

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN



TIỂU LUẬN CHƯƠNG III

Môn học: DỮ LIỆU LỚN

Đề tài: Tìm hiểu phần mềm GridGain trong hệ sinh thái Dữ liệu lớn

Giảng viên: ThS. Nguyễn Hồ Duy Trí

Lớp: Dữ liệu lớn – IS405.011

Sinh viên thực hiện:

Nguyễn Hiền Đức	-	20520450
Nguyễn Bảo Anh	-	20521068
Trương Vĩnh Thái	-	19520940
Lý Sô Ly	-	19521136

TP. Hồ Chí Minh, Thứ Ba, 07 Tháng Mười Một 2023

MỤC LỤC

MỤC LỤC	2
LỜI CẢM ƠN.....	4
NHẬN XÉT CỦA GIẢNG VIÊN.....	5
DANH MỤC HÌNH ẢNH	6
DANH MỤC BẢNG	8
CHƯƠNG 1: TỔNG QUAN VỀ GRIDGAIN.....	9
1. GridGain là gì?.....	9
2. Các dịch vụ cung cấp.....	9
3. Kiến trúc của GridGain	10
4. Yêu cầu hệ thống.....	12
CHƯƠNG 2: ĐẶC ĐIỂM CỦA GRIDGAIN	14
1. Các tính năng nổi bật của GridGain	14
1.1. Lưới dữ liệu trong bộ nhớ (In-Memory Data Grid)	14
1.2. Cơ sở dữ liệu trong bộ nhớ (In-Memory Database).....	15
1.3. Lưới tính toán trong bộ nhớ (In-Memory Compute Grid)	16
1.4. Mạng lưới dịch vụ trong bộ nhớ (In-Memory Service Grid)	18
1.5. Xử lý các luồng dữ liệu trong bộ nhớ (In-Memory Streaming)	18
1.6. Tăng tốc Hadoop trong bộ nhớ (In-Memory Hadoop Acceleration)	20
1.7. Kiến trúc phân tán.....	21
1.8. Đồng nhất API	23
2. Ưu và nhược điểm của GridGain.....	24
2.1. Ưu điểm	24
2.2. Nhược điểm	25
3. So sánh	26
3.1. Về hiệu suất.....	26
3.2. Về những sản phẩm tương tự	27
CHƯƠNG 3: ỨNG DỤNG CỦA GRIDGAIN	33

1. Tăng tốc Data Lake Hadoop.	33
2. Các dịch vụ tài chính.	34
3. Thương mại điện tử và bán lẻ.	34
4. Viễn thông.	34
5. Chăm sóc sức khỏe.	35
6. Năng lượng và tiện ích.	35
7. Internet of Things (IoT).	35
<i>CHƯƠNG 4: CÀI ĐẶT GRIDGAIN.</i>	<i>36</i>
<i>CHƯƠNG 5: MINH HỌA XỬ LÝ DỮ LIỆU LỚN TRÊN GRIDGAIN.</i>	<i>41</i>
<i>TÀI LIỆU THAM KHẢO.</i>	<i>48</i>

LỜI CẢM ƠN

Đầu tiên và quan trọng nhất, nhóm chúng em xin gửi lời cảm ơn sâu sắc nhất tới thầy ThS. Nguyễn Hồ Duy Trí đã hướng dẫn nhóm trong quá trình thực hiện bài tiểu luận chương III này. Thầy đã hướng dẫn, cung cấp kiến thức và tận tâm trong suốt thời gian thực hiện đề tài này. Sự chỉ dẫn và phản hồi từ giảng viên đã giúp nhóm hoàn thiện tiểu luận này một cách tốt nhất. Một lần nữa nhóm em xin chân thành cảm ơn thầy rất nhiều, chúc thầy dồi dào sức khỏe.

Trong quá trình thực hiện bài tiểu luận, nhóm chúng em vận dụng kiến thức nền tảng đã tích lũy để hiểu về các khái niệm và kỹ thuật cơ bản về dữ liệu lớn, hệ thống phân tán, và quản lý dữ liệu là cơ sở quan trọng để tiến hành nghiên cứu về phần mềm GridGain. Đồng thời, việc học hỏi và nghiên cứu các kiến thức mới từ thầy, bạn bè và các nguồn tham khảo đa dạng đã cung cấp cho chúng em cái nhìn sâu hơn về đề tài này. Từ đó, nhóm đã tiếp nhận được nhiều kiến thức bổ ích để hoàn thành bài tiểu luận một cách hiệu quả nhất.

Tuy nhiên, nhóm chúng em vẫn nhận thấy rằng mình còn nhiều thiếu sót. Nhóm mong muốn nhận được sự góp ý, chỉ bảo thêm từ quý thầy để hoàn thiện những kiến thức còn thiếu sót. Chúng em tin rằng, với những lời góp ý từ quý thầy chúng em sẽ có thể hoàn thiện bài tiểu luận một cách tốt hơn, đồng thời cũng giúp chúng em có thêm kiến thức và kinh nghiệm để thực hiện các đề tài khác trong tương lai.

Chúng em hy vọng rằng tiểu luận này có thể là một bước đệm quan trọng trong việc khám phá và hiểu sâu hơn về phần mềm GridGain. Kiến thức chúng em thu thập được trong quá trình này sẽ chúng em trong việc hoàn thành các tiểu luận và hỗ trợ chúng em trong học tập và làm việc sau này. Chúng em xin chân thành cảm ơn.

Trân trọng!

Thành phố Hồ Chí Minh, Tháng Mười Một 23

Nhóm sinh viên thực hiện

NHẬN XÉT CỦA GIẢNG VIÊN

DANH MỤC HÌNH ẢNH

Hình 1. Kiến trúc của GridGain [2]	11
Hình 2. Minh họa đặc điểm của In-Memory Data Grid	14
Hình 3. Minh họa đặc điểm của In-Memory SQL Grid.....	16
Hình 4. Minh họa đặc điểm của In-Memory Compute Grid.....	17
Hình 5. Minh họa đặc điểm của In-Memory Service Grid.....	18
Hình 6. Minh họa đặc điểm của In-Memory Streaming.....	19
Hình 7. Minh họa đặc điểm của In-Memory Hadoop Acceleration	20
Hình 8. Một số giao thức hỗ trợ trên nền tảng GridGain	24
Hình 9. Kiến trúc của Apache Ignite [2]	27
Hình 10. Kiến trúc Hazelcast Cluster [6].....	27
Hình 11. Kiến trúc của Apache Geode [7].....	28
Hình 12. Sơ lược về Redis.....	29
Hình 13. Kiến trúc Memcached.....	29
Hình 14. Kiến trúc của một Flink	30
Hình 15. Kiến trúc của Apache Kafka Streams.....	31
Hình 16. Kiến trúc của Apache Samza	31
Hình 17. Kiến trúc của Apache Storm	32
Hình 18. Ứng dụng của GridGain trên Data Lake Hadoop	33
Hình 19. Một số ứng dụng của GridGain trong thực tiễn [8]	35
Hình 20. Tải xuống phần mềm GridGain phiên bản Community Edition.....	36
Hình 21. Tập tin nén gridgain-community-8.9.0.zip.....	37
Hình 22. Các thư mục hiển thị sau khi giải nén tập tin gridgain-community-8.9.0.zip.....	37
Hình 23. Di chuyển thư mục ignite-rest-http đến đường dẫn {gridgain}/libs.....	38
Hình 24. Mở tệp .bashrc.....	39
Hình 25. Thêm dòng lệnh tạo biến môi trường.....	39

Hình 26. Tạo biến môi trường thành công.....	40
Hình 27. Tạo project mới trên IntelliJ IDEA	41
Hình 28. Nhập tên project mới và chọn Build System tương ứng là Maven.....	42
Hình 29. Viết đoạn mã trên để xử lý yêu cầu	46
Hình 30. Chọn Project Structure để thêm đường dẫn đến các thư viện.....	46
Hình 31. Thêm các thư viện cần thiết để thực thi.....	46
Hình 32. Chạy file <code>ComputeTaskExample.java</code>	47
Hình 33. Kết quả sau khi thực thi chương trình.....	47

DANH MỤC BẢNG

Bảng 1. Bảng so sánh hiệu suất của một số sản phẩm trên nền tảng dữ liệu lớn	26
--	----



CHƯƠNG 1: TỔNG QUAN VỀ GRIDGAIN

1. GridGain là gì?

GridGain là một nền tảng điện toán trong bộ nhớ mang lại tốc độ chưa từng có và khả năng mở rộng quy mô lớn cho hoạt động xử lý dữ liệu hiện đại. GridGain có khả năng lưu trữ và xử lý dữ liệu ngay trong RAM trên một cụm máy được kết nối với nhau. Được xây dựng trên dự án nguồn mở Apache Ignite cho phép giao dịch hiệu suất cao, phát trực tuyến theo thời gian thực và phân tích nhanh trong một lớp xử lý và truy cập dữ liệu toàn diện, duy nhất.

GridGain dễ dàng hỗ trợ cả ứng dụng hiện có và ứng dụng mới trong kiến trúc song song, phân tán trên diện rộng trên phần cứng tiêu chuẩn với giá cả phải chăng. GridGain có thể trên máy nội bộ, trong môi trường kết hợp hoặc trên nền tảng đám mây như AWS, Microsoft Azure hoặc Google Cloud Platform. [1]

2. Các dịch vụ cung cấp

GridGain cung cấp một API hợp nhất hỗ trợ SQL, C++, .NET, Java/Scala/Groovy, Node.js và nhiều quyền truy cập hơn cho lớp ứng dụng. API hợp nhất kết nối các ứng dụng quy mô đám mây với nhiều kho dữ liệu có chứa dữ liệu có cấu trúc, bán cấu trúc và không cấu trúc (SQL, NoSQL, Hadoop).

