Claims

1. An invalidation tracker system for tracking messages in a caching architecture of a pricing and shopping platform, wherein the caching architecture includes a number of levels, in which each level comprises one or more servers, in which invalidation messages are communicated from one level to another in order to send invalidation messages to all servers in the caching architecture, and wherein the invalidation tracker system receives data from provider databases to be communicated to the servers in the caching architecture; wherein the invalidation tracker system includes:

- a recording module for recording all invalidation messages communicated to the servers in the caching architecture to form a set of sent invalidation messages;
- an analysing module for determining the invalidation messages received at each server in the caching architecture and comparing this with the set of sent invalidation messages to identify one or more undelivered invalidation messages;
- a replay module for resending the one or more identified undelivered invalidation messages to an appropriate server in the caching architecture.

2. The system of claim 1, wherein the replay module resends invalidation messages based on a predetermined condition having been met.

3. The system of claim 2, wherein the predetermined condition comprises a plurality of recurring undelivered invalidation messages.

4. The system of claim 2 or claim 3, wherein the predetermined condition is triggered when the number of undelivered invalidation matched messages exceeds a predetermined threshold.

5. The system of any of claims 2 to 4, wherein the predetermined condition is identification of the server being non-operational.

6. The system of any preceding claim, further comprising a notification module for creating and communicating notifications of undelivered invalidation messages to a user.

7. The system of any preceding claim, further comprising a reporting module for creating reports and communicating said reports to a user.

8. The system of any preceding claim, wherein the invalidation tracker further comprises a supervisor module for controlling automatic retrieval of undelivered invalidation messages.

9. The system of any preceding claim, wherein the invalidation tracker system further comprises a configuration management module which stores information relating to the settings and topography of the caching architecture.

10. A method for tracking messages in a caching architecture of a pricing and shopping platform, wherein the caching architecture includes a number of levels, in which each level comprises one or more servers, in which invalidation messages are communicated from one level to another in order to send invalidation messages to all servers in the caching architecture, and wherein the messages are tracked by an invalidation tracker system which receives data from provider databases to be communicated to the servers in the caching architecture; wherein the method includes:

- recording via a computer all invalidation messages communicated to the servers in the caching architecture to form a set of sent invalidation messages;
- determining via a detector in a computer the invalidation messages received at each server in the caching architecture and comparing this with the set of sent invalidation messages to identify one or more undelivered invalidation messages;
- resending via a messaging system the one or more identified undelivered invalidation messages to an appropriate server in the caching architecture.

11. The method of claim 10, wherein replaying an invalidation message is based on a predetermined condition.

12. The method of claim 11, wherein the predetermined condition comprises identifying a plurality of recurring undelivered invalidation messages.

13. The method of claim 11 or 12, wherein the predetermined condition comprises triggering the replay of messages when the number of undelivered invalidation matched messages exceeds a predetermined threshold.

14. The method of any of claims 11 to 13, wherein the predetermined condition comprises identifying the server is non-operational.

15. The method of any of claims 10 to 14, further comprising the step of creating and communicating notifications of undelivered invalidation messages to a user.

16. The method of any of claims 10 to 15, further comprising the step of creating reports and communicating said reports to a user.

17. The method of any of claims 10 to 16, further comprising controlling automatic retrieval of undelivered invalidation messages via a supervisor module.

18. The method of any of claims 10 to 17, further comprising storing information relating to the setting and topography of the caching architecture via a configuration management module.

19. A computer program comprising instructions for carrying out the steps of the method according to any one of claims 10 to 18, when said computer program is executed on a computer system.

CLAIMS

1. An invalidation tracker system for tracking messages in a caching architecture of a pricing and shopping platform, wherein the caching architecture includes a number of levels, in which each level comprises one or more servers, in which invalidation messages are communicated from one level to another in order to send invalidation messages to all servers in the caching architecture, and wherein the invalidation tracker system receives data from provider databases to be communicated to the servers in the caching architecture; wherein the invalidation tracker system is distributed across the caching architecture and employs a central data console server, the invalidation tracker system including:

- a recording module arranged to record all invalidation messages communicated to the servers in the caching architecture to form a set of sent invalidation messages, the recording module being distributed across the servers and comprising, at the servers, invalidation message handlers, logging handlers and databases of compressed invalidation messages;

- a collecting module at the central data console server arranged to collect and record details of the received invalidation messages from the servers;

- an analysing module at the central data console server arranged to periodically determine the invalidation messages received at each server in the caching architecture and compare this with the set of sent invalidation messages to identify one or more undelivered invalidation messages;

- a replay module at the central data console server arranged to resend the one or more identified undelivered invalidation messages to an appropriate server in the caching architecture.

