Claims

What is claimed is:

1. A system including:

tier requisition circuitry configured to execute a tier requisition stack, the tier requisition circuitry configured to:

at a data input layer of the tier requisition stack:

obtain historical data including data types, the data types including processor utilization type data, operation rate type data, flush volume type data, or any combination thereof; and

obtain database-compute tolerance data establishing one or more database-compute tolerances;

at a coefficient layer of the tier requisition stack:

access the data types via a memory resource provided by a data-staging layer of the tier requisition stack;

determine a corresponding stratum from among strata for each of the data types, where each stratum of the strata is assigned a corresponding activity coefficient;

for each of the data types, obtain a corresponding type weight, and combine the corresponding activity coefficient in accord with the corresponding type weight to obtain an activity factor;

at a tolerance layer of the tier requisition stack:

determine an initial database-compute tier based on the activity factor, the initial database-compute tier associated with a database-compute allowance;

compare the database-compute allowance to the database-compute tolerance data to determine accommodation of the one or more database-compute tolerances by the initial database-compute tier;

at a time that the initial database-compute tier fails to accommodate the one or more database-compute tolerances, increase from the initial database-compute tier to an adjusted database-compute tier based on the initial database-compute tier and computation of the tolerance layer;

at a requisition layer of the tier requisition stack:

generate a provisional tier requisition token designating a request for date-compute resources at the adjusted database-compute tier; and

based on a finalization directive, generate a finalized tier requisition token; and

network interface circuitry configured to send the finalized tier requisition token to a host interface for control of database-compute resources.

2. The system of claim 1, where the tier requisition circuitry is further configured to, at the tolerance layer:

obtain a tier recommendation associated with the host interface;

compare the tier recommendation to the initial database-compute tier; and

at a time the tier recommendation and the initial database-compute tier differ, generate a difference token.

3. The system of claim 1, where the tier requisition circuitry is configured to, at the tolerance layer:

increase from the initial database-compute tier to the adjusted database-compute tier by:

until the adjusted database-compute tier is found, interactively determining an iterated database-compute tier by increasing from a previous database-compute tier by one increment, the previous database-compute tier being equal to the initial database-compute tier for a first iteration.

4. The system of claim 1, where the tier requisition circuitry is configured to, at a predictive engine layer of the tier requisition stack:

obtain forecasted utilization data including the data types;

obtain a forecasted stratum from among strata for each of the data types, where each forecasted stratum of the strata is assigned a forecast activity coefficient;

obtain forecasted coefficients in accord with the corresponding type weight to obtain a forecasted activity factor; and

determine a forecasted database-compute tier.

5. The system of claim 4, where the tier requisition circuitry is configured to, at the predictive engine layer, train a deep learning algorithm using the historical data to obtain the forecasted utilization data.

6. The system of claim 1, where tier requisition circuitry is configured to, at the tolerance layer, apply a core factor to scale the activity factor relative to a number of processor cores to which the historical data applies.

7. The system of claim 1, where the tier requisition circuitry is configured to, at a consumption savings layer of the tier requisition stack:

obtain consumption metric data for the initial and adjusted database-compute tiers;

determine an initial consumption level using the consumption metric data for the initial database-compute tier;

determine an adjusted consumption level using the consumption metric data for the adjusted database-compute tier; and

compare the initial consumption level to the adjusted consumption level to determine an accommodation consumption factor.

8. The system of claim 7, where the tier requisition circuitry is configured to, at the consumption savings layer, decrease from the adjusted database-compute tier when the accommodation consumption factor exceeds a threshold to obtain a consumption-adjusted database compute tier.

9. The system of claim 1, where the tier requisition circuitry is configured to, at a consumption savings layer of the tier requisition stack:

obtain consumption metric data for a historical database-compute tier and the adjusted database-compute tier, the historical database-compute tier associated with operation during capture of at least a portion of the historical data;

determine an historical consumption level using the consumption metric data for the historical database-compute tier;

determine an adjusted consumption level using the consumption metric data for the adjusted database-compute tier; and

compare the historical consumption level to the adjusted consumption level to determine a consumption change factor.

10. The system of claim 9, where the tier requisition circuitry is configured to, at the consumption savings layer, decrease from the adjusted database-compute tier when the consumption change factor exceeds a threshold to obtain a consumption-adjusted database compute tier.

11. The system of claim 1, where the finalization directive includes:
an operator input command received at a database-compute control-interface generated at a presentation layer of the tier requisition stack;
a feedback-trained machine learning adjustment determined based on an operator input command history; or
both.

12. The system of claim 1, where the one or more database-compute tolerances include:
a tolerance for non-online-transaction-processing (non-OLTP) storage;
a concurrent sessions tolerance factor;
a concurrent requests tolerance factor;
a storage volume tolerance;
a request accumulation tolerance; or
any combination of the foregoing.

