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

Jianxin Liao, Wenlin Wang, Xiaomin Zhu
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
Email: liaojianxin@ebupt.com

Abstract

Using cache in the voice platform based on VoiceXML (Voice eXtensible Makeup Language) could reduce the cost of network bandwidth, server-load and access delay. By 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 enhance the system performance.

1. Introduction

Voice eXtensible Makeup Language (VoiceXML) [1], which was established by VoiceXML forum, is a new computer language designed to make Internet content and information accessible via voice and phone. Its main goal is to bring the full power of web development and content delivery to voice response applications, and free the developers from low-level programming and resource management. It enables separation of the behavior and the logic of voice service, and makes the development easy and quickly.

Fig.1 shows one of the voice platform architecture based on VoiceXML. When a user accesses the platform via phone, an instance of VoiceXML interpreter will start up to retrieve a VoiceXML document from the Web server. According to the document, the interpreter dictates the implementation to interact with the user via voice, submits the user’s input and retrieves the next document. Obviously, the interval between user input his information and his hearing of the next voice, that is the user wait time, is composed of three parts: the time of starting up the interpreter instance, retrieving the document and executing the document. Among those parts, the second must be the longest and most uncontrollable.

Figure 1. Architecture of voice platform based on VoiceXML

The delay is more sensitivity when users access a server via voice than by a Web browser. Therefore, if users access server via voice, they will have a higher abandonment ratio because of the Quality of Service (QoS). So VoiceXML forum suggested: 1) integrate more dialogs in a single document as much as possible to reduce the interaction between the platform and Web server; 2) setting a cache server between the platform and Web server to reduce the network flow, the loads of server and the access delay.

However, the data in the server are renewed continually, and so are the VoiceXML documents. The consistency between the cache and the server must be considered to guarantee freshness of those cached documents. How to control this consistency will be discussed in this paper.

2. Related Work

There are basically two categories of cache consistency approaches: one is weak consistency mechanism, which concedes users to access some stale data by the best effort strategy; the second is strong consistency mechanism, which ensures the user to retrieve the latest data. The former is used when delay is more sensitive, while the later is used when data is more sensitive.

The main idea of weak consistency is that each document will be assigned a Time-to-Live (TTL) value by the client or server (via the “Expires” parameter of HTTP protocol) to estimate of its lifetime. When TTL expired, the document is regarded as stale and need to retrieve the latest version. Ref. [2] proposes the Alex protocol to assign a bigger TTL value to the documents
which are unchanged in a longer time. And ref. [3]
shows that the Alex protocol is the most effective
adapt TTL algorithm based on the change magnitude
of the data, but can not apply in the Web system for the
change magnitude of a document is immeasurable.
Ref. [5] supposes that the cache consistency may be in
inverse ratio with the enhancement of the performance,
and there is a balance between them. Ref. [6,7] divide
the document into many pieces, and maintain the
consistency of each piece independently.

In strong consistency mechanism, there are two
ways. The first, named Polling-Every-Time, is that
each request sent to the server with the “If-Modified-
Since” parameter set to the date of cached document.
The server will send the new document back with
status 200 if the cache is invalid. Otherwise, status 304
will be sent back. The second is invalidation algorithm.
The server monitors the documents continuously and
send messages to notify all of the clients to delete the
cached document once one document was modified.
Ref. [9] supposes the “two-tier-lease-augmented
invalidation” protocol, and proves it has the best
performance compared with TTL and Polling-Every-
Time. Ref. [10] proposes the server can know whether
the pages are modified or not by monitoring the log of
Distribution protocol (WCDP).

Ref. [12] analyzes the applications of the
consistency control mechanisms in the web
environment, and points out their shortcomings: TTL
can not avoid stale documents; Polling-Every-Time
can reduce some unnecessary file transfer, but can not
reduce the client delay; the main problem of the
Invalidation algorithm is that the server must store the
information of all clients and keep communications
with these clients, its another problem lies in its
difficulty to deal with failures.

3. Fitting & Predicting Algorithm

The goal of this paper is to develop a method that
does not entail any server modifications or changes to
the HTTP protocol, and then reduces the client delay as
possible as we can while meets the demand of voice
call. Therefore, the Invalidation mechanism is not
appropriate for its changes to the server and HTTP
protocol. Polling-Every-Time can not reduce the client
delay. Considering one ten-thousandth call loss
probability (CLP) is allowed in the voice call, we
suppose that the same probability of stale documents is
allowed. Therefore, we can maintain the consistency
between the cache and the server, and reduce the client
wait time by TTL.