GridGain cung cấp dữ liệu hiệu suất cao môi trường cho phép các công ty xử lý các giao dịch ACID và tạo ra những hiểu biết có giá trị bằng cách sử dụng ANSI-99 SQL từ các truy vấn thời gian thực, tương tác và hàng loạt. Nền tảng điện toán trong bộ nhớ cung cấp một cách tiếp cận chiến lược cho điện toán trong bộ nhớ.

GridGain cung cấp hiệu suất, quy mô và các khả năng toàn diện bao gồm các khả năng của cơ sở dữ liệu trong bộ nhớ (IMDB), lưới dữ liệu (View, Table,...) trong bộ nhớ (IMDG), công cụ phân tích luồng trực tuyến (Stream Analytics) và các giải pháp điểm dựa trên bộ nhớ khác trong một giải pháp.

Không giống như cơ sở dữ liệu trong bộ nhớ, GridGain có thể hoạt động trên cơ sở dữ liệu hiện có và không yêu cầu sao chép và thay thế hoặc bất kỳ thay đổi nào đối với RDBMS hiện có. Người dùng có thể giữ nguyên RDBMS hiện có của mình và triển khai GridGain dưới dạng một lớp phía trên nó. GridGain thậm chí có thể tự động tích hợp với các hệ thống RDBMS khác nhau như: Oracle, MySQL, Postgres, DB2, Microsoft SQL Server,... Tính năng tích hợp tự động này tạo ra mô hình miền ứng dụng dựa trên định nghĩa lược đồ của cơ sở dữ liệu cơ bản và sau đó tải dữ liệu.

Hơn nữa, IMDB thường chỉ cung cấp giao diện SQL trong khi GridGain cung cấp một hệ sinh thái rộng hơn nhiều gồm các mô hình xử lý và truy cập được hỗ trợ ngoài ANSI SQL. GridGain hỗ trợ lưu trữ khóa/giá trị, truy cập SQL, MapReduce, xử lý HPC/MPP, xử lý phát trực tuyến/CEP và tăng tốc Hadoop, nền tảng điện toán trong bộ nhớ được tích hợp tốt tất cả trong một.

Khi so sánh GridGain với lưới dữ liệu trong bộ nhớ, cần lưu ý rằng lưới dữ liệu trong bộ nhớ chỉ là một trong những khả năng mà GridGain cung cấp. Ngoài chức năng lưới dữ liệu, GridGain còn hỗ trợ xử lý HPC/MPP, SQL phân tán, phát trực tuyến, phân cụm và tăng tốc Hadoop, cho phép phạm vi sử dụng rộng hơn cho nhiều trường hợp hơn một IMDG điển hình.

GridGain cũng có thể được triển khai dưới dạng cơ sở dữ liệu trong bộ nhớ có tính giao dịch và phân tán. Khi được sử dụng làm IMDB, giải pháp cung cấp đảm bảo giao dịch ACID tuân thủ ANSI-99 SQL bao gồm hỗ trợ DDL và DML. Tính năng Persistent Store tùy chọn giải quyết các vấn đề tràn bộ nhớ và cho phép khởi động lại nhanh vì GridGain có thể xử lý dữ liệu cả trong bộ nhớ và trên đĩa trong khi dữ liệu đang được tải vào bộ nhớ trong quá trình khởi động. [1]

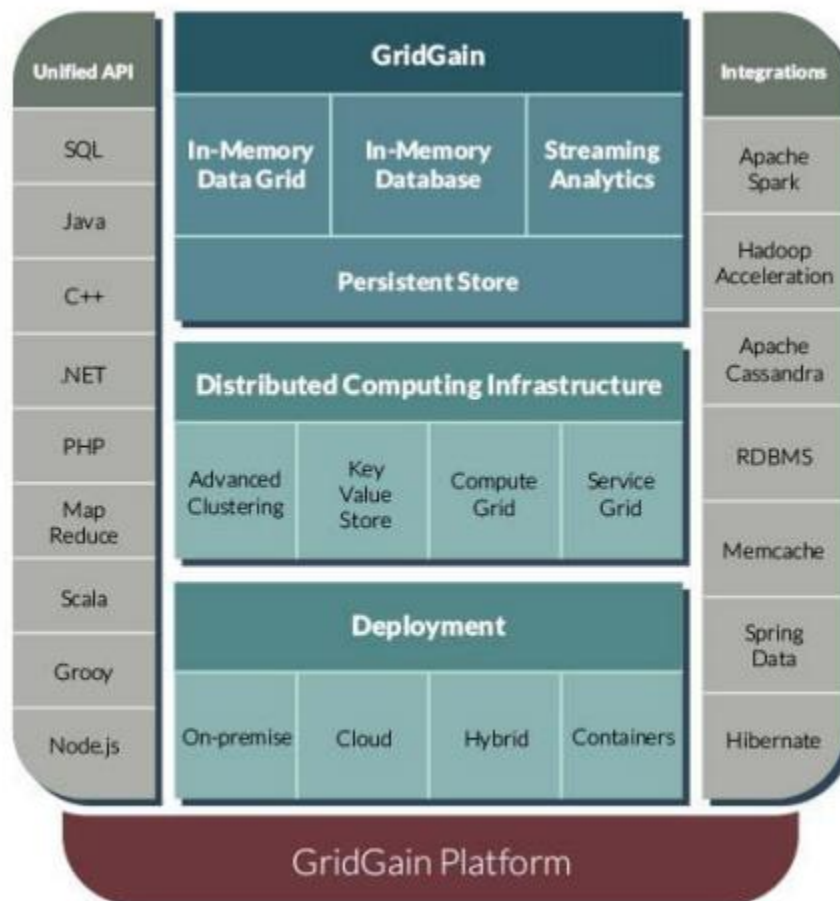
3. Kiến trúc của GridGain

GridGain là phần mềm trung gian phân tán dựa trên JVM. Chúng dựa trên việc triển khai cấu trúc liên kết cụm đồng nhất không yêu cầu các nút máy chủ và máy khách

riêng biệt. Tất cả các nút trong cụm GridGain đều bình đẳng và có thể đóng bất kỳ vai trò logic nào theo yêu cầu của ứng dụng trong thời gian chạy (peer-to-peer).

Cốt lõi của GridGain là thiết kế Giao diện nhà cung cấp dịch vụ (SPI). Thiết kế dựa trên SPI giúp mọi thành phần bên trong của GridGain có thể tùy chỉnh hoàn toàn bởi nhà phát triển. Điều này cho phép cấu hình hệ thống rất tốt, với khả năng thích ứng với mọi cơ sở hạ tầng máy chủ hiện tại hoặc tương lai.

Một nguyên lý cốt lõi khác của GridGain là hỗ trợ trực tiếp cho việc song song hóa các tính toán phân tán dựa trên xử lý kiểu Fork/Join, MapReduce hoặc MPP, hệ sinh thái triển khai lớn nhất của các thuật toán xử lý phân tán. GridGain sử dụng rộng rãi các tính toán song song phân tán trong nội bộ và chúng được hiển thị đầy đủ ở phân cấp API cho chức năng do người dùng xác định. [1]



Hình 1. Kiến trúc của GridGain [2]

Kiến trúc GridGain trong Hình 1 bao gồm các thành phần sau:

- **Unified API:** cung cấp một API thống nhất cho các nhà phát triển sử dụng các ngôn ngữ lập trình khác nhau, bao gồm Java, C++, .NET, PHP, Scala, Groovy, và Node.js. API này bao gồm các lớp và phương thức cho các tác vụ phổ biến, chẳng hạn như truy cập dữ liệu, thực thi các tác vụ, và quản lý cụm.
- **Integrations:** tích hợp với các công nghệ khác, các tích hợp này bao gồm:
 - SQL: cung cấp khả năng truy cập dữ liệu trong bộ nhớ bằng SQL.
 - MapReduce: hỗ trợ mô hình MapReduce cho phép phân tích dữ liệu lớn.
 - Apache Spark: cung cấp khả năng xử lý dữ liệu lớn theo thời gian thực.
 - Apache Cassandra: để cung cấp khả năng lưu trữ dữ liệu phi cấu trúc.
 - RDBMS: GridGain có thể truy cập dữ liệu từ các hệ thống cơ sở dữ liệu quan hệ (RDBMS) như MySQL, PostgreSQL, và Oracle.
- **In-Memory Data Grid:** cung cấp một bộ nhớ dữ liệu trong bộ nhớ cho phép truy cập dữ liệu nhanh chóng và hiệu quả, có thể được sử dụng để lưu trữ dữ liệu tĩnh, dữ liệu động, hoặc dữ liệu kết quả của các phép tính.
- **Distributed Computing Infrastructure:** cung cấp cơ sở hạ tầng tính toán phân tán cho phép các ứng dụng chạy trên nhiều máy, tận dụng sức mạnh của nhiều máy tính để gia tăng hiệu suất.
- **Deployment:** có thể được triển khai trên đám mây, kết hợp, hoặc tại chỗ, phù hợp với nhu cầu của từng doanh nghiệp. [3]

4. Yêu cầu hệ thống

Hệ thống cần cài đặt các phần mềm như sau: [2]

Phần mềm cần cài đặt	Phiên bản tối thiểu
JDK	Oracle JDK 8, 11 hoặc 17, Open JDK 8, 11 hoặc 17, IBM JDK 8, 11 hoặc 17

Hệ điều hành	Linux (bất kỳ phiên bản nào) , Mac OSX (10.6 trở lên), Windows (XP trở lên), Windows Server (2008 trở lên), Oracle Solaris, z/OS
ISA	x86, x64, SPARC, PowerPC
Kết nối mạng	Không hạn chế (khuyến nghị 10G)



