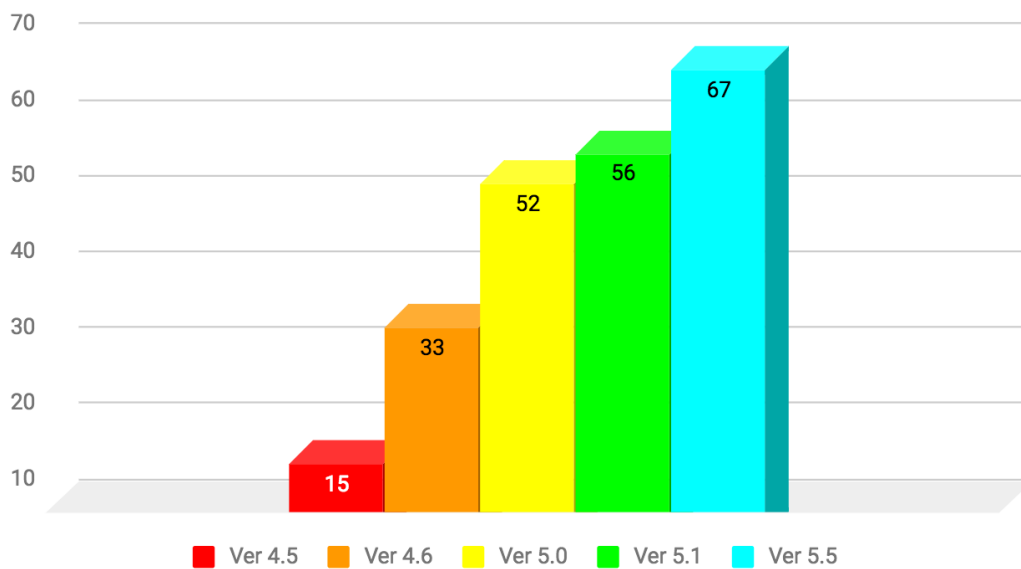


Differences between server versions

Each version of Couchbase Server has new features added and other features expanded. The following tables give you a quick overview of those differences.

Features (graph)

Number of Features



Features (table)

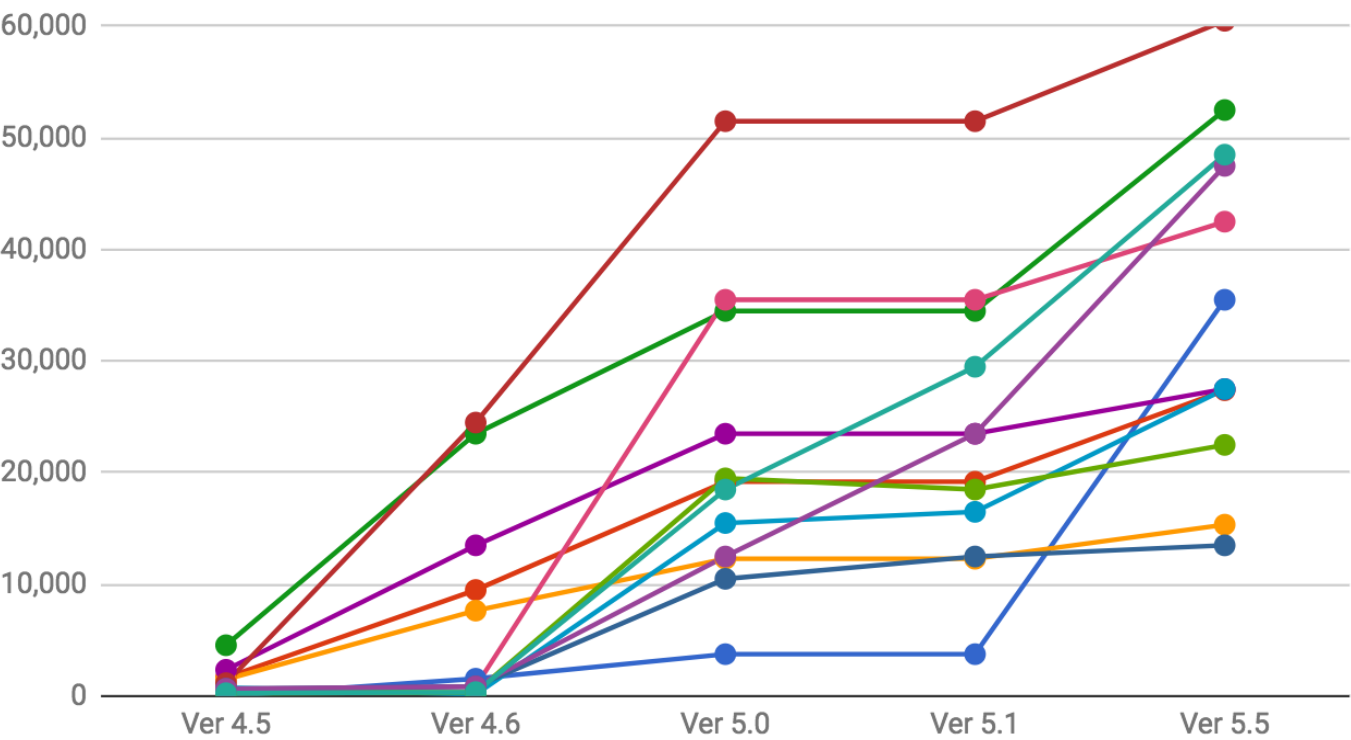
Feature Description	4.5.1	4.6.2	5.0	5.1	5.5
	<i>June 2016</i>	<i>Dec 2016</i>	<i>June 2017</i>	<i>Dec 2017</i>	<i>June 2018</i>
Ansi JOIN	-	-	-	-	✓
Ansi NEST	-	-	-	-	✓
Ansi JOIN Hints (Hash & NL)	-	-	-	-	✓
N1QL Backfill Support	-	-	-	-	✓
Index Partitioning	-	-	-	-	✓
Auto Re-Prepare	-	-	-	-	✓
IPV6 Support	-	-	-	-	✓
Log Redaction	-	-	-	-	✓
Aggregate Pushdown	-	-	-	-	✓
X.509 Support	-	-	-	Index only	All 3 services
ALTER INDEX	-	-	-	✓	✓
Client Certificates	-	-	One way	Flexible	✓
Full Text Search (FTS)	-	-	✓	✓	✓
Adaptive Indexes	-	-	✓	✓	✓
CURL() Functionality	-	-	✓	✓	✓
Monitoring & Profiling Queries	-	-	✓	✓	✓
META() Indexing & Querying	-	-	✓	✓	✓
Index Pushdown Optimizations	-	-	✓	✓	✓
Plasma Storage Engine	-	-	✓	✓	✓
Index HA & Manageability	-	-	✓	✓	✓

