

Vektör Veritabanları Karşılaştırma Raporu

1. Giriş

Bu rapor, sınırlı kaynaklara sahip (16 GB RAM) bir sistemde kullanılabilecek başlıca vektör veritabanlarını karşılaştırır. Özellikle MongoDB Atlas Vector Search, PostgreSQL + pgvector, Qdrant, Milvus, Weaviate ve Redis (Redisearch vector) dikkate alınarak; her birinin bellek kullanımı, indeksleme stratejileri, doğruluk/performans dengesi, ölçeklenebilirlik, entegrasyon kolaylığı ve Türkçe ağırlıklı RAG sistemlerinde uygulanabilirliği analiz edilmiştir.

2. Karşılaştırma Kriterleri

1. Bellek Verimliliği ve 16 GB RAM altında çalışabilirlik
2. İndeksleme algoritmaları ve doğruluk/hız trade-off'ları (HNSW, IVF, etc.)
3. Kalıcı depolama ve disk bazlı seçenekler (RAM yetersizse fallback)
4. Filtreleme, metadata desteği ve karma sorgular (vector + attribute)
5. Çok-dilli/Türkçe içeriklerle semantic arama performansı
6. Kolay entegrasyon (Python/Flask, client kütüphaneleri)
7. Yazılım olgunluğu, topluluk ve bakım durumu
8. Dağıtıklık vs. tek düğüm çalıştırma (local 16GB senaryosu)
9. Gecikme ve throughput (özellikle düşük kaynakta)
10. Tutarlılık ve ACID/transaction desteği

PostgreSQL + pgvector

PostgreSQL'e pgvector eklentisiyle yüksek boyutlu vektörlerin saklanması ve cosine/inner product benzerlik araması yapılabilir. Tek stack yaklaşımıyla metadata ve vektörler aynı tabloda tutulabilir.

Avantajlar:

- ACID uyumlu, güçlü tutarlılık ve transaction desteği. (Reddit ve blog kaynakları tarafından vurgulanıyor) [\[cite\]turn0search9\[](#)
- Ekstra sistemler gerekmeden hem metin hem vektör verisi aynı yerde durabilir. [\[cite\]turn0search9\[](#)
- Topluluk olgunluğu ve geniş PostgreSQL ekosistemi. [\[cite\]turn0search15\[](#)

Dezavantajlar:

- - Dedicated vector store'lara göre retrieval performansı (özellikle büyük veri setlerinde) daha düşük olabilir; ancak doğru index ve top-k stratejileriyle yeterli hale getirilebilir. [\[cite\]turn0search2\[](#)
- - HNSW gibi yapıların konfigürasyonu ve bellek kullanımı dikkatle ayarlanmalı; 16GB'ta büyük index'ler RAM'i zorlayabilir. [\[cite\]turn0search12\[](#)

16 GB RAM için öneriler / kullanım notları:

- - Düşük boyutlu/orta ölçekli koleksiyonlar için HNSW ile top-N retrieval yap, 'ivfflat' gibi varyantlar yok ama performans optimize edilebilir. [\[cite\]turn0search12\[](#)
- - Gerekirse partitioning ve konu bazlı segmentasyon ile bellek baskısını azalt.
- - Index oluştururken 'pgvector' parametrelerini (efConstruction, M) 16GB'a göre küçült ve test et. [\[cite\]turn0search2\[](#)

Kaynaklar:

- - Why Postgres Wins for AI and Vector Workloads (Tigerdata blog) Apr 2025
- - Pgvector vs. Qdrant comparison blog (Tigerdata) May 2025
- - Reddit tartışmaları ve deneysel karşılaştırmalar (Postgres pgvector avantajları)

MongoDB Atlas Vector Search

MongoDB Atlas, yerleşik vektör arama desteği sunar ve metadatalarla birlikte vektör sorgularını birleştirmeye izin verir. Atlas üzerinden yönetilen hizmet olarak ölçeklenebilir.

Avantajlar:

- - Native vektör arama ve filtreleme birlikte; JSON-benzeri esnek schema ile attribute+vector sorguları kolay. [\[cite\]turn0search20\[](#)
- - Atlas managed service olduğu için opsiyonlar (sharding, yedekleme) hazır; localda self-host edilecekse daha fazla kaynak gerekebilir. [\[cite\]turn0search20\[](#)

Dezavantajlar:

- - Managed servis olmayan yerel kurulum (örn. Community Server) üzerinde vektör search özellikleri sınırlı; genelde Atlas'a bağımlı. [\[cite\]turn0search20\[](#)
- - 16 GB RAM'de büyük embedding koleksiyonlarında indeksin belleğe sığdırılması karmaşık olabilir, sorgu optimizasyonu gerekli.