CHƯƠNG 2: ĐẶC ĐIỂM CỦA GRIDGAIN

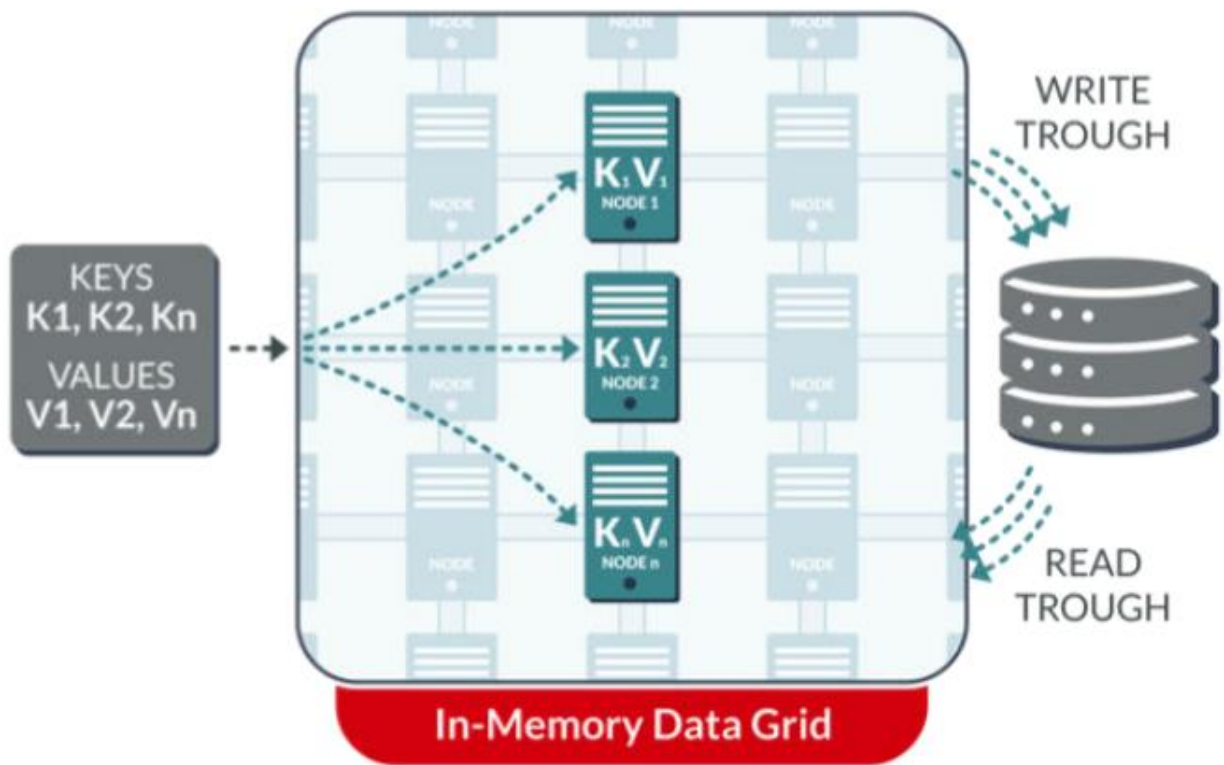
1. Các tính năng nổi bật của GridGain

1.1. Lưới dữ liệu trong bộ nhớ (In-Memory Data Grid)

Một trong những khả năng cốt lõi của GridGain là lưới dữ liệu trong bộ nhớ. Lưới dữ liệu xử lý việc quản lý dữ liệu trong bộ nhớ phân tán bao gồm các giao dịch ACID, chuyển đổi dự phòng và cân bằng tải nâng cao, ANSI-99 SQL bao gồm hỗ trợ DDL và DML cùng nhiều tính năng khác.

GridGain IMDG là một kho lưu trữ key-value trong bộ nhớ, dựa trên đối tượng, giao dịch ACID. GridGain lưu trữ dữ liệu trong bộ nhớ, trái ngược với các hệ thống quản lý cơ sở dữ liệu truyền thống sử dụng đĩa làm cơ chế lưu trữ chính.

Bằng cách sử dụng bộ nhớ hệ thống thay vì ổ đĩa, GridGain nhanh hơn rất nhiều so với các hệ thống DBMS truyền thống. [1]



Hình 2. Minh họa đặc điểm của In-Memory Data Grid

Những điểm nổi bật và khả năng chính:

- Truy vấn ANSI SQL-99 với các phép nối phân tán.
- Hiệu suất cực nhanh.
- Bộ nhớ cache trong bộ nhớ phân tán.
- Khả năng mở rộng linh hoạt để xử lý lên đến petabyte dữ liệu trong bộ nhớ.
- Các ACID transactions trong bộ nhớ được phân tán.
- Hàng đợi trong bộ nhớ phân tán và các cấu trúc dữ liệu khác.
- Phân cụm web theo các phiên (session).
- Tích hợp bộ nhớ đệm Hibernate L2.
- Lưu trữ ngoài heap theo tầng.
- Triển khai giao dịch không bế tắc duy nhất (Unique deadlock-free transactions) để có tốc độ xử lý giao dịch trong bộ nhớ nhanh nhất.

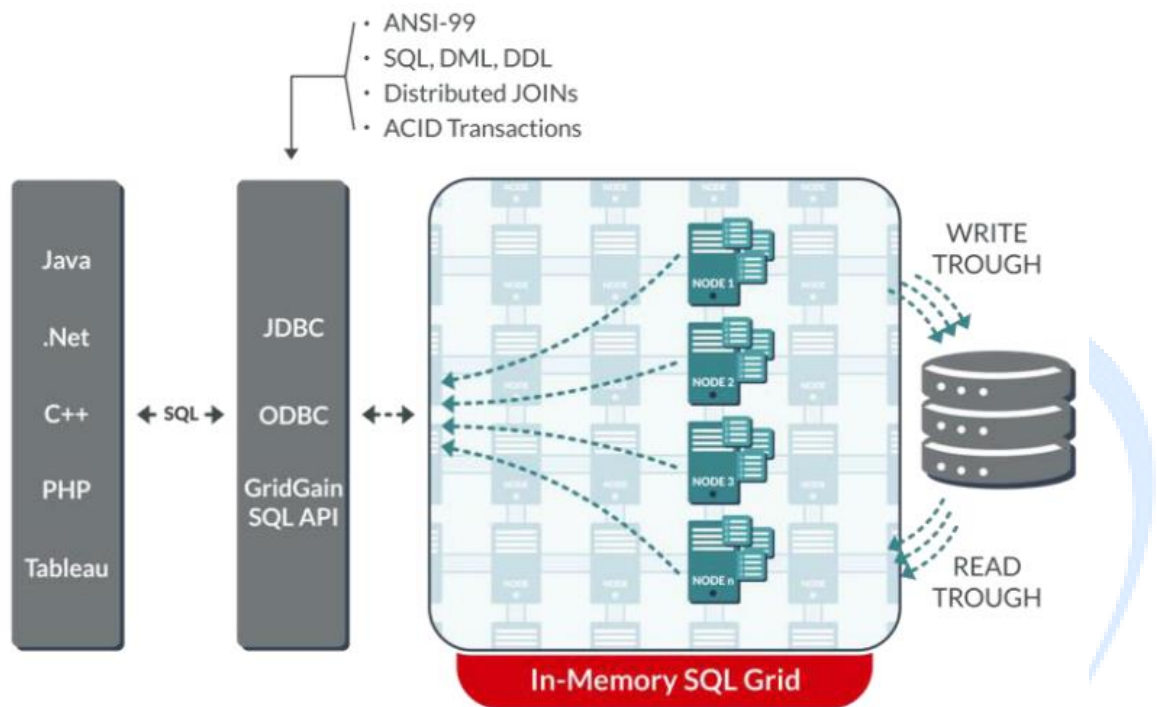
1.2. Cơ sở dữ liệu trong bộ nhớ (In-Memory Database)

Cơ sở dữ liệu trong bộ nhớ GridGain tận dụng các khả năng SQL được phân tán trên hệ thống. Chúng có khả năng mở rộng theo chiều ngang, có khả năng chịu lỗi và tuân thủ ANSI-99 SQL. Nó cũng hỗ trợ tất cả các lệnh SQL, DDL và DML bao gồm các truy vấn SELECT, UPDATE, INSERT, MERGE và DELETE cũng như bảng CREATE và DROP.

Cú pháp SQL tuân thủ ANSI SQL-99. GridGain có thể sử dụng bất kỳ hàm SQL, tập hợp hoặc nhóm nào. GridGain hỗ trợ các phép nối SQL phân tán và cho phép nối nhiều bộ đệm. Việc kết hợp giữa các bộ nhớ đệm được phân vùng và sao chép hoạt động không có giới hạn trong khi việc kết hợp giữa các tập dữ liệu được phân vùng yêu cầu các khóa phải được sắp xếp thứ tự. GridGain cũng hỗ trợ khái niệm truy vấn trường để giúp giảm thiểu chi phí mạng và tuần tự hóa.

