Title: HIGH VOLUME E-MAIL ASSEMBLY AND DELIVERY

Abstract: A high volume E-mail message assembly and delivery system uses a high level of parallelism and data flow control to enhance E-mail assembly and delivery. E-mail messages are assembled by first creating E-mail templates that contain links to content data, such as news or financial information. When the content data is finalized, the system is able to quickly insert it into the templates to form completely assembled E-mail messages. As the E-mails are assembled, they are forwarded to delivery processors for immediate delivery. If the delivery processors become overwhelmed, they instruct the processors assembling the E-mails to slow down.
HIGH VOLUME E-MAIL ASSEMBLY AND DELIVERY

Field of the Invention

This present invention relates to electronic mail, and more particularly, to high volume electronic mail delivery systems and methods.

Description of the Related Art

Electronic mail ("E-mail") has matured into a widely accepted standard for business and personal communication.

Commercial services exist with the goal of using E-mail as an information vehicle by sending personalized electronic E-mail. The services use computers to generate articles/stories that match a predefined profile of a particular subscriber. These services all have one thing in common--they manipulate information (e.g., AP wire service, Reuters, Business Wire) and match this information to a customer profile. The resulting personalized E-mail is then delivered.

Traditionally, E-mail transmission has been handled by a single sending program operating on a single queue of messages. That is, personalized E-mails are assembled and then lined up in a first-in-first-out (FIFO) queue, and the queue is processed by a transmitting program. Although this method works well for low-volume E-mail transmission, for sites that transmit thousands, or even hundreds of thousands of E-mail messages a day, this method is slow because bottlenecks may occur both in the assembly of the individual personalized E-mails and in their transmission.

One attempt to solve the traditional problems associated with conventional E-mail delivery is described in U.S. patent application serial number 08/710,964, filed on
September 24, 1996, titled "Method and Apparatus for High Volume E-mail Delivery," by the present assignee. This patent discloses the use of multiple, parallel mail delivery queues to transmit E-mail. By using multiple queues, the total number of delivered E-mails can be significantly increased. Although using multiple parallel queues may improve the rate of E-mail transmission, the maximum delivery rate is still limited by the speed with which E-mails are assembled and forwarded to the delivery queues, and by the degree to which the parallel queues are effectively used.

It is therefore an object of the present invention to provide efficient, high volume personalized E-mail assembly and transmission systems and methods.

Summary of the Invention

One system consistent with the present invention includes a database, a template processor, and at least one build processor. The template processor generates E-mail templates based on customer profile information stored in the database, the E-mail templates including data corresponding to information contained in the E-mail, and fields referencing content data to be inserted in the E-mail. The build processor includes computer instructions which, when executed by the build processor, perform the steps of: receiving the templates generated by the template processor, retrieving content information referenced by the fields in the templates from
the database, and replacing the fields in the templates with the retrieved content information to thereby create assembled E-mail messages.

A method of assembling an E-mail message consistent with the present invention includes retrieving a template generated based on a customer profile, the template including data for the assembled version of the E-mail and fields referencing variable content data; retrieving the content information referenced by the fields; and completing the assembly of the E-mail message by replacing the fields with the referenced content data.

Another system consistent with the present invention comprises a database storing content information; a first group of processors for inserting content information from the database into pre-generated E-mail templates to obtain complete E-mail messages, the first group of processors retrieving the content information from the database based on fields in the E-mail templates; and a second group of processors for receiving the complete E-mail messages from the first group of processors and transmitting the complete E-mail messages to their recipients using a plurality of mailing programs.

Yet another system consistent with the present invention comprises a computer network for assembling and delivering E-mail. The computer network includes means for storing content data to be included in the delivered E-mail; means for generating E-mail templates, each template including data corresponding to the assembled version of the E-mail and fields referencing the stored content data; means for retrieving the content data referenced by the fields and completing the assembly of the E-mail
message by replacing the fields with the referenced content data; and means for delivering the assembled E-mail over a network.