3.1. Algorithm Model

Ref. [13, 14] inspects the changes of 100,000 Web
pages and draws a conclusion that the modification
intervals of the page are approximate to an Exponential
distribution. So we suppose that the intervals of
VoiceXML document changes should be approximate
to a stochastic distribution marked as \( f(x) \).

According to this distribution and the modification
history data of document, it is easy to fit the
parameters and predict the change probability of the
VoiceXML document. The document in the cache is
regarded as freshness if the change probability is less
than the predefined limitation \( \alpha \), otherwise the cached
document is stale and should be retrieved from the server.

In general, parameter fitting uses point estimation
method. There are basically two methods, one is
Moment Estimation (ME) and the other is Maximum
Likelihood Estimation (MLE).

For Exponential distribution \( F(x) = 1 - \exp(-\lambda x) \), the
result of ME and MLE is \( \lambda = \frac{1}{n} \sum x_i \) equally.

For Lognormal distribution, it is difficult to obtain
the result of ME. But according to its definition, when
\( x \) is Lognormal distribution, \( \ln x \) must be a Normal
distribution with the same parameters. Thus, for \( \ln x \),
the result of ME is \( \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \ln x_i, \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (\ln x_i)^2 - \hat{\mu}^2 \), same as the result of MLE.

For Pareto distribution \( F(x) = 1 - \left( \frac{b}{x} \right)^a \) (\( b > 0, a > 0 \)), when \( a > 1 \), its mathematical expectation exists. And when \( a > 2 \), its variance exists. In general, the mean of the intervals of the document changes always exists, so \( a \) must bigger than \( 1 \) while \( a > 2 \) is uncertain. Therefore, it is impossible to calculate ME. For MLE, there is:

\[
L = \prod_{i=1}^{n} \left( \frac{ab^x}{x^{a+1}} \right) = \ln L = \sum_{i=1}^{n} \left( \ln a + a \ln b - (a + 1) \ln x_i \right)
\]

If \( \frac{\partial L}{\partial b} = 0 \), it results \( \frac{a}{b} = 0 \). Obviously, it conflicts with the definition of Pareto distribution. So it is also impossible to calculate MLE. Two theorems will be proved as follows to get the estimator of Pareto distribution.

**Theorem 1.** In Pareto distribution \( F(x) = 1 - \left( \frac{b}{x} \right)^a \) (\( b > 0, a > 0 \)), the estimator \( \hat{\mu} = \frac{n \bar{X} - C}{n(\bar{X} - C)} \), \( \hat{b} = \frac{(n-1)\bar{X}C}{n\bar{X} - C} \) is unbiased. Here, \( \bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i \), \( C = \min(X_i) \).

**Proof.** The distribution of \( C \) must be:

\[
F_C(x) = 1 - \prod_{i=1}^{n} (1 - F_i(x)) = 1 - \left( \frac{b}{x} \right)^a
\]

Thus \( E(C) = \frac{nab}{na-1} \). And \( E(\bar{X}) = E(X) = \frac{ab}{a-1} \). So:

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

\[
E(\hat{b}) = \frac{(n-1)\bar{X}E(X)E(C)}{nE(\bar{X}) - E(C)} = \frac{(n-1)\bar{X}E(X)E(C)}{na - 1 - na - 1} = b
\]

Theorem 1 is proved.

**Theorem 2.** In Pareto distribution \( F(x) = 1 - \left( \frac{b}{x} \right)^a \) (\( b > 0, a > 0 \)), the estimator \( \hat{\mu} = \frac{n \bar{X} - C}{n(\bar{X} - C)} \), \( \hat{b} = \frac{(n-1)\bar{X}C}{n\bar{X} - C} \) is consistency. Here, \( \bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i \), \( C = \min(X_i) \).

**Proof.** From the proof of theorem 1, we know \( \lim_{n \to \infty} \bar{X} = \frac{ab}{a-1}, \lim_{n \to \infty} C = \frac{ab}{na - 1} = b \). So:

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

\[
\lim_{n \to \infty} \hat{b} = \lim_{n \to \infty} \frac{(n-1)\bar{X}C}{n \bar{X} - C} = \lim_{n \to \infty} \frac{n \bar{X} - C}{n \bar{X} - C} = b
\]

Theorem 2 is proved.

According to theorem 1 and 2, the estimator \( \hat{\mu} = \frac{n \bar{X} - C}{n(\bar{X} - C)} \), \( \hat{b} = \frac{(n-1)\bar{X}C}{n\bar{X} - C} \) can be used to fit the parameters of Pareto distribution.