Các khả năng SQL phân tán trong bộ nhớ cho phép người dùng tương tác với nền tảng GridGain không chỉ bằng cách sử dụng các API được phát triển nguyên bản cho Java, .NET và C++ mà còn sử dụng các lệnh SQL tiêu chuẩn thông qua API GridGain

JDBC hoặc ODBC. Điều này cung cấp kết nối đa nền tảng thực sự từ các ngôn ngữ như PHP, Ruby,... [1]



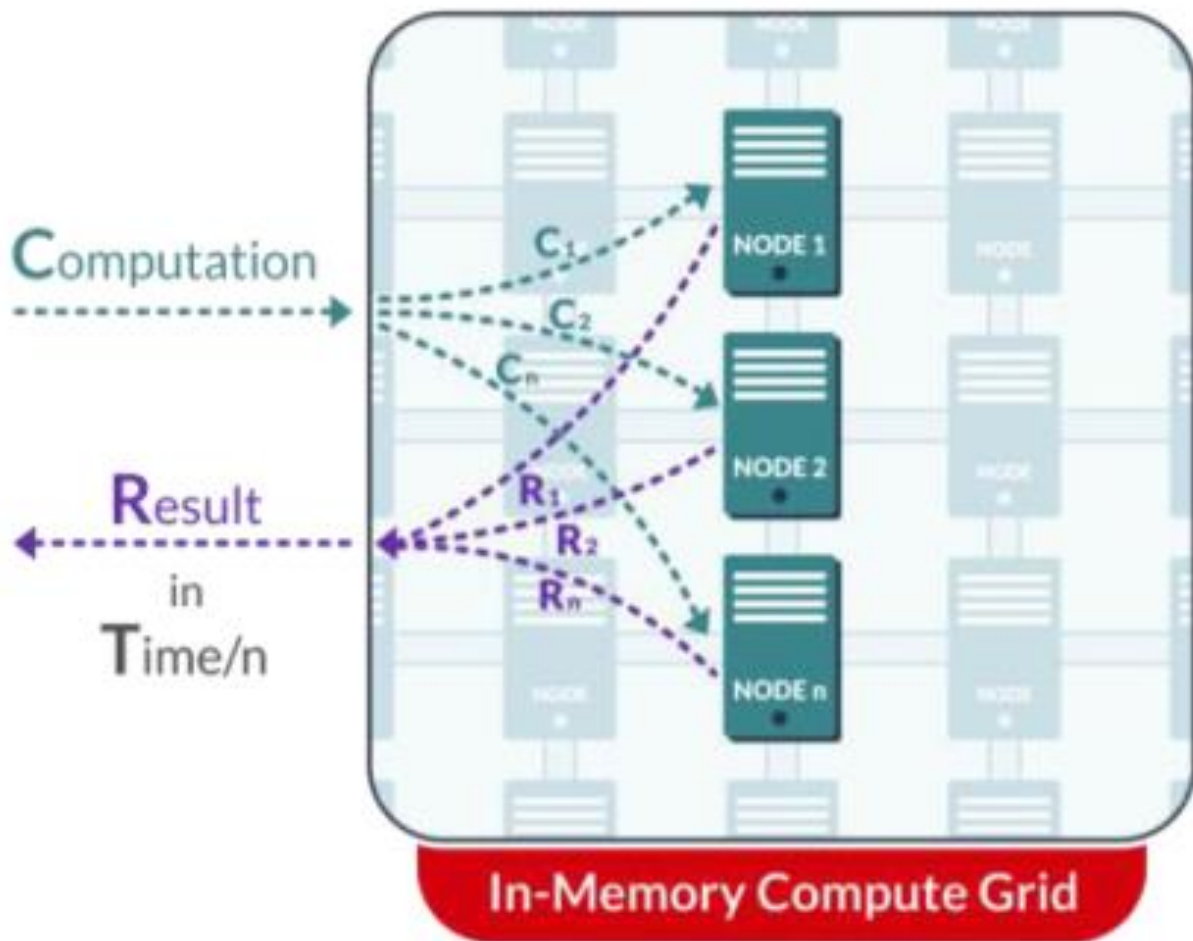
Hình 3. Minh họa đặc điểm của In-Memory SQL Grid

Những điểm nổi bật và khả năng chính:

- Tuân thủ ANSI.
- Hỗ trợ các lệnh SQL và DML: SELECT, UPDATE, INSERT, MERGE và DELETE.
- Hỗ trợ các lệnh DDL bao gồm bảng CREATE và DROP.
- SQL phân tán.
- Giao tiếp SQL qua API GridGain JDBC và ODBC không cần mã hóa tùy chỉnh.
- Hỗ trợ không gian địa lý.

1.3.Lưới tính toán trong bộ nhớ (In-Memory Compute Grid)

GridGain tích hợp một mạng lưới tính toán cho phép xử lý song song, trong bộ nhớ của các tác vụ đòi hỏi nhiều tài nguyên, bao gồm cả tính toán hiệu suất cao truyền thống (HPC) và xử lý song song hàng loạt (MPP). [1]



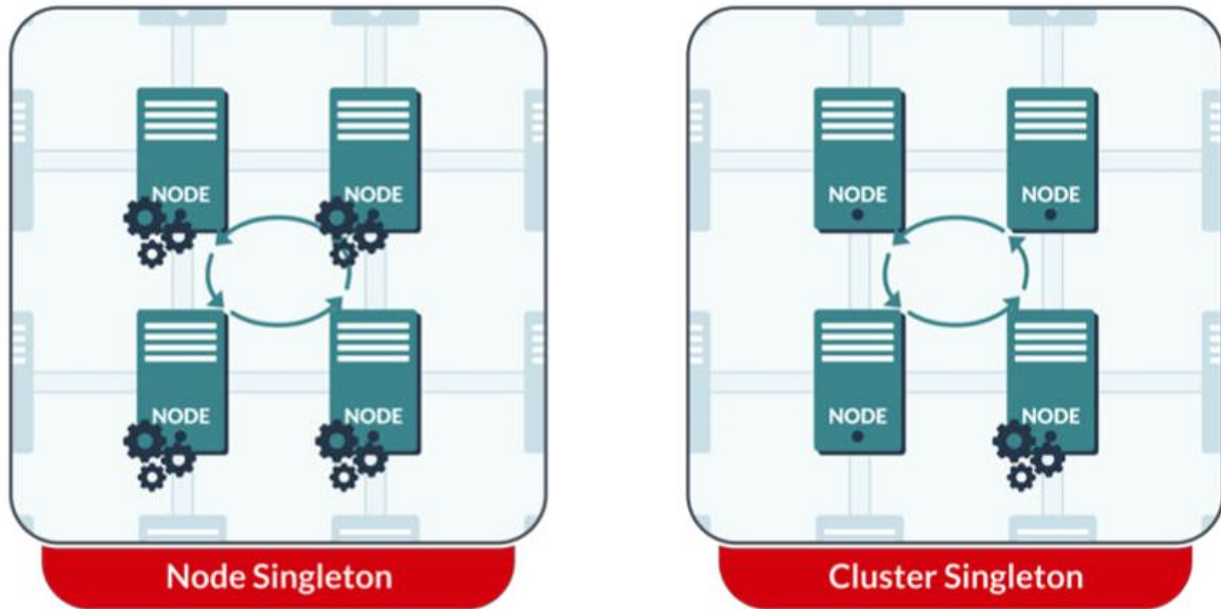
Hình 4. Minh họa đặc điểm của In-Memory Compute Grid

Những khả năng chính bao gồm:

- Phân cụm động.
- Xử lý Fork-Join và MapReduce.
- Thực thi đóng gói phân tán.
- Cân bằng tải và khả năng chống lỗi.
- Gửi thông điệp và sự kiện phân tán.
- Tính mở rộng tuyến tính.
- Hỗ trợ Standard Java ExecutorService.

1.4. Mạng lưới dịch vụ trong bộ nhớ (In-Memory Service Grid)

Lưới Dịch Vụ GridGain cung cấp cho người dùng sự kiểm soát hoàn toàn đối với các dịch vụ được triển khai trên cụm. Nó cho phép người dùng kiểm soát số lượng các phiên bản của dịch vụ của họ nên được triển khai trên mỗi nút của cụm, đảm bảo triển khai đúng và khả năng chịu lỗi. Mạng lưới dịch vụ này đảm bảo tính khả dụng liên tục của tất cả các dịch vụ được triển khai trong trường hợp lỗi nút. [1]



Hình 5. Minh họa đặc điểm của In-Memory Service Grid

Những khả năng chính bao gồm:

- Triển khai tự động nhiều phiên bản của một dịch vụ.
- Triển khai tự động dịch vụ dưới dạng duy nhất (singleton).
- Triển khai tự động các dịch vụ khi nút khởi động.
- Triển khai khả năng chống lỗi.
- Xóa các dịch vụ đã triển khai.
- Truy xuất thông tin cấu trúc liên kết dịch vụ
- Truy cập từ xa đến các dịch vụ đã triển khai thông qua dịch vụ proxy.

1.5. Xử lý các luồng dữ liệu trong bộ nhớ (In-Memory Streaming)

Xử lý các luồng dữ liệu trong bộ nhớ đáp ứng một loạt ứng dụng lớn mà phương pháp xử lý truyền thống và lưu trữ dựa trên ổ đĩa, chẳng hạn như cơ sở dữ liệu dựa

trên ổ đĩa hoặc hệ thống tệp, không đủ để xử lý. Những ứng dụng như vậy đang mở rộng giới hạn của cơ sở hạ tầng xử lý dữ liệu truyền thống.

Hỗ trợ luồng dữ liệu cho phép truy vấn các cửa sổ trượt của dữ liệu đang đến để người dùng có thể trả lời những câu hỏi như "10 sản phẩm phổ biến nhất trong 2 giờ qua là những sản phẩm nào?" hoặc "Giá trung bình của sản phẩm thuộc một danh mục cụ thể trong ngày qua là bao nhiêu?".