**Brief Description Of the Drawings**

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments consistent with this invention and, together with the description, help explain the principles of the invention. In the drawings,

Fig. 1 is an exemplary diagram of an E-mail assembly and delivery system consistent with the present invention;

Fig. 2 is a diagram illustrating an exemplary customer E-mail profile;

Fig. 3 is a diagram illustrating an E-mail template;

Fig. 4 is a diagram illustrating an assembled E-mail; and

Fig. 5 is a functional diagram of a simplified version of the E-mail system.

**Detailed Description**

Reference will now be made in detail to the construction and operation of preferred implementations of the present invention which are illustrated in the accompanying drawings. In the drawings, like elements and operations are designated with the same reference numbers where possible.
Fig. 1 is a diagram illustrating high-volume E-mail delivery system 100. The system includes a database 10, build processors 30, QMUX delivery processors 40, mail delivery programs 50-51, and a network 60.

Personalized E-mails are derived from database 10, which stores E-mail content information (e.g., news, weather, financial information) and customer profile information. The profile information is preferably information relating to the individual preferences of each customer. The customer’s personalized E-mails are derived from the profile information.

Although database 10 is illustrated as a single database, database 10 is not necessarily implemented as a single “physical” database, but may be a group of multiple databases and related hardware and software.

Fig. 2 is a diagram illustrating an exemplary profile 200 for the user "John Doe." As shown, profile 200 includes the E-mail address 201 of John Doe and information fields, labeled as fields 202-204, representing the information services selected by the user. For example, field 202 indicates that John Doe would like to receive a daily weather report at 9pm for the area code 22304. Profile 200 also indicates that John Doe would like to receive finance related new alerts (field 203) and closing prices for selected securities (field 204).

Typically, customer profile information 200 is known well in advance of when the E-mail is scheduled to be sent. What is not known in advance is the content information to include in the E-mail. In fact, it is desirable to wait as long as possible until the scheduled E-mail delivery time before finalizing the content information. This
gives the content editors as long as possible to create the content. In the case of news, for example, the news stories can be as fresh as possible when they are sent.

Fig. 3 illustrates an exemplary E-mail template 300. Different templates exist for whether the E-mail to be sent is text-only or includes both graphics and text. A format such as html can be used for E-mails that are to contain both graphics and text.

Further, templates permit the use of a technique known as co-branding whereby E-mails can originate from the same source, and include the same content information, but be in a different format. For example, both company A and company B want to provide as a service to its customers the closing stock prices for the stocks its customers own. These E-mails can all originate from the same source, but with different formats according to which company is sponsoring the E-mail. Thus, E-mails sponsored by Company A will be in a format specific to Company A, while E-mails sponsored by Company B will be in a different format. As such, a template can be created for E-mails that are sponsored by Company A, and a different template for E-mails sponsored by Company B.

Template 300 includes E-mail header fields 301-303, advertisement field 304, and E-mail body 305. Field 301 is the address of the intended recipient, field 302 is the address of the sender (i.e., the address of the entity associated with system 100), and field 303 is the subject. Field 304 contains advertising information, such as a specific advertisement for a company sponsoring the E-mail. Body 305 contains the information requested by the user. Depending on whether the format of the E-mail is text or html, the body field can include text-only or both graphics and text.
The pre-built templates, such as template 300, are stored on processors 30. Processors 30 are arranged in a network and act in parallel to finish assembling the E-mail. When the content information is finalized in database 10, processors 30 access content and profile information, based on fields 310-305, and integrate the content and profile information with the templates 300, by inserting the respective information into fields 301-305.

For example, for the profile of customer John Doe illustrated in figure 2 that includes the closing security prices for IBM and MSFT, a processor 30 accesses the database to determine the E-mail addresses to insert into fields 301 and 302. The processor further accesses the database to determine the Subject to insert into field 303, which in this case is "Finance-early closing bell." The processor 30 also accesses the database for the Advertisement to insert in field 304. This advertisement may include both graphics and text if the template is an html template. For this profile, the processor 30 determines the closing security prices for IBM and MSFT, and then places them in a table, which is a text table for a text template, or an HTML table if the E-mail uses an html template. Processors 30 may cache the content information accessed from database 10 to reduce the need to access database 10 multiple times for the same content.

