

ANZOGRAPH[®] BENCHMARKS

Why “Right Tool, Right Job” Applies to Graph Databases

White Paper



CONTENTS

Executive Summary	2
Test Conditions	3
The Results	4
Conclusion	8

“When selecting a database for your enterprise architecture, the choice depends very much on the workload that is presented to it.”

Executive Summary

The interest in graph databases has increased a great deal over these past years. Graph databases provide a unique focus on data relationships, or the connection in data. The perceived value of connected data has exploded, as businesses look to drive innovation as they connect supply chain, IoT devices, marketing technology, logistics, payment history, making the value of connectedness across all those data elements increase exponentially. The market has expanded with AnzoGraph from Cambridge Semantics, AWS Neptune, Janus Graph, OrientDB, ArangoDB, StarDog, AllegroGraph and RedisGraph to name just a few.

It's an important concept that these databases scale in different ways. Graph OLTP (Online Transactional Processing) databases like Neo4j, AWS Neptune and most of the other mentioned above scale to accommodate more transactions and more users while graph OLAP (Online Analytical Processing) databases are the basis for scalable graph analytics (includes graph-based BI-style analytics, graph algorithms, inferencing and other graph-based analytics). When selecting a database for your enterprise architecture, the choice depends very much on the workload that is presented to it.

More specifically, graph OLTP databases read and write single records fast. Graph OLAP databases read millions of records fast, but do not read/write individual records fast. Often you need both to create a solution that includes transaction as well as analytic components. Both forms of database can provide real-time data access. Both forms of database can use replication to address more users. Specifically, graph OLAP databases can be scaled by adding additional servers to process the queries in parallel to provide improved computation and performance.

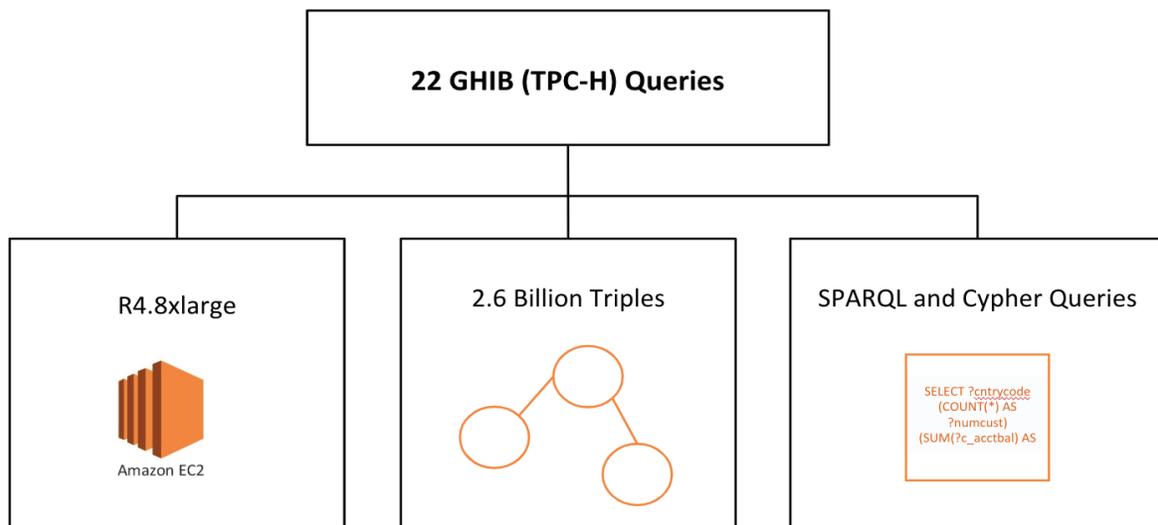
In this paper, we'll shine a light on the difference between graph OLTP and graph OLAP databases by comparing analytical style queries on both Neo4J, a graph OLTP database, and AnzoGraph by Cambridge Semantics, a graph OLAP database. The results show that scaling for this type of query with huge volumes of data simply isn't feasible on most graph OLTP platforms. However, graph OLAP-style databases like AnzoGraph can handle this unique workload and scale to handle trillions of triples or more with elegance.