Một trường hợp sử dụng phổ biến khác cho việc xử lý luồng là kiểm soát và xây dựng đường ống công việc cho sự kiện phân tán một cách chính xác. Khi các sự kiện đến hệ thống với tốc độ cao, việc xử lý sự kiện được chia thành nhiều giai đoạn và mỗi giai đoạn phải được định tuyến đúng cách trong một cụm để xử lý. [1]



Hình 6. Minh họa đặc điểm của In-Memory Streaming

Những khả năng chính bao gồm:

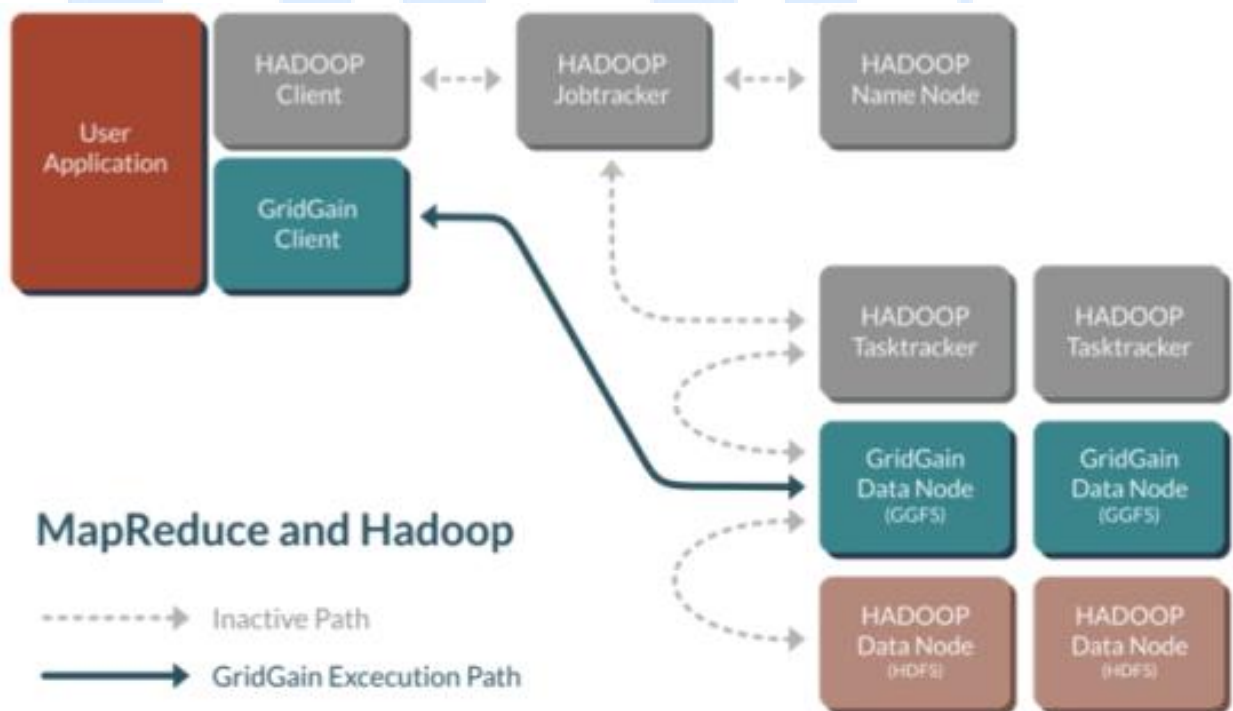
- Truy vấn dựa trên cửa sổ theo cách lập trình.
- Luồng công việc sự kiện tùy chỉnh / Xử lý sự kiện phức tạp (CEP).
- Đảm bảo ít nhất một lần (at-least-once guarantee).
- Cửa sổ trượt tích hợp sẵn và do người dùng xác định.
- Chỉ mục luồng dữ liệu.
- Truy vấn luồng phân tán.
- Đặt chung với hệ thống lưới dữ liệu trong bộ nhớ.

1.6. Tăng tốc Hadoop trong bộ nhớ (In-Memory Hadoop Acceleration)

Bộ gia tốc GridGain cho Hadoop tăng cường môi trường Hadoop hiện có bằng cách cho phép xử lý dữ liệu nhanh chóng bằng cách sử dụng các công cụ và công nghệ mà các tổ chức đã sử dụng hiện nay.

Việc gia tốc Hadoop Trong Bộ Nhớ trong GridGain dựa trên hệ thống tệp trong bộ nhớ với hiệu suất cao hai chế độ đầu tiên trong ngành, hoàn toàn tương thích với Hadoop HDFS, cùng với việc tối ưu hóa triển khai MapReduce trong bộ nhớ. HDFS trong bộ nhớ và MapReduce trong bộ nhớ cung cấp các phần mở rộng dễ sử dụng cho HDFS dựa trên đĩa và MapReduce truyền thống, mang lại hiệu suất nhanh hơn lên đến 100 lần.

Tính năng này có thể cài và chạy mà không yêu cầu tích hợp tối thiểu hoặc không yêu cầu tích hợp gì cả. Nó hoạt động với phiên bản Hadoop mã nguồn mở hoặc bất kỳ phiên bản thương mại nào của Hadoop, bao gồm Cloudera, HortonWorks, MapR, Apache, Intel, AWS, cũng như bất kỳ phiên bản phân phối Hadoop 1.x hoặc Hadoop 2.x nào khác. [1]



Hình 7. Minh họa đặc điểm của In-Memory Hadoop Acceleration

1.7. Kiến trúc phân tán

- *Hệ thống tập tin trong bộ nhớ phân tán (Distributed In-Memory File System)*

GridGain hỗ trợ giao diện hệ thống tập với dữ liệu trong bộ nhớ của nó được gọi là Ignite File System (IGFS). IGFS cung cấp chức năng tương tự như Hadoop HDFS, bao gồm khả năng tạo một hệ thống tập đầy đủ chức năng trong bộ nhớ. IGFS là cốt lõi của Bộ tăng tốc trong bộ nhớ GridGain cho Hadoop.

Dữ liệu từ mỗi tập được chia thành các khối dữ liệu riêng biệt và được lưu trong bộ nhớ đệm. Có thể truy cập dữ liệu trong mỗi tập bằng luồng Java API tiêu chuẩn. Đối với mỗi phần của tập, có thể tính toán mối quan hệ và xử lý nội dung của tập trên các nút tương ứng để tránh kết nối mạng không cần thiết.

Các tính năng chính của hệ thống tập tin trong bộ nhớ phân tán bao gồm:

- Chế độ xem hệ thống tập tiêu chuẩn trên dữ liệu trong bộ nhớ.
- Liệt kê các thư mục hoặc thông tin cho một đường dẫn.
- Tạo/di chuyển/xóa tập tin hoặc thư mục.
- Ghi/đọc luồng dữ liệu vào/từ tập tin. [1]

- *Phân cụm nâng cao (Advance Clustering)*

Nền tảng điện toán trong bộ nhớ GridGain cung cấp một trong những công nghệ phân cụm phức tạp nhất trên máy ảo Java (JVM). Với GridGain, các nút có thể tự động tìm thấy lẫn nhau, giúp mở rộng quy mô cụm khi cần mà không cần phải khởi động lại toàn bộ cụm. Các nhà phát triển cũng có thể tận dụng sự hỗ trợ của đám mây lai trong GridGain, cho phép người dùng thiết lập kết nối giữa đám mây riêng và đám mây công cộng như Amazon Web Services hoặc Microsoft Azure.

Các tính năng chính của phân cụm nâng cao trong GridGain bao gồm:

- Quản lý cấu trúc liên kết động.
- Tự động phát hiện trên mạng LAN, WAN và đám mây công khai.
- Tự động “chia nhỏ” (phân đoạn mạng) .

- Trao đổi tin nhắn unicast, broadcast và dựa trên trao đổi truyền tin giữa các nhóm.
- Triển khai theo yêu cầu và trực tiếp.
- Hỗ trợ các cụm ảo và nhóm nút. [1]
- *Truyền tin phân tán (Distributed Messaging)*

GridGain cung cấp chức năng truyền tin trên toàn cụm, hiệu suất cao để trao đổi dữ liệu thông qua các mô hình liên lạc *publish-subscribe* và trực tiếp *point-to-point*.

Các khả năng chính của truyền tin phân tán bao gồm:

- Hỗ trợ mô hình đăng ký xuất bản theo chủ đề.
- Hỗ trợ phương thức truyền point-to-point trực tiếp.
- Lớp vận chuyển truyền thông có thể tùy chỉnh được.
- Hỗ trợ sắp xếp truyền tin.
- Tự động triển khai trình nghe tin nhắn nhận biết cụm. [1]
- *Sự kiện phân tán (Distributed Events)*

Chức năng sự kiện phân tán trong GridGain cho phép ứng dụng nhận thông báo về các sự kiện bộ đệm xảy ra trong môi trường lưới phân tán. Nhà phát triển có thể sử dụng chức năng này để được thông báo về việc thực thi các tác vụ từ xa hoặc bất kỳ thay đổi nào về dữ liệu bộ đệm trong cụm.

Trong GridGain, các thông báo sự kiện có thể được nhóm lại với nhau và gửi theo đợt và/hoặc theo khoảng thời gian. Thông báo theo đợt giúp đạt được hiệu suất bộ nhớ đệm cao và độ trễ thấp.

Các tính năng chính của sự kiện phân tán trong GridGain bao gồm:

- Đăng ký người nghe cục bộ và từ xa.
- Khả năng kích hoạt và vô hiệu hóa bất kỳ sự kiện nào.
- Bộ lọc cục bộ và từ xa để kiểm soát chi tiết các thông báo.
- Tự động phân nhóm thông báo để nâng cao hiệu suất. [1]

- *Cấu trúc dữ liệu phân tán (Distributed Data Structures)*

GridGain cho phép sử dụng hầu hết các cấu trúc dữ liệu từ framework `java.util.concurrent` theo cấu trúc phân tán.

