标则表示其算法性能较好. 反之, 距离坐标越远的曲线表示其算法性能越差.

根据本文实验的结果, 客户时延, 服务器与缓存之间的通信量, 服务器的磁盘读写次数, CPU 使用率, 内存使用率都与缓存未命中率成类似正比的关系. 故, 令文档未命中率代表缓存带来的性能的提高, 文档过期率代表缓存的一致性, 可以得到图 7.

如图 7(*a*) 所示, 当文档修改时间间隔满足负指数分布时, 采用负指数分布拟合预测的结果距离坐标轴最近; 而在图 7(*b*) 中, 文档的修改时间间隔满足对数正态分布, 所以采用对数正态分布拟合预测的效果最佳; 图 7(*c*) 中, 文档的修改时间满足 Pareto 分布, 故采用 Pareto 分布拟合预测能得到最好的结果. 显然, 满足显然当文档的修改时间间隔满足某一随机分布时, 采用同一分布拟合将取得最优的文档过期率及性能, 明显超过采用其他的分布拟合预测及 Alex 算法. 而图 7(*b*)、图 7(*c*) 中, 预测拟合算法并没有得到如图 7(*a*) 中的完美效果, 是因为我们没有具体计算 T_m 的最优值, 而是为了简单起见直接使用了 $T_m = T_{\text{return}} - \Delta T$, 而对数正态分布和 Pareto 分布与负指数分布不同, 前两者都没有无后效性的特性.

5　结束语及未来工作

本文讨论了在 Web 环境下, 在不对服务器进行任何修改的前提下, 如何实现 VoiceXML 文档缓存一致性的问题, 考虑到 VoiceXML 文档的特殊性, 根据 Web 文档的修改时间间隔满足负指数分布这一结论[14, 15], 提出使用参数估计的方法来拟合 VoiceXML 文档的随机分布, 再通过此分布预测 VoiceXML 文档改变的概率, 用以指导是否应访问服务器. 仿真研究表明: (1) 与 Alex 算法相比, 使用拟合分布预测算法能在得到较小文档过期率的同时得到较大的性能提高. (2) 在门限值 α 在 0.0002 左右时, 拟合分布算法能得到小于 0.0001 的文档过期率, 满足了语音平台低于万分之一呼叫出错率的要求. 未来的工作将是(1) 在不知道具体的 VoiceXML 文档修改时间间隔分布

的情况下,如何预测文档的修改概率,以达到最优的一致性与最优的性能提高.(2)考察实际中 VoiceXML 文档修改的分布具体为何种随机分布,便于根据该分布实现最优的拟合预测.

参考文献:

[1] Scott M, Daniel CB, Jerry C, et al. Voice Extensible Markup Language (VoiceXML) Version 2.0[Z]. W3C. http://www.w3.org/TR/voicexml20/. 2004.

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

[3] Cate V. Alex-a global filesystem[A]. Proceedings of the 1992 USENIX File System Workshop[C]. Ann Arbor: USENIX, 1992. 1- 12.

[4] James G, Margo S. World-wide web cache consistency[A]. Proceedings of the 1996 USENIX Technical conference[C]. San Diego: USENIX, 1996. 141- 151.

[5] Srinivasan R, Chao L, Ramamritham K. Maintaining temporal coherency of virtual data warehouses[A]. Real-Time Systems Symposium[C]. 1998. Madrid: IEEE, 1998. 60- 70.

[6] Satitsamitpong M, Thomas H. Cache freshness optimization: sliding scale guarantees[A]. Advanced Issues of E-Commerce and Web-Based Information Systems[C]. San Jose: WECWIS, 2001. 228- 230.

[7] Datta A, Dutta K, Thomas H, VanderMeer D, Ramamritham K, Suresha. A proxy-based approach for dynamic content acceleration on the WWW[A]. Advanced Issues of E-Commerce and Web-Based Information Systems [C]. Newport Beach: WECWIS, 2002. 159- 164.

[8] Yuan C, Hua Z G, Zhang Z. Proxy+: simple proxy augmentation for dynamic content processing[A]. Proceedings of Web Caching and Distribution[C]. Hawthorne: WCW, 2003. http://2003.iwcw.org/papers/yuan.pdf.

[9] Alonso R, Barbara D, Garcia-Molina H. Data caching issues in an information retrieval system[J]. ACM Transactions on Database Systems (TODS), 1990, 15(3): 359- 384.

[10] Cao P, Liu C J. Maintaining strong cache consistency in the world-wide web[J]. IEEE Transactions on Computers, 1998, 47(4): 445- 457.

[11] Wen SL, Po P, Wang P H, Canda K S, Agrawal D. Freshness-driven adaptive caching for dynamic content[A]. Proceeding of Database Systems for Advanced Applications, 2003[C]. Kyoto: DASFAA, 2003. 203- 212.

[12] Renu T, Thirumale N, Srikanth R. WCDP: A protocol for web cache consistency[A]. Proceedings of Web Caching and Distribution, 2002[C]. Boulder: WCW, 2002. http://2002.iwcw.org/papers/18500180.pdf.

[13] Cao L. Consistency Control Algorithms for Web Caching[Z]. http://db.uwaterloo.ca/~tozsu/courses/cs748t/weekly.html. 2001.

[14] Brian B, George C. How dynamic is the web? [A]. Proceedings of the Ninth International World Wide Web Conference[C]. IW3C2, 2000. http://www9.org/w9cdrom/264/264.html.

[15] Brian B, George C. Keeping up with the changing web[J]. IEEE Computer, 2000, 33(5): 52- 58.

[16] 盛骤,谢式千,潘承毅. 概率论与数理统计[M]. 第二版. 北京: 高等教育出版社, 1989. 157- 165.

作者简介:

王文林 男, 1979 年出生于广西全州, 2001 年在北京邮电大学获学士学位,现在北京邮电大学网络与交换技术国家重点实验室攻读博士学位,主要研究领域为多媒体通信、下一代网络增值业务.

E-mail: wang.wenlin@gmail.com

廖建新 男, 1965 年出生于四川,分别于 1985、1991、1996 年获电子科技大学学士、硕士、博士学位, 1996 年至 1998 年在北京邮电大学网络与交换技术国家重点实验室从事博士后研究,现为北京邮电大学教授,博士生导师.在国内外核心刊物及会议上已发表论文数百篇.主要研究方向是通信软件,增值业务提供技术.

E-mail: liaojx@bupt.edu.cn

朱晓民 男, 1974 年出生于浙江,博士,北京邮电大学副研究员,已在国内外学术期刊及会议上发表论文 80 余篇,主要研究方向为智能网、下一代业务网络、协议工程.个人主页 http://zhuxm.ik8.com