CLAIMS

1. A system comprising:

at least one processor; and

at least one memory communicatively coupled to the at least one processor, the at least one memory having stored therein computer-executable components, comprising:

a cache framework that:

generates a data structure that identifies a plurality of caches in a cache tier order defining an order in which respective caches of the plurality of caches are to be accessed to service data requests;

in response to a change in volume of the data requests, modifies the data structure to add at least one additional cache to or remove at least one cache from the plurality of caches according to the change in volume of the data requests, and adjusts the cache tier order according to the addition or removal associated with the plurality of caches;

in response to receipt of a data request from a requesting entity for data, attempts to return the data in response to the request, including to read the data structure to access each cache according to the cache tier order until the data is obtained from a cache of the plurality of caches or return an indication that at least part of the data was not obtained from the plurality of caches, and

wherein at least a portion of the cache framework operates on a server of a front-end data service, and wherein the front-end data

service is coupled to communicate a request for the data to a back-end data service upon the return of the indication from the cache framework that at least part of the data was not obtained from the plurality of caches,

wherein the data request for data comprises a batch request for data items, and wherein the cache framework accesses a lowest cache of the plurality of caches to attempt to locate data for the data items, and for each data item not found, to access a higher cache of the plurality of caches to attempt to locate data for the data items until each data item is found or no higher cache remains to access, and

wherein the cache framework writes through data to any lower cache of the plurality of caches if the data is located in a higher cache of the plurality of caches relative to that lower cache.

2. The system of claim 1, wherein the cache tier order includes an in-memory cache as a lowest level cache, in which the in-memory cache is in a same machine as a server that includes the at least the portion of the cache framework.

3. The system of claim 1, wherein the cache tier order includes a distributed cache shared by a server that includes the at least the portion of the cache framework and at least one other server.

4. The system of claim 1, wherein the cache tier order includes an in-memory cache as a lowest level cache, in which the in-memory cache is in a same

machine as a server that includes the at least the portion of the cache framework, and a distributed cache shared by the server and at least one other server.

5. The system of claim 1, wherein the cache framework comprises an instantiated library.

6. The system of claim 1, wherein the change in volume is a decrease in the volume of the data requests, and the cache framework modifies the data structure to remove the at least one cache from the plurality of caches.

7. The system of claim 1, wherein the change in volume is an increase in the volume of the data requests, and cache framework modifies the data structure to add the at least one additional cache to the plurality of caches.

8. The system of claim 1, wherein at least another portion of the cache framework operates on a server of back-end data service, wherein the back-end data service is coupled to the front-end data service that is the requesting entity, and communicates a response to the front end data service that includes the data.

9. The system of claim 1, wherein at least another portion of the cache framework operates on a server of back-end data service, and wherein the plurality of caches comprises a virtual cache corresponding to a backing data source.

10. The system of claim 9 wherein the back-end data service is coupled to the front-end data service that is the requesting entity, and communicates an error to

the front-end data service upon the return of the indication from the cache framework that at least part of the data was not obtained from the plurality of caches.

11. A method comprising:

generating, by a cache framework of a system comprising a processor, a data structure that identifies a plurality of caches in a cache tier order defining an order in which respective caches of the plurality of caches are to be accessed to service data requests;

in response to a change in volume of the data requests, modifying, by the cache framework, the data structure to add or remove at least one cache to the plurality of caches according to the change in volume of the data requests;

adjusting, by the cache framework, the cache tier order according to the addition or removal of the least one cache to the plurality of caches;

in response to receiving a request for data from a requesting entity, attempting, by the cache framework, to return the data in response to the request, including reading the data structure to access each cache according to the cache tier order until the data is obtained from a cache of the plurality of caches or returning an indication that at least part of the data was not obtained from the plurality of caches, and

wherein at least a portion of the cache framework operates on a server of a front-end data service, and wherein the front-end data service is coupled to communicate a request for the data to a back-end data service upon the return of the indication from the cache framework that at least part of the data was not obtained from the plurality of caches,

wherein the data request for data comprises a batch request for data items, and wherein the cache framework accesses a lowest cache of the plurality of caches to attempt to locate data for the data items, and for each data item not found, to access a higher cache of the plurality of caches to attempt to locate data for the data items until each data item is found or no higher cache remains to access, and

wherein the cache framework writes through data to any lower cache of the plurality of caches if the data is located in a higher cache of the plurality of caches relative to that lower cache.

12. The method of claim 11, further comprising, for any data obtained from the cache, writing through the data to any lower cache of the plurality of caches relative to the cache.

13. The method of claim 11, further comprising receiving a change to the cache tier order.

14. A non-transitory computer-readable medium having instructions stored thereon that, in response to execution, cause cache framework of a system including a processor to perform operations comprising:

generating a data structure that identifies a plurality of caches in a cache tier order defining an order in which respective caches of the plurality of caches are to be accessed to service data requests;

in response to a change in volume of the data requests, modifying the data structure to add or remove at least one cache to the plurality of caches according to the change in volume of the data requests;

adjusting the cache tier order according to the addition or removal of the least one cache to the plurality of caches;

in response to receiving a request for data from a requesting entity, attempting to return the data in response to the request, including reading the data structure to access each cache according to the cache tier order until the data is obtained from a cache of the plurality of caches or returning an indication that at least part of the data was not obtained from the plurality of caches, and

wherein at least a portion of the cache framework operates on a server of a front-end data service, and wherein the front-end data service is coupled to communicate a request for the data to a back-end data service upon the return of the indication from the cache framework that at least part of the data was not obtained from the plurality of caches,

wherein the data request for data comprises a batch request for data items, and wherein the cache framework accesses a lowest cache of the plurality of caches to attempt to locate data for the data items, and for each data item not found, to access a higher cache of the plurality of caches to attempt to locate data for the data items until each data item is found or no higher cache remains to access, and

wherein the cache framework writes through data to any lower cache of the plurality of caches if the data is located in a higher cache of the plurality of caches relative to that lower cache.

15. The non-transitory computer-readable medium of claim 14, wherein the change in volume is a decrease in the volume of the data requests, and the cache framework modifies the data structure to remove the at least one cache from the plurality of caches.

16. The non-transitory computer-readable medium of claim 14, wherein the change in volume is an increase in the volume of the data requests, and cache framework modifies the data structure to add the at least one additional cache to the plurality of caches.