Ví dụ: Sử dụng `java.util.concurrent.BlockingDeque` và thêm vào trên một nút và thăm dò nó từ một nút khác. Hoặc bạn có thể có một trình tạo khóa chính được phân tán, đảm bảo tính duy nhất trên tất cả các nút.

Cấu trúc dữ liệu phân tán trong GridGain bao gồm hỗ trợ cho các API Java tiêu chuẩn như sau:

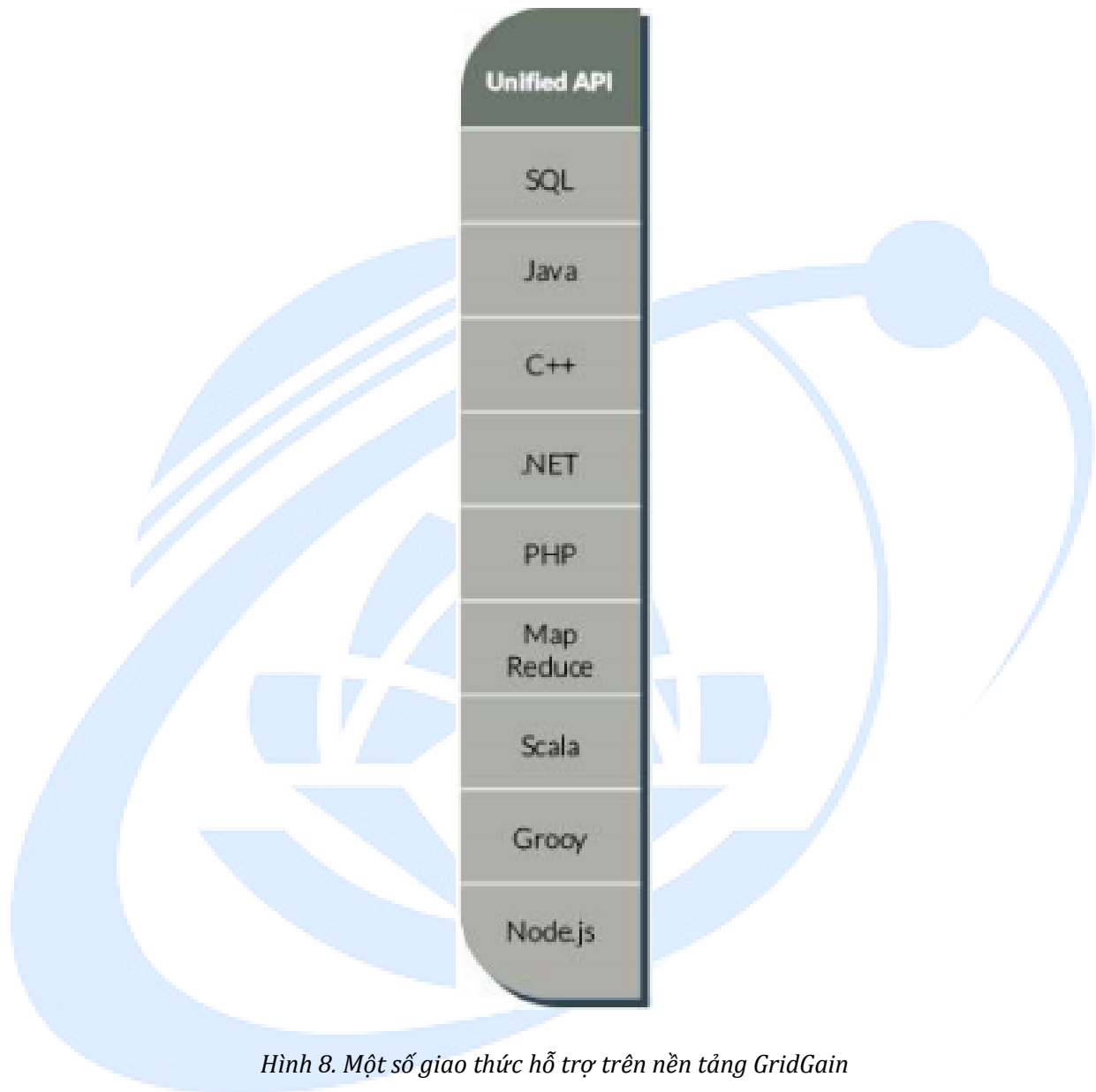
- Concurrent map
- Distributed queues and sets
- AtomicLong
- AtomicSequence
- AtomicReference
- CountdownLatch [1]

1.8. Đồng nhất API

Các giao thức được hỗ trợ đồng nhất API bao gồm:

- SQL
- JAVA
- C++
- .NET
- PHP
- MapReduce
- Scala
- Groovy
- Node.js

GridGain hỗ trợ một số giao thức để kết nối máy khách với các cụm Ignite, bao gồm Ignite Native Client, REST/HTTP, SSL/TLS và Memcached. [1]



Hình 8. Một số giao thức hỗ trợ trên nền tảng GridGain

2. Ưu và nhược điểm của GridGain

2.1. Ưu điểm

- **Hiệu suất cao:** GridGain mang lại hiệu suất có độ trễ mili giây cho cả khối lượng công việc giao dịch và phân tích, cho phép đưa ra quyết định theo thời gian thực và phản ứng nhanh với các điều kiện kinh doanh thay đổi.

- **Khả năng mở rộng:** GridGain có thể mở rộng theo chiều ngang để hỗ trợ hàng petabyte dữ liệu và hàng triệu giao dịch mỗi giây, khiến nó phù hợp với khối lượng công việc đòi hỏi khắt khe nhất của doanh nghiệp.
- **Tính linh hoạt:** GridGain hỗ trợ nhiều nguồn dữ liệu, mô hình xử lý và ngôn ngữ lập trình, giúp nhà phát triển tự do lựa chọn các công cụ và công nghệ đáp ứng tốt nhất nhu cầu của họ.
- **Độ tin cậy:** GridGain cung cấp các tính năng tích hợp sẵn về khả năng chịu lỗi, sao chép dữ liệu và tính nhất quán dữ liệu để đảm bảo tính sẵn sàng cao và tính toàn vẹn dữ liệu khi xảy ra lỗi nút và gián đoạn mạng.
- **Hiệu quả về mặt chi phí:** Mô hình hỗ trợ và giấy phép nguồn mở của GridGain khiến nó trở thành một giải pháp thay thế hiệu quả về mặt chi phí cho các nền tảng điện toán trong bộ nhớ độc quyền. [4]

2.2. Nhược điểm

- **Phức tạp trong triển khai và quản lý:** GridGain là một hệ thống phức tạp đòi hỏi trình độ chuyên môn nhất định để triển khai và quản lý một cách hiệu quả.
- **Tích hợp với các hệ thống và nguồn dữ liệu cũ:** Việc tích hợp GridGain với các hệ thống và nguồn dữ liệu cũ hiện có có thể là một thách thức vì việc này đòi hỏi phải lập kế hoạch và phối hợp cẩn thận để tránh mất mát hoặc hỏng dữ liệu. Nó cũng có thể yêu cầu các công cụ bổ sung hoặc phần mềm trung gian để hỗ trợ giao tiếp giữa các hệ thống.
- **Yêu cầu về kỹ năng và đào tạo:** GridGain yêu cầu trình độ chuyên môn kỹ thuật nhất định để hoạt động hiệu quả. Các chương trình đào tạo và chứng nhận có thể cần thiết để đảm bảo rằng nhân viên có những kỹ năng và kiến thức cần thiết để quản lý và vận hành hệ thống.
- **Các cân nhắc về bảo mật và tuân thủ:** GridGain được thiết kế để xử lý dữ liệu nhạy cảm và quan trọng, điều đó có nghĩa là phải tính đến các cân nhắc về bảo mật và tuân thủ. Điều này có thể bao gồm việc triển khai các giao thức bảo mật, công cụ giám sát và khung tuân thủ để đảm bảo dữ liệu được giữ an toàn và đáp ứng các yêu cầu quy định. [4]

3. So sánh

3.1. Về hiệu suất

Rất khó để đưa ra bảng so sánh chính xác về tốc độ của các sản phẩm khác nhau vì hiệu suất có thể thay đổi đáng kể tùy thuộc vào trường hợp sử dụng cụ thể, cấu hình phần cứng và việc triển khai phần mềm. Tuy nhiên, đây là so sánh sơ bộ của một số sản phẩm được liệt kê về hiệu suất được công bố: [5]

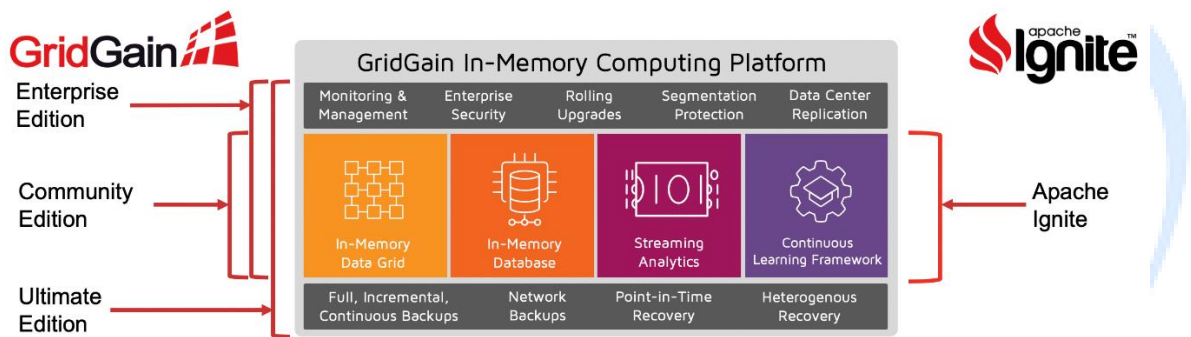
Bảng 1. Bảng so sánh hiệu suất của một số sản phẩm trên nền tảng dữ liệu lớn