2. An invalidation tracker system according to claim 1, wherein the invalidation tracker system is arranged to store all recorded and replayed messages in a historic database.

3. An invalidation tracker system according to claim 1 or 2, further comprising a notification module arranged to create and communicate notifications of undelivered invalidation messages to a user.

4. An invalidation tracker system according to any one of claims 1 to 3, further comprising a reporting module arranged to create reports and communicate said reports to a user.

5. An invalidation tracker system according to any one of claims 1 to 4, wherein the invalidation tracker further comprises a supervisor module arranged to control automatic retrieval of undelivered invalidation messages.

6. An invalidation tracker system according to claim 5, wherein the supervisor module is arranged to generate a time range and a set of servers for which lost invalidation messages are to be detected and replayed.

7. An invalidation tracker system according to claim 6, wherein the supervisor module is arranged to send a check request to a detector module, the request indicating a source server, a list of target servers and the time range.

8. An invalidation tracker system according to claim 7, wherein the detector module is arranged to:

- check whether the source server is operational;
- request the source server to build a source invalidation message list for the time range;
- check whether the target servers indicated by the list of target servers are operational;
- request target invalidation message lists from the target servers, and to
- determine lost invalidation messages from the source invalidation message list and the target invalidation message lists.

9. An invalidation tracker system according to any one of claims 1 to 8, wherein the invalidation tracker system further comprises a configuration management module which stores information relating to the settings and topography of the caching architecture.

10. A method for tracking messages in a caching architecture of a pricing and shopping platform, wherein the caching architecture includes a number of levels, in which each level comprises one or more servers, in which invalidation messages are communicated from one level to another in order to send invalidation messages to all servers in the caching architecture, and wherein the messages are tracked by an invalidation tracker system which receives data from provider databases to be communicated to the servers in the caching

architecture, wherein the invalidation tracker system is distributed across the caching architecture and employs a central data console server, wherein the method includes:

- recording by a recording module all invalidation messages communicated to the servers in the caching architecture to form a set of sent invalidation messages, the recording module being distributed across the servers and comprising, at the servers, invalidation message handlers, logging handlers and databases of compressed invalidation messages;

- a collecting module at the central data console server collecting and recording details of the received invalidation messages from the servers;

- periodically determining via a detector by an analysing module at the central data console server the invalidation messages received at each server in the caching architecture and comparing this with the set of sent invalidation messages to identify one or more undelivered invalidation messages;

- resending via a a replay module at the central data console server the one or more identified undelivered invalidation messages to an appropriate server in the caching architecture.

11. A method according to claim 10, further comprising the step of storing all recorded and replayed messages in a historic database.

12. A method according to claim 10 or 11, further comprising the step of creating and communicating notifications of undelivered invalidation messages to a user.

13. A method according to any one of claims 10 to 12, further comprising the step of creating reports and communicating said reports to a user.

14. A method according to any one of claims 10 to 13, further comprising controlling automatic retrieval of undelivered invalidation messages via a supervisor module.

15. A method according to claim 14, further comprising supervisor module generating a time range and a set of servers for which lost invalidation messages are to be detected and replayed.

16. A method according to claim 15, further comprising the supervisor module sending a check request to a detector module, the request indicating a source server, a list of target servers and the time range.

17. A method according to claim 16, further comprising the detector module

- checking whether the source server is operational;
- requesting the source server to build a source invalidation message list for the time range;
- checking whether the target servers indicated by the list of target servers are operational;
- requesting target invalidation message lists from the target servers, and

- determining lost invalidation messages from the source invalidation message list and the target invalidation message lists.

18. A method according to any one of claims 10 to 17, further comprising storing information relating to the setting and topography of the caching architecture via a configuration management module.

19. A computer program comprising instructions for carrying out the steps of the method according to any one of claims 10 to 18, when said computer program is executed on a computer system.