Test Conditions

We started our testing by setting up a single Amazon EC2 instance. All tests were executed using Amazon EC2 R4.8xlarge instances. These virtual machines have 32 virtual CPUs, which corresponds to 16 cores. The nodes have 244 GiB of memory per node with 500 GB SSD drives. Both AnzoGraph and Neo4J primarily uses RAM memory and therefore the size of the SSD plays a smaller role than memory.

Analytical Queries

The Graph AdHoc Business Intelligence Benchmark (GHIB), which is based on the TPC-H standard, measures the timing of 22 queries upon a synthetic data set. The queries are ported from the TPC-H benchmark and operate on a generated data set that itself is compatible with TPC-H. The original TPC-H queries were written in SQL, while GHIB uses SPARQL and/or Cypher. Even though it's possible to run these queries on a OLTP database, analytical queries like this are better suited for OLAP.



Data

The GHIB data set contains “orders” for “parts” made by “customers” and the associated “suppliers” from which the customers purchased. The GHIB data set itself is highly scalable with a scale factor of 25 being suitable for the chosen AWS instance type. At a scale factor of 25 the GHIB data set contains 2,637,695,761 triples. In vertex-edge terminology, this is about 200 million vertexes, with about 1 billion edges and 1.4 billion properties.

We later scaled up to a multi-node database system and over 100 billion triples. With the multi-node test, network speed is incredibly important. For these test machines, Amazon guaranteed 10 GiB, although faster systems are available and would likely improve query performance. Persistence, which replicates a copy of data on disk, was disabled for this benchmark.

Solutions Compared

The test compared AnzoGraph 4.x to Neo4J 3.4.9 community edition to provide a comparison point with a popular graph database. Neo4J is an OLTP style graph database. Like most of the other graph databases on the market today, Neo4J scales to meet the needs of high concurrency of small-transactions on small data sizes, while AnzoGraph scales to be able to transverse the database with analytical style queries that is efficient for small amounts of data but can scale to a LOT of data. The queries tested tended to be ones that traversed the database.

Why this is an unfair test

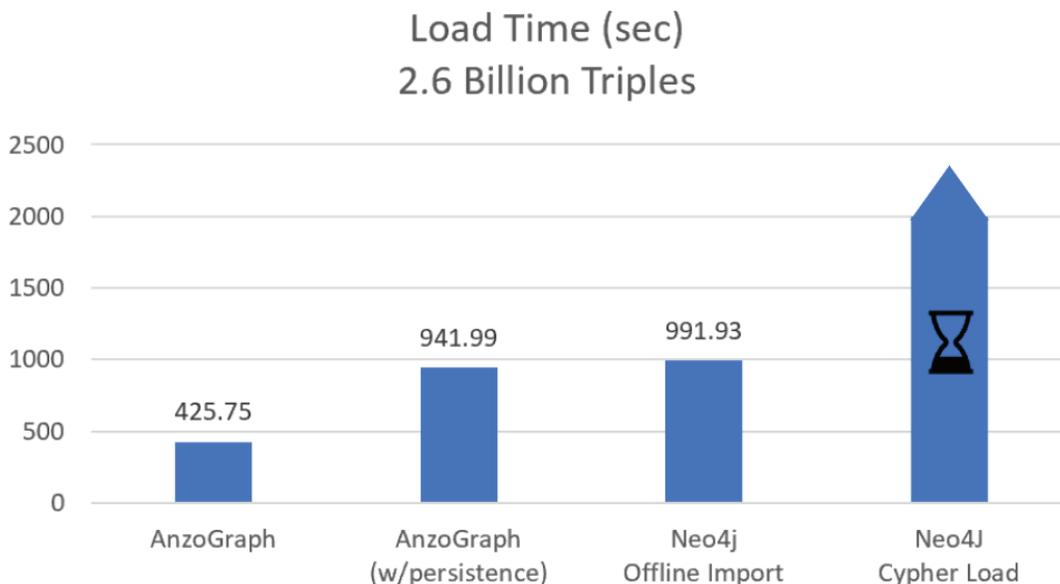
This GHB benchmark is intended to bring awareness to the difference between OLAP and OLTP queries. It's unfair because this benchmark asks an OLTP solution (Neo4J) to do something it's not designed to do, namely perform bi-style analytics. The intent is to demonstrate benefit that is possible from using the right tool for the right job. BI-style analytics with graph databases can be very important when reporting, aggregating and analyzing data and for a variety of other use cases.