### 3.3. Predict Algorithm

We suppose that the cached document modify time is \( T_{doc} \) and the half of time-to-round (TTR) from the cache to the server is \( \Delta T \) (here, it is supposed the time from the cache to the server is equal to the time from the server to the cache). It is assumed that when a user accesses the cache at the time \( T \), he may want the document available in the time \( T + \Delta T \). So, the server will estimate the change probability in \( (T_{doc}, T + \Delta T) \):

\[
P(X < T + \Delta T - T_{doc}) = F(T + \Delta T - T_{doc})
\]

The cached document will be used if \( P(X < T + \Delta T - T_{doc}) < \alpha \), otherwise the request with “If-Modified-Since” parameter will be sent to the server. The validity of cached document will lie on the status code (200 or 304) returned by the server.

But formula (1) gets a very low efficiency. If a user accesses the cache at time \( T_1 \) and \( P(X < T_1 + \Delta T - T_{doc}) \geq \alpha \), this request will be sent to the server. If the document is not modified, status code 304 will be returned. So the cached document can be used. And then, if the cache is accessed at the time \( T_2 (T_2 > T_1) \), \( P(X < T_2 + \Delta T - T_{doc}) > P(X < T_1 + \Delta T - T_{doc}) \geq \alpha \), which means this request will be sent to the server again, and can not profit from the access at the time \( T_1 \).

Thus, formula (1) should be improved. There are two scenes listed below:

1. \( T_2 - T_1 \geq 2\Delta T \)

![Figure 3. User Access Sequence](image-url)
Let $T_m$ be the VoiceXML document modification time, so $T_m \in (T_{doc}, T_{fresh} + \Delta T)$. Then, the cache should be refreshed and the value of $T_m$ should be assigned to $T_{doc}$. So, it can be ensured that the document is unchanged in $(T_{doc}, T_{fresh} + \Delta T)$.

Therefore, the document change probability is:

$$P[X < T + \Delta T - T_{doc} \mid X > T_{fresh} + \Delta T - T_{doc}]$$ \hspace{1cm} (2)

2. $T_2-T_1<2\Delta T$

In the VoiceXML applications, the user’s access intervals are very short, and usually less than the RTT. In this scene as Fig.3-b, if we are waiting for the result of the last access instead of accessing the server again, we will get the latest document at time $T_{fresh} + \Delta T$ instead of $T_2 + \Delta T$, and which will result in the uncontrollable document stale.

Let $T_{return}$ be the time when the cache received the last information about this document. There will be one of the two results for the access at the time $T_1$:

Firstly, the returned status code is 304. It means the document has been unchanged in $(T_{doc}, T_{fresh} + \Delta T)$. So in $(T_{fresh} + \Delta T, T_2 + \Delta T)$, the change probability of the document is $P[X < T_2 + \Delta T - T_{doc} \mid X > T_{fresh} + \Delta T - T_{doc}]$.

Secondly, if the returned status code is 200, the document has been changed in $(T_{return} - \Delta T, T_{fresh} + \Delta T)$. But we can not obtain the actual modification time now. If let $T_m \in (T_{return} - \Delta T, T_{fresh} + \Delta T)$ be the VoiceXML document modification time, the change probability of the document will be $P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m]$. We will guess the best value of $T_m$ below.

Because the stale ratio is the main goal of this algorithm, we should enhance the probability to access the server. Therefore, if we want to get a best value of $T_m$, we should make the value of $P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m]$ be the maximum.

The derivative of $P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m]$ is:

$$\left( P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m] \right)'$$

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

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

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

Here, $f(x)$ is the probability density function. According to the character of the probability, we can know $F(x) \leq 1$. Therefore, let’s set formula (3) be zero, we can get the value of $T_m$ when $P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m]$ reaches its extreme value. So by judging whether the extreme value is maximum value, and according $T_m \in (T_{doc}, T_{fresh} + \Delta T)$, we will get the best value of $T_m$.

Specially, if the distribution is an Exponential, formula (3) is always zero. Thus, the probability $P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m]$ is irrelative to $T_m$. This is the memoryless property of exponential.