Fig. 4 is an illustration of template 300 after the fields have been replaced by the content information from database 10. At this point, template 300 is a fully assembled E-mail, labeled as E-mail 400.
Build processors 30 each continually integrate the pre-stored template information with the new content information (i.e., the variable portion of template 300 referenced by fields 305) to form assembled E-mails. As the E-mails are formed, build processors 30 forward the E-mails to one of Qmux processors 40 via a TCP/IP connection. Qmux processors 40, like processors 30, are a network of computers. Qmux processors 40 transmit the completed E-mails to an outside network 60, such as the Internet, using a plurality of mail delivery programs 50 running on the Qmux processors.

Mail delivery programs 50 run in parallel on the Qmux processors, allowing the E-mails to be transmitted in parallel to network 60. This allows for high total E-mail output. Preferably, mail delivery programs 50 are each a program such as "Sendmail," which is well known in the art and is widely used for sending E-mail. Standard "Sendmail" programs, when receiving E-mail messages to send, initially queue the E-mail messages to disk in a first-in-first-out (FIFO) queue. Consistent with the present invention, mail delivery programs 50 may be a modified version of "Sendmail" so that E-mail messages, when initially received, instead of being queued to disk, are immediately processed by mail delivery programs 50. By not queuing received E-mails to disk, mail delivery programs 50 save time and increase system performance because they eliminate the need to access relatively slow disk drives. Build processors 30 and delivery processors 40 may similarly save time by keeping all processes in random access memory (i.e., solid state memory) to avoid hard disk access.
Occasionally, one of mail delivery programs 50 may have difficulty delivering its E-mail to its destination. For example, the destination host may be temporarily non-responsive. When this happens, the mail delivery program terminates transmission of the E-mail and forwards it to a “slow” mail delivery program 51, which acts as a secondary queue for transmitting troublesome E-mail. In this manner, slow E-mails do not unnecessarily bog down the system.

As shown in Fig. 1, build processors 30 and Qmux processors 40 are coupled together through a TCP/IP connection. Through this connection, build processors 30 transmit E-mail messages to Qmux processors 40. Additionally, Qmux processors 40 provide feedback to build processors 30 so that if Qmux processors 40 begin to fall behind in handling E-mails from processors 30, Qmux processors 40 instruct processors 30 to throttle back the rate at which E-mails are assembled and forwarded to Qmux processors 40. Similarly, Qmux processors 40 monitor the status of E-mail delivery at each of mail delivery programs 50 and distribute the assembled E-mails to mail delivery programs 50 as each becomes ready to receive a new E-mail message.

Build processors 30 and Qmux processors 40 may be implemented in a UNIX based environment, although many other types of computer operating systems or hardware configurations may be used.

As described above, using templates in the build process increases the parallelism and efficiency of the mail delivery system. Further, by providing feedback from Qmux processors 40 to build processors 30 or from mail delivery programs 50 to Qmux processors 40, the system can efficiently control the flow of E-mail through the
various stages of the mail delivery process. Flow control of the E-mail is important to the overall efficiency of the system because by limiting the number of E-mails currently in process, the overhead associated with switching among multiple parallel processes can be held at an optimal level and lockup due to too many E-mails at a particular point in the system can be eliminated.

Fig. 5 is a diagram of a simplified version of E-mail system 100 illustrating operation of certain functional elements of the system. Database 10, build processors 30, Qmux processors 40, mail delivery programs 50, and network 60 are identical to the systems shown in Fig. 1. In addition, message transfer agents ("MTA") 70 are shown connected to network 60. Each MTA 70 corresponds to a particular delivery domain. For example, one of MTA's 70 may correspond to the domain for "aol.com." MTAs 70 accept and process E-mails transmitted from mail delivery programs 50.