The Results

Data Loading

Loading is a very important part of the utility of a graph database. Often a graph database is paired with other solutions in the enterprise architecture. The ability to spin up, load and query in a small amount of time is crucial when part of an enterprise architecture. If large amounts of data take forever to load, the graph engine becomes less useful.

For loading data in Neo4J, you can either use Cypher to load a running database or an offline loader. In our test, Neo4J could not complete the load in a reasonable timeframe using Cypher loads, so we leveraged the offline loader. The offline importer is suitable for the initial creation of the database, but the database must be offline to load data. If you need to incrementally add data then you must use the load function.



For Anzograph, we loaded data into a running database. On this platform, you have the option of loading data with or without persistence. If you load data with persistence, data is loaded in memory and in a persistence layer on disk, allowing the database to be quickly re-started with the full dataset. With persistence turned on, AnzoGraph beat the Neo4J offline loader by a slim margin, and AnzoGraph did not require shutting the database and loading it offline. However, if you just ask for a memory load, which is often the case in analytic workloads, AnzoGraph completed the load in about half the time, also fully online.

These times do not reflect the additional work required to format the data appropriately for the offline import. If the data is, for example, CSV files from a relational database, expect to spend time with Neo4J in preparing the data in a custom format for Neo4J offline loader. After the data is loaded, Neo4J takes some time to generate indexes, too. This benchmark does not reflect the time required for indexes to be generated. However, keep in mind that you may need to load data and generate indexes in some solutions. AnzoGraph does not require that you generate indexes.

Benchmark Impact

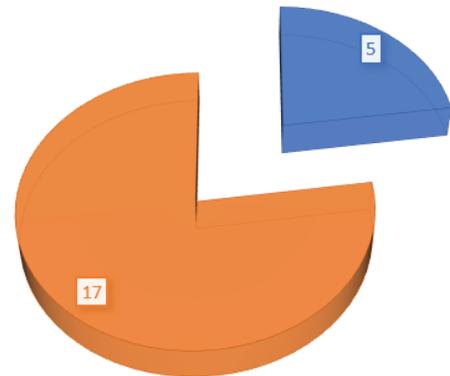
Consider load time and load procedure when choosing a graph analytics solution. In real-world production environments, analytics operations are akin to changing a tire while the car is traveling down the highway. It's difficult to stop operations for loading and refreshing the database.

Query Completeness

Part of the true test of any database is how complete the analytics are. Does the database allow for standard queries, like benchmark queries, to be executed with very limited modification and fuss. The next part of the benchmark called for running 22 analytical queries on a single node with a medium scale factor. For our first test, we compared SPARQL on AnzoGraph with Cypher on Neo4J. Of the 22 queries, we were able to complete all 22 on AnzoGraph while Neo4J could not complete 5 of the 22 queries.

On two of the queries, Neo4J times out. On three of the queries, our labs could not determine how to run the equivalent query in Cypher.

QUERY COMPLETION



Timed-out or Incomplete Queries Skipped in this Benchmark

Remove Query	Name	Description	Reason
2	Minimum Cost Supplier	Determines which supplier offers the lowest price for a given part in a given region. Since multiple suppliers in a region could offer the same part at the same (minimum) cost, the query also lists the suppliers with the 100 highest account balances. For each supplier, the query returns the part's number and manufacturer and the supplier's account balance, name, nation, address, phone number, and comments.	Timed Out
15	Top Supplier	Finds the supplier who contributed the most overall revenue for parts shipped between 12/1/1994 and 3/1/1995. The results list the supplier's key, name, address, phone number, and total revenue.	Timed Out
20	Potential Part Promotion	Identifies suppliers in a particular nation who have selected parts that may be good candidates for a promotional offer.	Not Implemented
21	Suppliers Who Kept Orders Waiting	Identifies suppliers who did not ship required parts on time.	Not Implemented
22	Global Sales Opportunity	Identifies geographies where customers might be likely to make a purchase. The query reports on customers in seven specific countries (as identified by country code) who have not placed orders in seven years but who still have a positive account balance that is greater than the average amount.	Not Implemented

Benchmark Impact

Make certain that the analytical solution you choose can easily perform the type of analytics required by your team. Database vary greatly in this regard.

“AnzoGraph was significantly faster on the analytical queries than Neo4J on a single node.”