13. A method including:
at a data input layer of a tier requisition stack:
obtaining historical data including data types, the data types including processor utilization type data, operation rate type data, flush volume type data, or any combination thereof; and
obtaining database-compute tolerance data establishing one or more database-compute tolerances;
at a coefficient layer of the tier requisition stack:
accessing the data types via a memory resource provided by a data-staging layer of the tier requisition stack;

determining a corresponding stratum from among strata for each of the data types, where each stratum of the strata is assigned a corresponding activity coefficient;

for each of the data types, obtaining a corresponding type weight, and combining the corresponding activity coefficient in accord with the corresponding type weight to obtain an activity factor;

at a tolerance layer of the tier requisition stack:

determining an initial database-compute tier based on the activity factor, the initial database-compute tier associated with a database-compute allowance;

comparing the database-compute allowance to the database-compute tolerance data to determine accommodation of the one or more database-compute tolerances by the initial database-compute tier;

at a time that the initial database-compute tier fails to accommodate the one or more database-compute tolerances, increasing from the initial database-compute tier to an adjusted database-compute tier based on the initial database-compute tier and computation of the tolerance layer;

at a requisition layer of the tier requisition stack:

generating a provisional tier requisition token designating a request for date-compute resources at the adjusted database-compute tier; and

based on a finalization directive, generating a finalized tier requisition token; and

sending, via network interface circuitry, the finalized tier requisition token to a host interface for control of database-compute resources.

14. The method of claim 13, further including:

at the tolerance layer:

obtaining a tier recommendation associated with the host interface;

comparing the tier recommendation to the initial database-compute tier; and

at a time the tier recommendation and the initial database-compute tier differ, generating a difference token.

15. The method of claim 13, further including:

at the tolerance layer:

increasing from the initial database-compute tier by:

until the adjusted database-compute tier is found, interactively determining an iterated database-compute tier by increasing from a previous database-compute tier by one increment, the previous database-compute tier being equal to the initial database-compute tier for a first iteration.

16. The method of claim 13, further including:

at a predictive engine layer of the tier requisition stack:

obtaining forecasted utilization data including the data types;

obtaining a forecasted stratum from among strata for each of the data types, where each forecasted stratum of the strata is assigned a forecast activity coefficient;

obtaining forecasted coefficients in accord with the corresponding type weight to obtain a forecasted activity factor; and

determine a forecasted database-compute tier.

17. The method of claim 16, further including, at the predictive engine layer, training a deep learning algorithm using the historical data to obtain the forecasted utilization data.

18. A product including:

machine-readable media other than a transitory signal; and

instructions stored on the machine-readable media, the instructions configured to cause a machine to:

at a data input layer of a tier requisition stack:

obtain historical data including data types, the data types including processor utilization type data, operation rate type data, flush volume type data, or any combination thereof; and

obtain database-compute tolerance data establishing one or more database-compute tolerances;

at a coefficient layer of the tier requisition stack:

access the data types via a memory resource provided by a data-staging layer of the tier requisition stack;

determine a corresponding stratum from among strata for each of the data types, where each stratum of the strata is assigned a corresponding activity coefficient;

for each of the data types, obtain a corresponding type weight, and combine the corresponding activity coefficient in accord with the corresponding type weight to obtain an activity factor;

at a tolerance layer of the tier requisition stack:

determine an initial database-compute tier based on the activity factor, the initial database-compute tier associated with a database-compute allowance;

compare the database-compute allowance to the database-compute tolerance data to determine accommodation of the one or more database-compute tolerances by the initial database-compute tier;

at a time that the initial database-compute tier fails to accommodate the one or more database-compute tolerances, increase from the initial database-compute tier; and

determine an adjusted database-compute tier based on the initial database-compute tier and computation of the tolerance layer;

at a requisition layer of the tier requisition stack:

generate a provisional tier requisition token designating a request for date-compute resources at the adjusted database-compute tier; and

based on a finalization directive, generate a finalized tier requisition token; and

send, via network interface circuitry, the finalized tier requisition token to a host interface for control of database-compute resources.

19. The product of claim 18, where the finalization directive includes:

an operator input command received at a database-compute control-interface generated at a presentation layer of the tier requisition stack;

a feedback-trained machine learning adjustment determined based on an operator input command history; or

both.

20. The product of claim 18, where the one or more database-compute tolerances include:

a tolerance for non-online-transaction-processing (non-OLTP) storage;

a concurrent sessions tolerance factor;

a concurrent requests tolerance factor;

a storage volume tolerance;

a request accumulation tolerance; or

any combination of the foregoing.