MTAs 70 and Sendmail programs 50 preferably communicate with one another using the Simple Mail Transmission Protocol ("SMTP"). SMTP provides for a status signal, called an acknowledgment "dot," transmitted from MTAs 70 back to mail delivery programs 50. When an acknowledgment dot reaches its mail delivery program, the mail delivery program forwards the acknowledgment dot back to Qmux processors 40. Qmux processors 40 similarly transmit the acknowledgment dot upstream to build processors 30 and finally to database 10. If the E-mail was successfully transmitted, it is registered as such in database 10. Other status messages such as a message delivery failure may also be carried by the
acknowledgment dot. The path of the acknowledgment dot is illustrated in Fig. 5 by transmission path 502.

As previously mentioned, undeliverable E-mail messages may be transferred to slow mail delivery program 51. However, if the E-mail message fails delivery too many times, as indicated by the acknowledgment dots, system 100 may either cancel delivery altogether, or attempt to deliver the message with an alternate mechanism. For example, if the message is important, the message may then be delivered via facsimile transmission.

In addition to using an alternate delivery mechanism, the status information stored in database 10 from the acknowledgment dots may be used for a number of other E-mail "policy" decisions. For example, if all message transmissions from a particular domain are failing, system 100 may temporarily suspend building new E-mail messages intended for that domain. Policy based decisions may even be communicated across multiple builds, such as between an evening news transmission and an evening stock market update. Other domain based policy decisions are possible, such as, for example, different E-mail failure timeout periods per domain.

Each E-mail working its way through system 100 is preferably run in a separate program thread, such as one of threads 506-508. Threads 506-508 run independently of one another in E-mail system 100. Typically, threads are opened when the build of an E-mail is started, and closed when the E-mail has been transmitted and a successful acknowledgment dot returned to database 10.
Too many active threads destined for a single destination MTA may overcome that MTA. Accordingly, system 100 can limit the number of active threads to any particular MTA. Further, a destination that has multiple MTAs may request that all messages are transmitted to a specified one of the multiple MTAs. In this situation, system 100 may instruct all threads to transmit to the specified MTA.

E-mail delivery is not always time critical. It may be important that the E-mails be delivered, but not that they be delivered immediately. By keying the E-mail build process to system load, system 100 may efficiently transmit non time-sensitive E-mails. Thus, messages from a low priority batch can be built and delivered only when system load (as measured, for example, by the number of active E-mail threads) is below a certain threshold.

It will be apparent to those skilled in the art that various modifications and variations can be made to the present invention without departing from the scope or spirit of the invention. For example, although the Qmux and Sendmail programs were described as independent programs, they may alternatively be implemented as a single process.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with the true scope and spirit of the invention being indicated by the following claims.
What Is Claimed:

1. A method of assembling an E-mail message comprising the steps of:
   retrieving a template generated based on a customer profile, the template
   including data for the assembled version of the E-mail and fields referencing variable
   content data;
   retrieving the content information referenced by the fields; and
   completing the assembly of the E-mail message by replacing the fields with the
   referenced content data.

2. The method of claim 1, wherein the retrieving step is performed by
   accessing portions of a database referenced by the fields.

3. The method of claim 1, wherein the retrieving step is performed by
   accessing the content data from cache memory.
4. A computer system comprising:
   a database; and
   at least one build processor, the build processor including
   means for storing E-mail templates based on customer profile
   information stored in the database, the E-mail templates including data corresponding
to information contained in the E-mail and fields referencing content data to be
inserted in the E-mail, and
   means for storing computer instructions for, when executed by the build
processor, performing the steps of:
   receiving the stored templates,
   retrieving content information referenced by the fields in the
templates from the database, and
   replacing the fields in the templates with the retrieved content
information to thereby create assembled E-mail messages.

5. The computer system of claim 4, further including
   at least one delivery processor, coupled to the build processor, for receiving
and delivering the assembled E-mail messages, the delivery processor instructing the
build processor to reduce the rate at which the build processor generates the
assembled E-mail messages.
6. The computer system of claim 5, wherein the delivery processor reduces the rate at which the build processor generates the assembled E-mail messages when the delivery processor falls behind in delivering the assembled E-mail messages.

7. The computer system of claim 5, wherein the delivery processor further comprises a plurality of mail delivery programs for delivering the assembled E-mail messages.