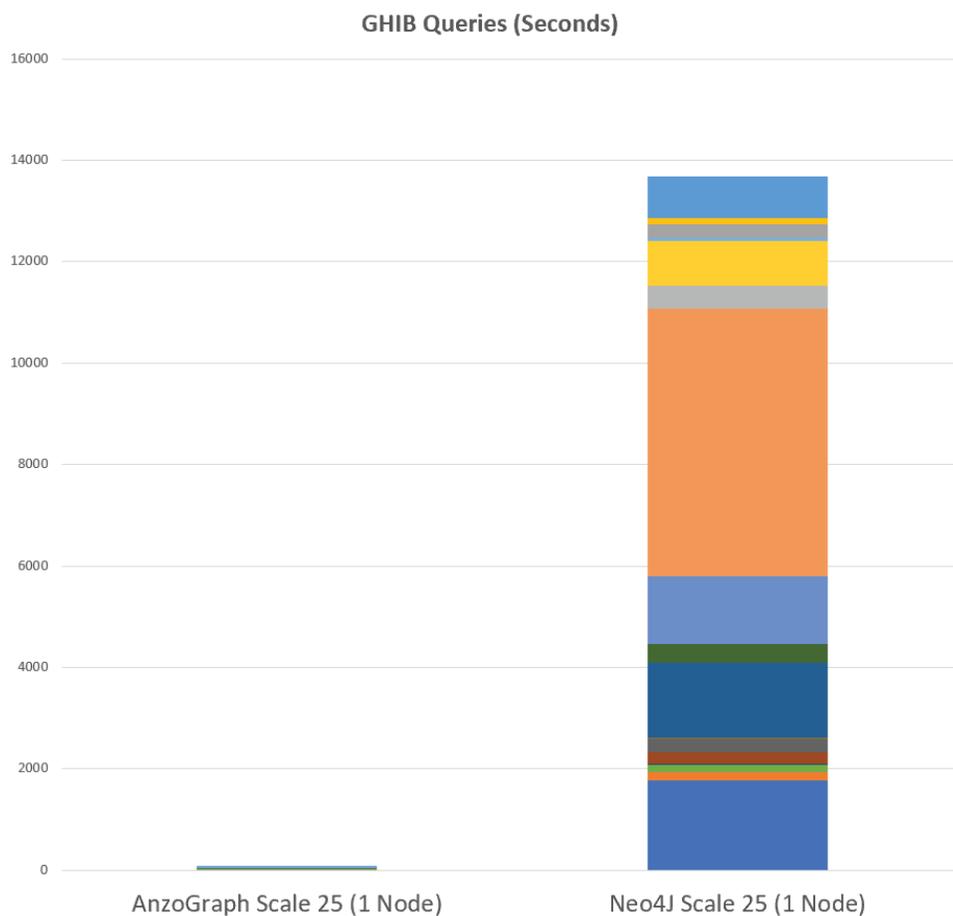
Single Node Speed Comparison

Now that we've established which of the 22 queries of our GHIB benchmark are executable on both solutions, the test compared the 17 (of 22) runnable queries that remained. Remember, the data in this scale factor 25 test contained about 2.6 billion triples. The GHIB queries were designed to have a collection of long-running and short-running queries against that data set.

In our test, AnzoGraph was significantly faster on the analytical queries than Neo4J on a single node. AnzoGraph took about 1 minute total to complete the entire set of the queries, while Neo4J took 3.8 hours. In other words, you can complete the benchmark queries 217 times with AnzoGraph for every 1 completion of Neo4J. The five queries that wouldn't run on Neo4J each added a fraction of a second to the run.

Benchmark Impact

When it comes to analytical queries, an OLAP-style database like AnzoGraph can speed through queries at small scale. The difference, even at this smaller scale can mean analytics that return in seconds rather than minutes, let alone hours.



For this workload, you can complete the analytical benchmark queries 217 times with AnzoGraph for every 1 completion of Neo4J

“Even with over 100 billion triples, AnzoGraph was able to speed through the GHIB queries at break-neck speed.”

More Data, More Nodes

AnzoGraph was built with massive scalability in mind. As a part of our current benchmarking effort we did a multi-node cluster benchmark to understand the impact of scale on the GHIB benchmark.