According to the above, when $T_2 - T_1 < 2\Delta T$, if the change probability of the document is $p$ at the time $T_1$, the change probability of the document at the time $T_2$ should be less than:

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

$$= P[X < T_2 + \Delta T - T_m \mid X > T_{fresh} + \Delta T - T_m]$$

Here, $T_m \in (T_{return} - \Delta T, T_{fresh} + \Delta T)$ is the best modified time. In fact, we can simply assume $T_m = T_{return} - \Delta T$. The formula (4) is changed to:

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

(5)

4. Performance

In the Fitting & Predicting Algorithm, the main goal is how to enhance the performance of the system while ensure the consistency of the cache as possible as we can. Here, the performance of this algorithm is inspected compared with the Alex protocol.

It is assumed that: (1) the user’s access to services follows a Poisson distribution and its mean arrival ratio is 1 user per second; (2) the delay from the voice platform to the cache is zero for the cache module is in the voice platform; (3) the RTT from the cache to the server is 3 second, that is $\Delta T = 1.5s$; (4) the mean changed ratio of VoiceXML documents is 1 day.

Three groups of experiment are carried out to measure the performance of this algorithm. In the first group, we assume the document modification intervals accord to the Exponential distribution with parameter $\lambda = 86400$. In the second, the document modification intervals follow a Lognormal distribution, and the changed probability reaches its maximum at the time 1 hour (3600s). So the parameter $\mu = \ln(3600)$. According to the formula $E(x) = \exp(\mu + \sigma^2/2)$, we know that the parameter $\sigma = \sqrt{\lambda \times (\ln(36400) - \mu)}$. In the third group, the intervals of document changes follow the Pareto distribution with the parameters $b = 3600$, $a = 24/23$.

Each group is tested with Exponential distribution, Lognormal distribution, Pareto distribution and the Alex protocol. The limitation $\alpha$ is assigned ten values from 0.00010 to 0.00055 to ensure the stale ratio of the document is around one ten-thousandth.
Fig. 4 shows the client delays of the four algorithms by the three distributions. From Fig. 4, we know that the client delay is lower than 1.4s while the delay is 3s without cache module. Therefore, the delay halved.

To inspect the flow between the cache and the server, Fig. 5 shows the un-hit ratio of the four algorithms, that is, the ratio of the times of accessing the server and the total times of accessing the cache. It shows that only about 20% accesses are needed to transfer to the server. Compared with Fig. 4, we know that the un-hit ratio increases as the client delay increases, and the client delay is in direct ratio with the un-hit. Similarly, the flow between the cache and the server, the server’s CPU usage and the memory usage are all in direct ratio with the un-hit ratio.

Fig. 6 shows the stale ratio of the four algorithms, that is, the ratio of the times of user obtained stale document and the total times of user accessed the
cache. Here, in Fig. 6(c), the stale ratio of lognormal is bigger than 88%, so it has been ignored. We can learn from Fig. 6 that the stale ratio is in direct ratio with the limitation α.

We known that if an algorithm is better than another, its stale ratio must be less, so does its delay. But, from Fig. 4, 5, 6, it is too difficult to evaluate those algorithms.

According the conclusion of Ref. [5], the cache consistency is in inverse ratio with the enhancement of the performance, we can evaluate those algorithms by setting a new coordinates whose x-axis represents the performance and y-axis represents the consistency. Therefore, if the curve of the algorithm is closer to the axis, the algorithm is better.

As shown in Fig. 7, Fitting & Predicting by the same distribution is the best algorithm and surpasses the Alex protocol dramatically for its curve is closer to the axis than the Alex protocols’ in main time.

5. Conclusion and Future Work

How to maintain the consistency of the cached VoiceXML document without any server modifications in the Web environment is discussed in this paper. Considering the special character of VoiceXML document and the conclusion of Ref. [13, 14], the Fitting & Predicting algorithm is proposed, which states the stochastic distribution parameters of the document modification intervals, and predicts the document change probability to estimate the validity of the cached document. The simulation results show: (1) compared with the Alex protocol, the algorithm can obtain a less stale ratio and a higher performance; (2) when the limitation is around 0.0002, the stale ratio is less than 0.0001, which meets the demand of the voice call.

The future work is: (1) to inspect how to predict the chance probability when the distribution of the document change’s interval is unknown; (2) to inspect which distribution does the intervals indeed according to so that we can perform the best prediction.

6. Acknowledgement

This work is jointly supported by: (1) National Science Fund for Distinguished Young Scholars (No. 60525110); (2) Program for New Century Excellent Talents in University (No. NCET-04-0111); (3) Specialized Research Fund for the Doctoral Program of Higher Education (No. 20030013006); (4) Development Fund Project for Electronic and Information Industry (Mobile Service and Application System Based on 3G); (5) Development Fund Key Project for Electronic and Information Industry (Core Service Platform for Next Generation Network).

7. References