Special instructions if you are using XdbCloud:

1. Disable App_Config/Include/Sitecore.ContentSearch.Z.Index.Analytics.hotfix-132631.config
2. Enable App_Config/include/ XdbCloud/ Sitecore.ContentSearch.Z.Index.Analytics.hotfix-132631.config

Hotfix description:

This hotfix improves the speed of contact processing and ContactBulkUpdateManager that are ran while a contact list in the List Manager is created, indexed and fully ready to be used. The speed improvement also applies to part of the processing done for live traffic, which not only includes contact change processing but also interaction aggregation (live and historical).

The hotfix also fixes a race condition where multiple contact processing or interaction aggregation threads were used and in some occasions the search related processors failed to be initialized.

Below the specific changes:

1. The system no longer indexes the latest visit date for the contacts.
    a. The problem: the latest visit date is not directly available on the contact during indexing and it was using the interactions index to fetch it. This happened for every contact, which not only slows down aggregation due to the latency to resolve those requests but also puts pressure into the Solr server.
    b. The patch removes this logic completely in the LoadFields processor hooked in the following pipeline: contactindexable.loadfields. This logic is used by both contact and interaction aggregation pipelines processors: ContactChangeContactAggregator and AnalyticsContactAggregator
    c. Breaking change: The LatestVisitDate field has been removed from the IndexedContact class.
2. The logic that ensures removed contact addresses are removed from the index has been updated.
    a. The problem: it is not possible to determine if an address has been removed from the contact, so the system was querying the search index for existing addresses to detect any addresses that are no longer valid. Like above, this added latency to the aggregation and put pressure into the Solr server.
    b. The patch changes the logic so that, instead of trying to detect which addresses have been removed from the contact, it replaces the addresses of the contact with the new list. For this it includes in the set of changes to index a delete of all contact's addresses just before the new list is added.
    c. The change is done in:
        i. the AggregatorHelper used by both contact and interaction aggregation pipelines processors: ContactChangeAddressAggregator and AnalyticsAddressAggregator
        ii. breaking changes:

        1. New Delete overload was added to IIndexOperations in Sitecore.ContentSearch.dll
        2. New Delete overload was added to IProviderUpdateContext in Sitecore.ContentSearch.dll
3. The throttling logic for the sitecore analytics index has been fixed and is always enabled. ThrottlingEnabled setting is not used.
    a. The logic was greatly simplified so all it does is make sure that the indexing queues don't grow over the MaximumQueueSize
    b. This was needed because contact processing and aggregation was unintendedly being implicitly throttled by the slow logic related to #1 and #2 above, and with that removed the indexing queues can grow too large.
4. The interval for the sitecore analytics index refresh strategy has been updated to 15 seconds (instead of 1 minute).
5. The ContactChangeProcessor and ObservableAggregator base classes in Sitecore.ContentSearch.Analytics.dll have been updated to fix the race condition mentioned earlier.
6. More debug level logging has been added to the indexing related operations.