16 GB RAM için öneriler / kullanım notları:

- - Filter + vector kombinasyonlarını kullanarak ilk önce daraltma, sonra similarity yaparak RAM kullanımını kontrol et. [\[cite\]turn0search20\[](#)
- - Atlas yerine lokal test için daha küçük embedding alt kümeleriyle başla ve bellek kullanımını izleyerek ölçeklendir.

Kaynaklar:

- - AImultiple comparison updated July 2025
- - Cloudfraft top vector databases 2025 overview

Qdrant

Rust ile yazılmış, disk ve RAM kombinasyonlu, ACID-benzeri güvenilirlik ve filtresi güçlü open-source vektör database. HNSW destekler; memory-efficient modlar sunar.

Avantajlar:

- - Memory-efficient: 1 milyon vektörü 135MB RAM ile disk bazlı yedekleme kullanarak servise alabilir. [\[cite@turn0search19\]](#)
- - Hızlı, Rust tabanlı ve düşük overhead; filtering ve payload desteği güçlü. [\[cite@turn0search14\]](#)
- - Topluluk odaklı benchmark karşılaştırmaları ve optimize edilmiş implementasyon detayları var. [\[cite@turn0search1\]](#)

Dezavantajlar:

- - Bazı gelişmiş dağıtık senaryolar (dynamic sharding) eksik kalabilir; büyük sistemlerde manuel tasarım gerekebilir. [\[cite@turn0search3\]](#)
- - Sıfırdan çok büyük global cluster'lar kurarken Milvus kadar yerleşik yatay ölçeklenebilirlik özelliği algılanabilir şekilde farklı olabilir.

16 GB RAM için öneriler / kullanım notları:

- - 16 GB RAM'li sistemde disk-backed mod ile büyük koleksiyonları düşük RAM ile çalıştır; HNSW parametrelerini RAM'a göre küçült. [\[cite@turn0search19\]](#)
- - Candidate retrieval aşamasında top-k'ı sınırlayarak bellek kullanımını kontrol et.

Kaynaklar:

- - Qdrant official benchmarks page
- - Medium comparison guide Qdrant vs pgvector May 2025
- - Latest vector db landscape analysis Medium (2025)

Milvus

Yüksek ölçeklenebilir, GPU desteği olan açık kaynak vektör veritabanı. IVF, HNSW, PQ gibi çoklu indeksleme stratejileri sunar ve büyük veri setlerinde optimize edilebilir.

Avantajlar:

- - Çeşitli index tipi seçenekleri ve büyük ölçekli dataset'te güçlü hız/doğruluk dengesi. [\[cite?turn0search0\]](#)
- - Hem RAM içi hem disk üstü (yaklaşık spilling) stratejilerle 16GB altında da yönetilebilir; dikkatli index config gerekir. [\[cite?turn0search21\]](#)

Dezavantajlar:

- - Varsayılan kurulumda bellek açlığı olabilir; özellikle büyük embedding'lerde RAM tüketimi yüksek olabilir. [\[cite?turn0search3\]](#)
- - Karmaşık konfigürasyon (index tuning) öğrenme eğrisi yaratabilir.

16 GB RAM için öneriler / kullanım notları:

- - Index'leri IVF+PQ kombinasyonu ile disk kullanımını artırıp RAM baskısını azaltarak çalıştır. [\[cite?turn0search21\]](#)
- - Local 16GB sistemde GPU yoksa CPU modunda HNSW parametreleriyle küçük batch'lerle test et.

Kaynaklar:

- - Shakudo top vector databases July 2025
- - SingleStore ultimate guide vector database landscape Jan 2025

Weaviate

Schema tabanlı, GraphQL destekli vektör veritabanı. Hybrid sorgular (vector + semantic/classical) ve modüler eklentiler sunar.

Avantajlar:

- - Graph özellikleriyle zengin bağlamsal sorgular, schema ile metadata entegrasyonu güçlü. [\[cite?turn0search16\]](#)
- - Plug-in mimarisi ve entegre semantic eklentiler (örn. text2vec-contextionary / custom) ile esnek kullanım. [\[cite?turn0search6\]](#)

Dezavantajlar:

- - Bazı benchmark'larda saf vector-only görevlerde diğerlerine göre geride kalabiliyor; ek graph/semantic katmanlar latency artırabilir. [\[cite?turn0search16\]](#)
- - Schema yönetimi ve başlangıç konfigürasyonu diğerlerinden daha fazla tasarım gerektirebilir.