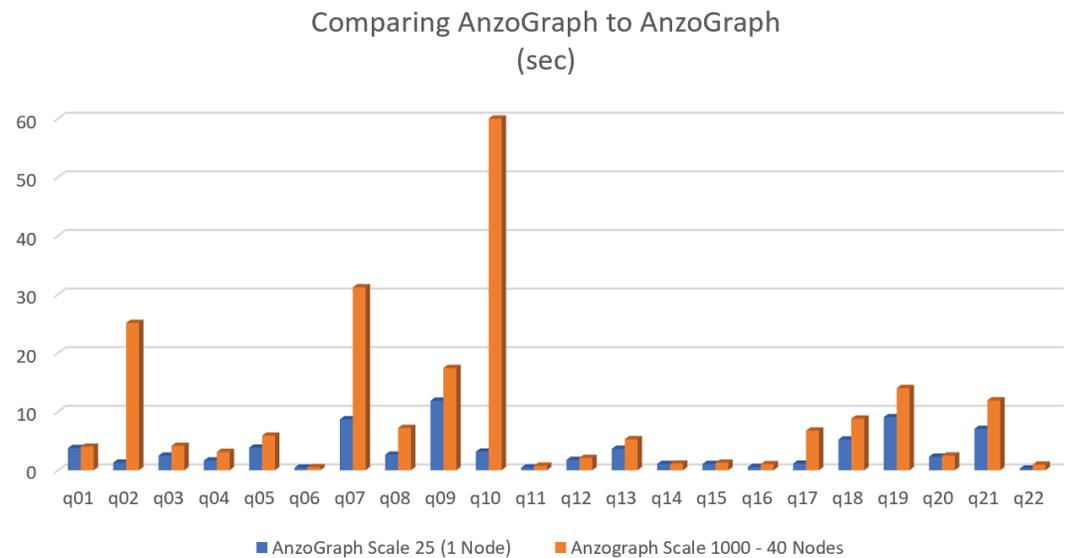
Using the GHIB data set at a scale factor of 1000 brought the number of triples up from 2.6 billion to over 100 billion triples. We tested this data set on a 40 node cluster of Amazon r4.8xlarge nodes, launched and managed as a Cloud Formation Service. The entire run-thru of all 22 queries at that scale took about 3 ½ minutes with the longest running query at about 1 minute.

We didn't attempt to run the benchmark at this scale for Neo4J. The community edition of Neo4J supports one node. Also, given one node benchmarks, we anticipated that both load and execution of the queries would take too long.

Benchmark Impact

In an MPP database like AnzoGraph, you can scale the number of nodes to align the desired query performance with the amount of data you have. Even with over 100 billion triples, AnzoGraph was able to speed through the GHIB queries at break-neck speed.

In 2016, we did a trillion triple benchmark. Since then AnzoGraph has improved significantly.



At scale AnzoGraph speeds through the 100 Billion Triples benchmark, tackling 22 GHIB queries in 3 ½ minutes

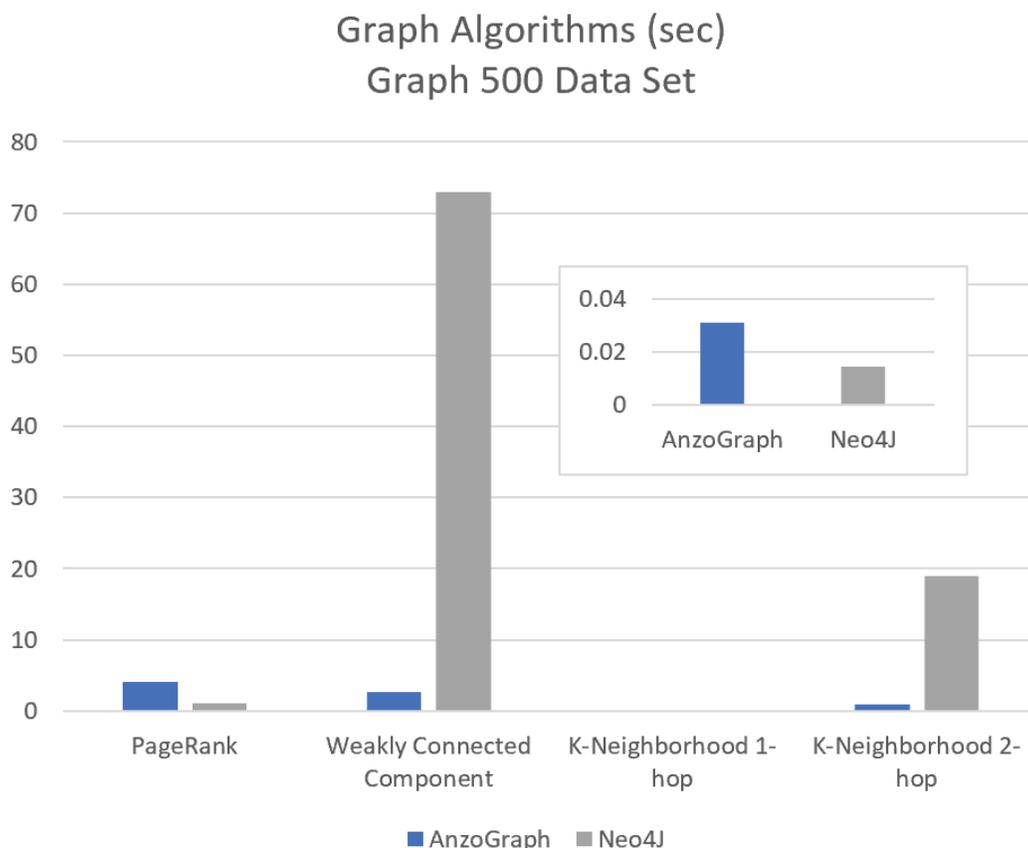
“Consider the scale and types of algorithms you’re going to run in determining the best engine for your use case.”