Tên sản phẩm	Tốc độ xử lý truy vấn tối đa	Tốc độ ghi tối đa	Tốc độ đọc tối đa	Kích thước cụm tối đa
GridGain	40 triệu truy vấn/giây	3,5 triệu lần ghi/giây	6,8 triệu lần đọc/giây	Hơn 1000 node
Hazelcast	12 triệu truy vấn/giây	4,6 triệu lần ghi/giây	5,5 triệu lần đọc/giây	Hơn 1000 node
Apache Ignite	10 triệu truy vấn/giây	3,5 triệu lần ghi/giây	5,8 triệu lần đọc/giây	Hơn 1000 node
Apache Geode	10 triệu truy vấn/giây	3,3 triệu lần ghi/giây	5,8 triệu lần đọc/giây	Hơn 1000 node
Apache Cassandra	5 triệu truy vấn/giây	1,5 triệu lần ghi/giây	1,5 triệu lần đọc/giây	Hơn 1000 node
Couchbase	4 triệu truy vấn/giây	2,2 triệu lần ghi/giây	2,2 triệu lần đọc/giây	Hơn 100 node
Redis	1,2 triệu truy vấn/giây	0,7 triệu lần ghi/giây	1,2 triệu lần đọc/giây	Hơn 500 node

3.2. Về những sản phẩm tương tự

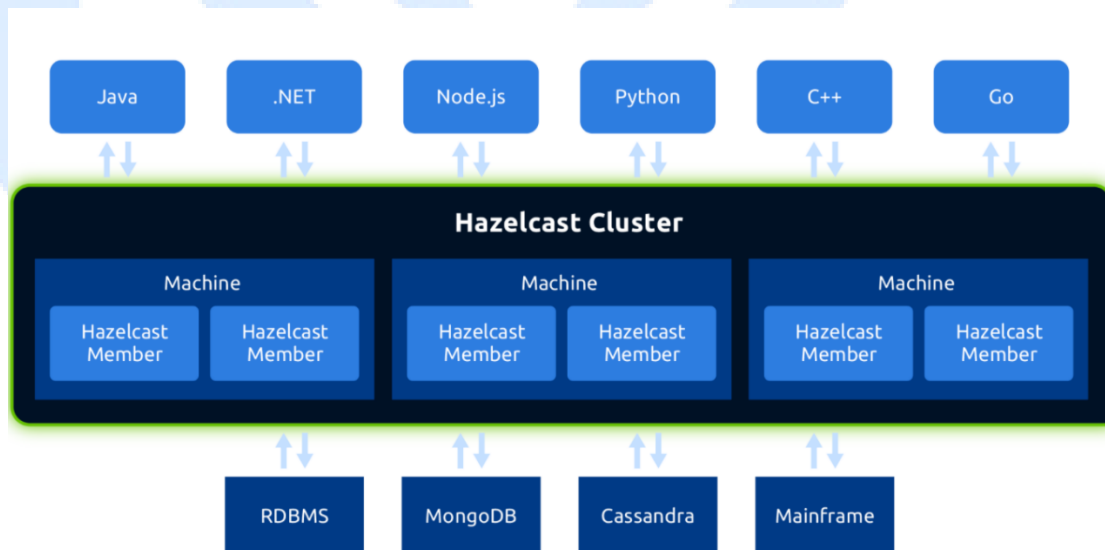
Một số sản phẩm hệ thống phân tán và điện toán trong bộ nhớ khác hiện có trên thị trường cung cấp các chức năng tương tự như GridGain. Một số trong những sản phẩm đáng chú ý là:

- **Apache Ignite:** Apache Ignite là một nền tảng xử lý, bộ đệm và cơ sở dữ liệu phân tán nguồn mở được thiết kế cho khối lượng công việc giao dịch, phân tích và truyền phát.



Hình 9. Kiến trúc của Apache Ignite [2]

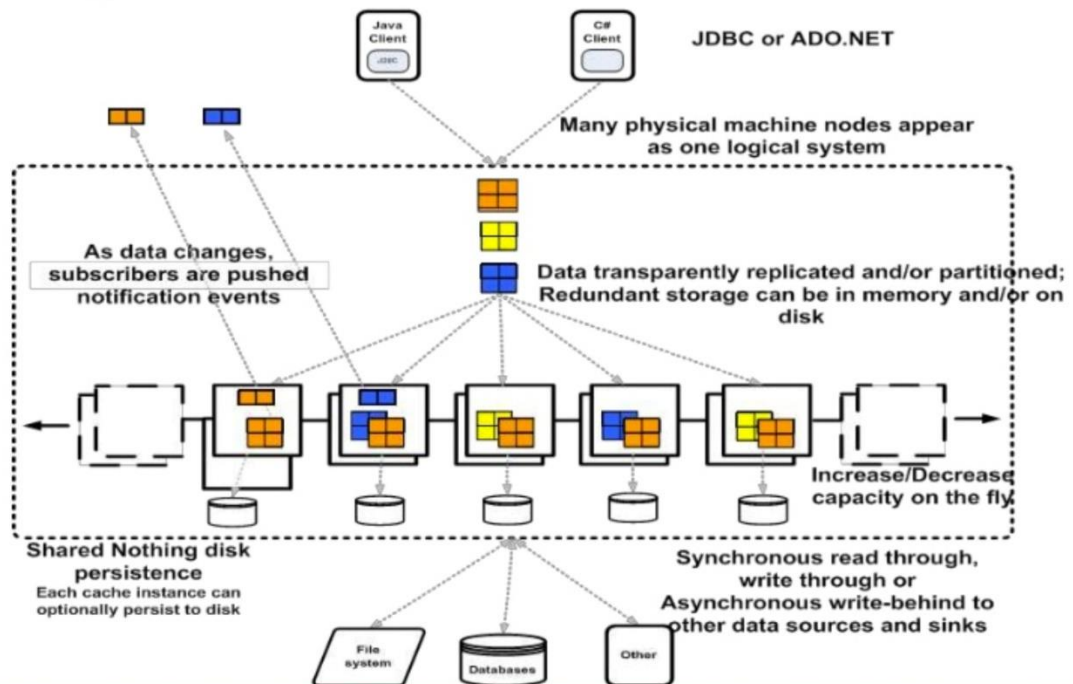
- **Hazelcast:** Hazelcast là một nền tảng xử lý luồng và lưới dữ liệu trong bộ nhớ nguồn mở, cung cấp môi trường điện toán phân tán cho các ứng dụng Java.



Hình 10. Kiến trúc Hazelcast Cluster [6]

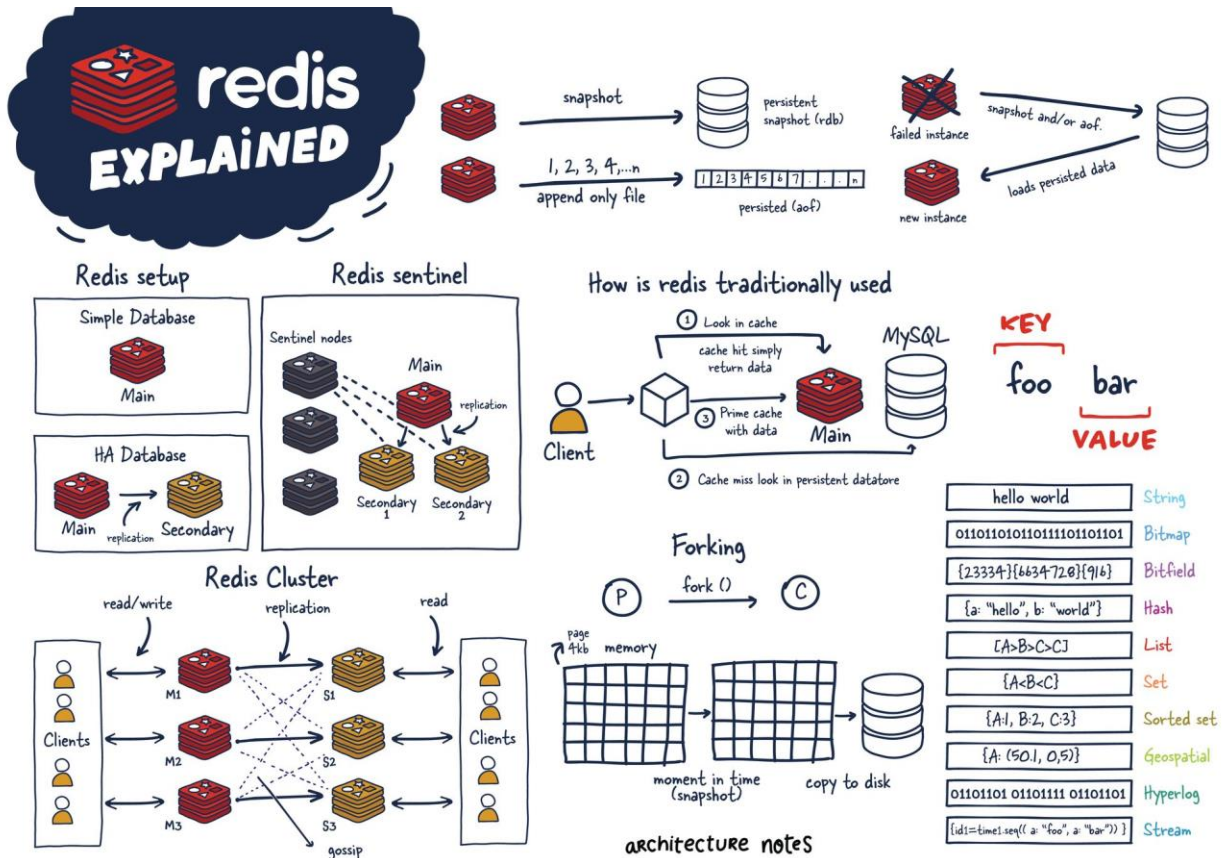
- **Apache Geode:** Apache Geode là một hệ thống quản lý dữ liệu trong bộ nhớ phân tán nguồn mở, cung cấp khả năng phân tích và xử lý dữ liệu theo thời gian thực.