8. The computer system of claim 4, wherein a mail delivery program directly delivers input E-mail messages without storing the received E-mail messages in a disk queue.

9. A system for assembling and delivering E-mail messages comprising:
   a database storing content information;
   a first group of one or more processors for inserting content information from the database into E-mail templates to obtain complete E-mail messages, the first group of processors retrieving the content information from the database based on fields in the E-mail templates;
   a second group of one or more processors for receiving the complete E-mail messages from the first group of one or more processors and transmitting the complete E-mail messages to their recipients using one or more mailing programs.
10. The system of claim 9, wherein the second group of processors further includes means for informing the first group of processors to reduce the rate of sending the complete E-mail messages to the second group of processors when the E-mail load at the second group of processors is too high.

11. The system of claim 9, wherein each of the processors of the first group of processors includes means for storing E-mail templates based on a customer profile stored in the database.

12. The system of claim 9, further comprising a mailing program designated by the second group of processors as a slow mailing program, the slow mailing program delivering E-mail messages that are determined to be difficult to deliver.

13. A computer network for assembling and delivering E-mail comprising:
   means for storing content data to be included in the delivered E-mail;
   means for storing E-mail templates, each template including data corresponding to the assembled version of the E-mail and fields referencing the stored content data;
means for retrieving the content data referenced by the fields and completing
the assembly of the E-mail message by replacing the fields with the referenced
content data; and
means for delivering the assembled E-mail over a network.

14. A computer system comprising:
a database;
at least one build processor for retrieving customer information from the
database and assembling E-mail messages based on the retrieved customer
information; and
at least one delivery processor, coupled to the build processor, for delivering
the assembled E-mail messages, the delivery processor instructing the build
processor to reduce the rate at which the build processor generates the assembled E-
mail messages when the delivery processor falls behind in delivering the assembled
E-mail messages.

15. The computer system of claim 14, wherein the delivery processor includes
means for accepting an acknowledgment message indicating whether the delivered E-
mail messages were successfully delivered and forwarding the acknowledgment
message to the build processor.
16. The computer system of claim 15, wherein the build processor further includes means for storing the acknowledgment message received from the delivery processor in the database.

17. The computer system of claim 15, wherein the build processor refrains from retrieving select customer information from the database when system load is high.

18. The computer system of claim 14, wherein the build processor further includes means for storing E-mail templates based on customer profile information stored in the database, the E-mail templates including data corresponding to information contained in the E-mail and fields referencing content data to be inserted in the E-mail.

19. The computer system of claim 14, wherein the delivery processor further includes a mail delivery program for delivering the E-mail messages over a computer network.

20. The computer system of claim 19, wherein the delivery processor further includes a facsimile transmission device for delivering the E-mail messages when delivery over the computer network fails.
21. The computer system of claim 14, wherein the delivery processor modifies delivery parameters based on a domain of an address of the E-mail messages.

22. A computer system comprising:

a build processor for assembling an E-mail message intended for a user based on content information and on a profile of the user indicating the content information desired by the user; and

a delivery processor for receiving the assembled E-mail and routing the E-mail to a mail delivery program; wherein

the build processor, the delivery processor, and the mail delivery program process the E-mail in a program thread maintained in solid state memory of the processors.
FIG. 1
SUBSTITUTE SHEET (RULE 26)
FIG. 2

TO: JOHND@EMAIL.COM
FROM: FINANCE@INFOBEAT.COM
SUBJECT: FINANCE -- FULL CLOSING BELL

[ADVERTISEMENT]

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>CLOSE</th>
<th>CHANGE</th>
<th>HIGH</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSFT</td>
<td>131 9/16</td>
<td>-2 1/16</td>
<td>134 5/8</td>
<td>131 3/8</td>
</tr>
<tr>
<td>IBM</td>
<td>165</td>
<td>-4 1/2</td>
<td>170</td>
<td>163 3/4</td>
</tr>
</tbody>
</table>

FIG. 3

FIG. 4
FIG. 5

SUBSTITUTE SHEET (RULE 26)