Robust Failure Detection	-	-	✓	✓	✓
Ephemeral Buckets	-	-	✓	✓	✓
CBImport & CBExport	-	-	✓	✓	✓
Visual Query Plan	-	-	✓	✓	✓
Sync Gateway & CB Lite	-	-	✓	✓	✓
Big Data Connectors	-	-	✓	✓	✓
Talend Connector	-	-	✓	✓	✓
Web Console	-	✓	Redesigned	✓	Redesigned again
RBAC	-	Admin only	Admin & Applications	✓	✓
FROM w/generic N1QL expression	-	✓	✓	✓	✓
Array indexing	-	✓	✓	✓	✓
Cross Datacenter Replication	-	✓	✓	✓	✓
Security with pluggable authentication module	-	✓	✓	✓	✓
Pushdown LIMIT and ORDER when UNNEST is covered	-	✓	✓	✓	✓
Pushdown LIMIT during IntersectScan, UnionScan, and DistinctScan	-	✓	✓	✓	✓
OrderedIntersectScan	-	✓	✓	✓	✓
Secret Management	-	✓	✓	✓	✓
Block nested loop JOINS	-	✓	✓	✓	✓
Sorting by any field	-	✓	✓	✓	✓
Use of functional indexes SUFFIXES() and TOKENS()	✓	7 added functions	✓	✓	✓

Date Functions	✓	15 added	✓	✓	✓
Array Functions	✓	9 added	✓	✓	✓
Token Functions	✓	4 added	✓	✓	✓
String Functions	✓	2 added	✓	✓	✓
Object Functions	✓	4 added	✓	✓	✓
Query monitoring & profiling	✓	✓	Much more fine-grained	✓	✓
Index JOIN	✓	✓	✓	improved	✓
N1QL "pretty" setting	✓	✓	✓	✓	✓
Non-blocking DISTINCT Operator	✓	✓	✓	✓	✓
Simplified Transitioning from RDBMS to NoSQL	✓	✓	✓	✓	✓
SUFFIXES() for Efficient Pattern Matching	✓	✓	✓	✓	✓
UPDATE for nested arrays	✓	✓	✓	✓	✓
ARRAY_INTERSECT()	✓	✓	✓	✓	✓
Backup and Restore	✓	✓	✓	✓	✓

Performance (graph)

Queries per Second for each Query Type



Performance (table)

Each feature is tested with various queries that range in complexity, so the below table lists the range of 1,000 queries per second throughput from extremely complex queries to simple/common queries.