16 GB RAM için öneriler / kullanım notları:

- - 16 GB'ta performing test edilirken sadece gerekli sınıfları ve vector boyutunu düşür; hybrid sorgularda önce daraltma yap. [\[cite\]turn0search16\[](#)
- - Complex relationship sorgularında caching stratejisi uygula.

Kaynaklar:

- - Medium comparison guide Top 5 Open Source Vector Search Engines 2025
- - Latest Weaviate performance analysis (Medium)

Redis (Redisearch)

Redis, vector search desteğini Redisearch modülü aracılığıyla sağlar. Hızlı in-memory sorgu kabiliyetiyle yüksek throughput ve düşük latency sunar.

Avantajlar:

- - Yüksek throughput ve düşük latency; benchmarklarda yüksek recall ve performans gösteriyor. [\[cite\]turn0search5\[](#)
- - Basit kurulum ve Python client ile entegrasyon kolay. [\[cite\]turn0search12\[](#)

Dezavantajlar:

- - Tamamen RAM bazlı çalıştığında 16 GB sınırında embedding'leri tutmak zor olabilir; evict/eviction ve quantization stratejisi gerekebilir. [\[cite\]turn0search12\[](#)
- - Veri kalıcılığı için ekstra konfigürasyon gerekebilir (snapshotting vs persistence).

16 GB RAM için öneriler / kullanım notları:

- - Düşük boyutlu embed setleri için Redis kullan, gerekirse hybrid ile önce daralt sonra similarity. [\[cite\]turn0search5\[](#)
- - Quantization ve approximate search ayarları ile bellek kullanımını optimize et. [\[cite\]turn0search12\[](#)

Kaynaklar:

- - Redis benchmarking results for vector databases (Redis official blog June 2024)
- - Pgvector vs Redis blog detailed analysis April 2024

4. Özet Karşılaştırma Tablosu

Veritabanı	Bellek Uyumlu? (16GB)	İndeksler	Doğruluk/Hız	Disk Desteği	Entegrasyon	Notlar
PostgreSQL + pgvector	Orta (tuning ile)	HNSW	İyi dengeli	Yok (temel), partition ile	Python, SQL	ACID, tek stack; büyük index'lerde tuning gerekir.
MongoDB Atlas Vector Search	İyi (managed) / Orta (local sınırlı)	Approximate (özelleşmiş)	Esnek	Evet (Atlas)	JSON filtreleme	Attribute+vector birlikte.
Qdrant	Çok iyi (disk-backed düşük RAM)	HNSW, hybrid	Yüksek	Evet	REST/Client	Rust tabanlı performans; configurable.
Milvus	Orta (index tuning ile)	IVF, HNSW, PQ	Yüksek (büyük)	Evet	gRPC/REST	Karmaşık ama güçlü.
Weaviate	İyi (schema optimize)	HNSW + hybrid	Orta-yüksek	Evet	GraphQL/REST	Zengin metadata, latency artabilir.
Redis (Redisearch)	Sınırlı (RAM tabanlı)	HNSW benzeri approx	Çok hızlı but memory bound	Snapshot	Simple clients	In-memory; quantization ile optimize.

5. Tavsiyeler ve Sonuç

16 GB RAM’li tek düğüm senaryosunda en dengeli yaklaşımlar: küçük/orta ölçekli embedding koleksiyonları için Qdrant (disk-backed mod ile) ve PostgreSQL + pgvector (tuning ile) ilk tercih olmalıdır. Redis, düşük latency için candidate retrieval’da hibrit olarak kullanılabilir; Weaviate, ilişkilendirilmiş metadata ve Graph sorguları gerektiğinde değer katar. Milvus, büyük veri setlerinde index tuning yaparak güçlü performans verir ama bellek sınırına dikkat edilmelidir. MongoDB Atlas, managed servis olarak esneklik ve filtreleme avantajı sunar ama local 16 GB ortamda kaynak sınırı nedeniyle dikkatli ölçeklendirme gerekir.

6. Kaynakça

- Shakudo blog: Top 9 Vector Databases July 2025
- Qdrant official benchmarks page
- Tigerdata blog: Pgvector vs Qdrant May 2025
- Reddit discussions on vector databases (r/vectordatabase, r/LocalLLaMA)
- Cloudfraft blog: Top 5 Vector Databases 2025
- Redis official blog: Benchmarking results for vector databases June 2024
- Medium comparison articles (Weaviate vs Milvus vs Qdrant etc., 2024-2025)
- SingleStore ultimate guide to vector database landscape Jan 2025
- Aimultiple comparison updated July 2025
- Tigerdata blog: Why Postgres Wins for AI and Vector Workloads Apr 2025