Graph Algorithm Queries

Graph Algorithms are an important part of the unique value of graph databases. We took a look at how fast graph algorithms like PageRank, Weakly Connected Component and k-Neighborhood run on both OLAP and OLTP systems.

The data set we selected is called the Graph 500 set. While some algorithms ran faster, the algorithms that traversed the databases tended to do better on the OLAP database. For the tests, we loaded 2.4 million vertices and 64 million edges and ran the tests specified.

Like the GHIB tests above, this chart only represents the queries running on a single node. However, algorithms can scale approximately proportionally to larger data-sets on larger clusters of AnzoGraph.



Benchmark Impact

There was not a significant difference in the graph algorithms we chose to run for the Graph 500 test except for the Weakly Connected Component and K-Neighborhood algorithms with multiple hops. Some algorithms ran faster on the OLTP solution. Those algorithms that traversed the database ran faster on the OLAP solution. Consider the scale and types of algorithms you’re going to run in determining the best engine for your use case.

“With AnzoGraph, analysts can perform deep analysis and traverse the database across billions and trillions of triples or more”

“Warming Up” a Database

Many databases, particularly databases that can handle big data, have differences in performance when running a “cold” database or a warm one. There are several potential reasons why query performance may vary on the first run vs later runs.

For AnzoGraph, we benefit from a warm-up because of our feature that compiles queries the first time each is run. Although query compiling takes a small amount of time on the first run, it means that queries will run fast, even with huge amounts of data. The parameters of the query can change from run to run, but the engine uses previously compiled segments on subsequent runs. Thus, for example, a query originally referencing “Pennsylvania” will not recompile when a similar query referencing “Ohio” is submitted.

Other solutions improve performance with warming up because they cache query-results. This is not the case with AnzoGraph.

When performing your own benchmarks, be aware that warming up the database by running each query once will have an impact on performance. Compare first run times to subsequent run times for the best comparison.

Conclusion

In the graph database world, there are wide distinctions. There are differences in the use case between OLAP and OLTP databases. The distinction is similar to the RDBMS world where both fast small-transaction engines and analytical engines exist. With AnzoGraph, analysts can perform deep analysis and traverse the database across billions and trillions of triples or more. There is an expectation of scale with AnzoGraph which allows for administrators choose the cluster size for the data size, and more important, the performance desired at that data size.

Cambridge Semantics Inc., The Smart Data Company®, is a big data management and enterprise analytics software company that offers a universal semantic layer to connect and bring meaning to all enterprise data. The company offers two award winning products: Anzo® for Enterprise Knowledge Graphs and integrated analytics and AnzoGraph®, a graph analytics database.



1 Beacon Street, 15th Floor,
Boston, MA 02108 USA

CambridgeSemantics.com

info@cambridgesemantics.com



//CODiE//
2018 SIIA CODiE FINALIST