The following table contains numbers from a CentOS machine with 64 GB RAM.

Feature (1,000 Queries/Second)	Δ Between 4.5.1 & 5.5	4.5.1	4.6.2	5.0	5.1	5.5
Aggregation	-	-	-	-	-	0.02 - 1.6
Inner Index JOIN	-	-	-	-	-	8 - 22
Multi-key Inner Index JOIN	-	-	-	-	-	8 - 22
SOE, SH2, 4 Node	-	-	-	7.9 - 8	8 - 8.1	26 - 27
UNNEST	-	-	5.5 - 7.4	9 - 11	9 - 11	9 - 11
Sub-doc search; MOI; request_plus	-	-	1.5 - 1.6	3.7 - 3.8	3.7 - 3.8	11 - 12
Sub-doc search; MOI; not_bounded	-	-	1.5 - 1.6	3.7 - 3.8	3.7 - 3.8	35 - 36
Array Indexing	30x	1.6 - 1.7	2 - 17	2.4 - 36	2.4 - 36	2.8 - 52
JOIN	11x	0.3 - 2.7	0.3 - 15	0.6 - 24	0.6 - 24	0.7 - 30
Key-Value UPDATE	11x	4.5 - 4.6	23 - 24	34 - 35	34 - 35	52 - 53
Range Scan w/MOI	3x	0.3 - 0.4	1 - 1.1	1.2 - 1.3	1.2 - 1.3	1.3 - 1.4
Pagination; MOI; request_plus	6x	2.5 - 2.6	11 - 11.5	15 - 16	16 - 17	16 - 17
Pagination; MOI; not_bounded	11x	2.3 - 2.4	13 - 14	23 - 24	23 - 24	27 - 28
Range Scan with Limit; request_plus	35x	0.7 - 0.8	0.1 - 0.2	15 - 16	16 - 17	27 - 28
Range Scan with Limit; not_bounded	8x	2.6 - 2.7	12 - 13	19 - 20	19 - 20	21 - 22

Range Scan w/Limit; Offset; Covering; MOI; request_plus	30x	0.5 - 0.6	0.8 - 0.9	17 - 18	17 - 18	17 - 18
Range Scan w/Limit; Offset; Covering; MOI; not_bounded	71x	0.5 - 0.6	0.8 - 0.9	35 - 36	35 - 36	42 - 43
Range Scan w/Limit; Offset; MOI; request_plus	257x	0.06 - 0.07	0.1 - 0.2	16 - 17	16 - 17	17 - 18
Range Scan w/Limit; Offset; MOI; not_bounded	76x	0.2 - 0.3	0.3 - 0.4	19 - 20	18 - 19	22 - 23
Index Projection; MOI; not_bounded	50x	1.1 - 1.2	24 - 25	51 - 52	51 - 52	60 - 61
Composite Filters (1000 -> 1), Covered, MOI, request_plus	20x	0.6 - 0.7	0.8 - 0.9	10 - 11	12 - 13	13 - 14
Composite Filters (1000 -> 1), Covered, MOI, not_bounded	68x	0.6 - 0.7	0.8 - 0.9	12 - 13	23 - 24	47 - 48
Composite Filters (1000 -> 1), MOI, request_plus	280x	0.04 - 0.05	0.2 - 0.3	11 - 12	12 - 13	13 - 14
Composite Filters (1000 -> 1), MOI, not_bounded	163x	0.2 - 0.3	0.3 - 0.4	18 - 19	29 - 30	48 - 49
COUNT DISTINCT, MOI, request_plus	8x	1.3 - 1.4	7 - 8	10 - 11	10 - 11	11 - 12
COUNT DISTINCT, MOI, not_bounded	12x	1.3 - 1.4	9 - 9.1	14 - 15	14 - 15	17 - 18

Limitations (table)

Feature Description	4.5.1	4.6.2	5.0	5.1	5.5
<i>Release Date</i>	<i>June 2016</i>	<i>Dec 2016</i>	<i>June 2017</i>	<i>Dec 2017</i>	<i>June 2018</i>
Documents per bucket	10,000	10,000	No limit	No limit	No limit
Document size	20 MB	20 MB	No limit	No limit	No limit
Cluster sub-doc path length	-	-	1024 k	1024 k	1024 k
Cluster sub-doc path nesting level	-	-	32	32	32