Geode High Level Architecture



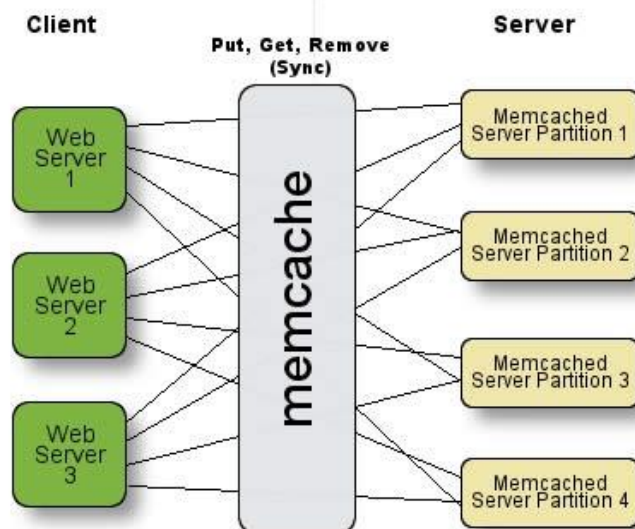
Hình 11. Kiến trúc của Apache Geode [7]

- **Oracle Coherence:** Oracle Coherence là một giải pháp lưới dữ liệu trong bộ nhớ cung cấp bộ đệm phân tán cho các ứng dụng Java.
- **Redis:** Redis là kho lưu trữ cấu trúc dữ liệu trong bộ nhớ nguồn mở có thể được sử dụng làm cơ sở dữ liệu, bộ nhớ đệm và môi giới truyền tin.



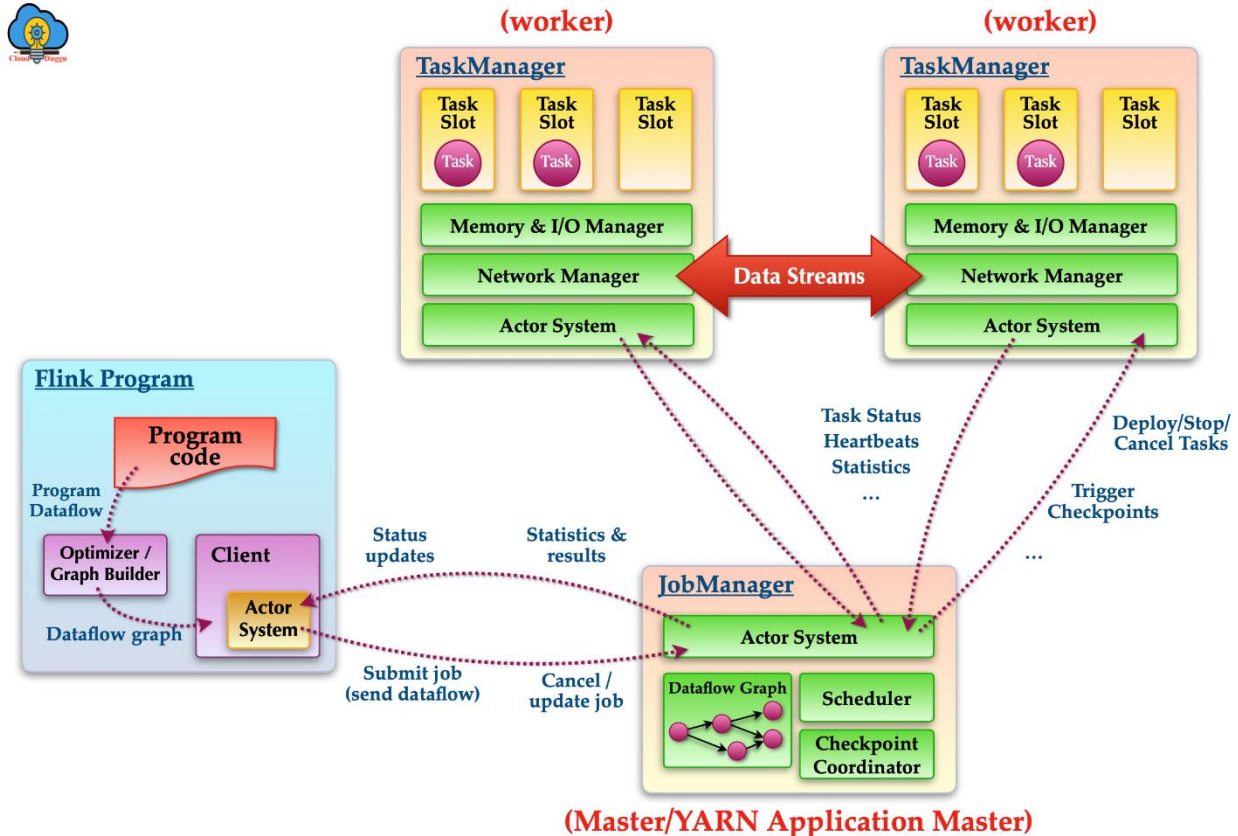
Hình 12. Sơ lược về Redis

- **Memcached:** Memcached là một hệ thống bộ nhớ đệm phân tán thường được sử dụng để tăng tốc các ứng dụng web động.



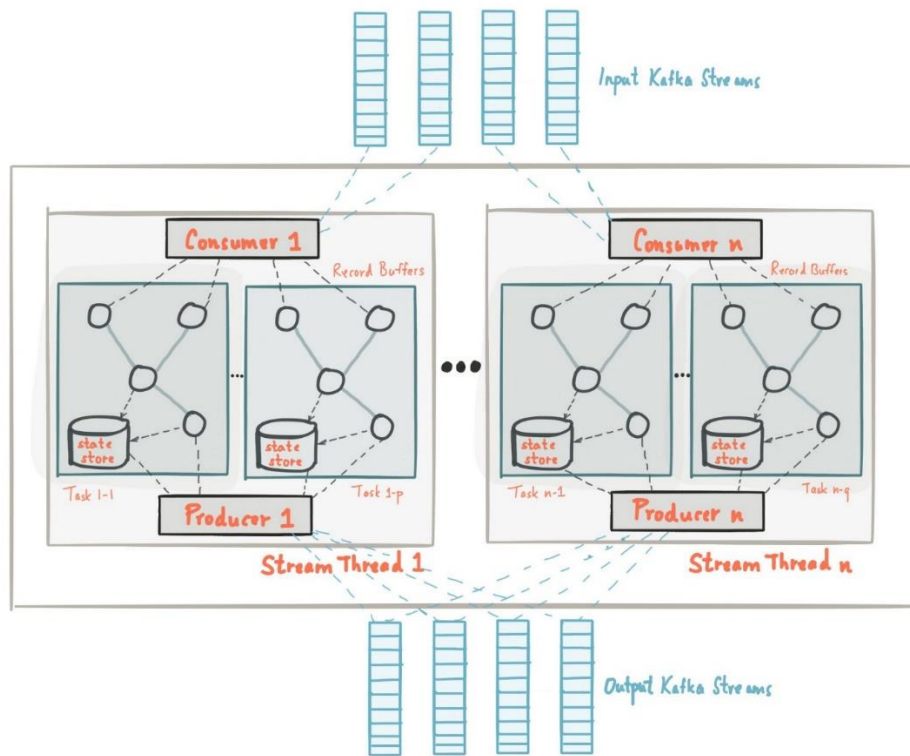
Hình 13. Kiến trúc Memcached

- **Apache Flink:** Apache Flink là một framework xử lý luồng mã nguồn mở cung cấp khả năng xử lý dữ liệu có thông lượng cao, độ trễ thấp cho các ứng dụng thời gian thực.



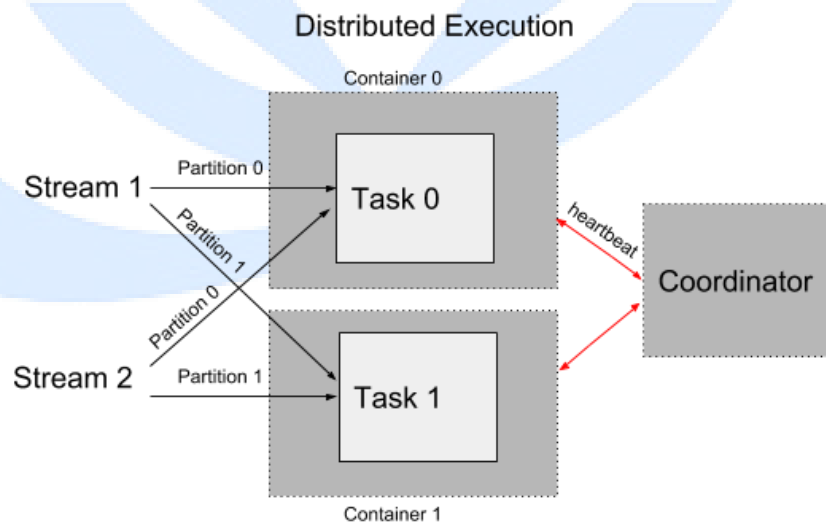
Hình 14. Kiến trúc của một Flink

- **Apache Kafka Streams:** Apache Kafka Streams là một thư viện xử lý luồng nguồn mở cung cấp một công cụ xử lý phân tán để xử lý và phân tích dữ liệu theo thời gian thực.



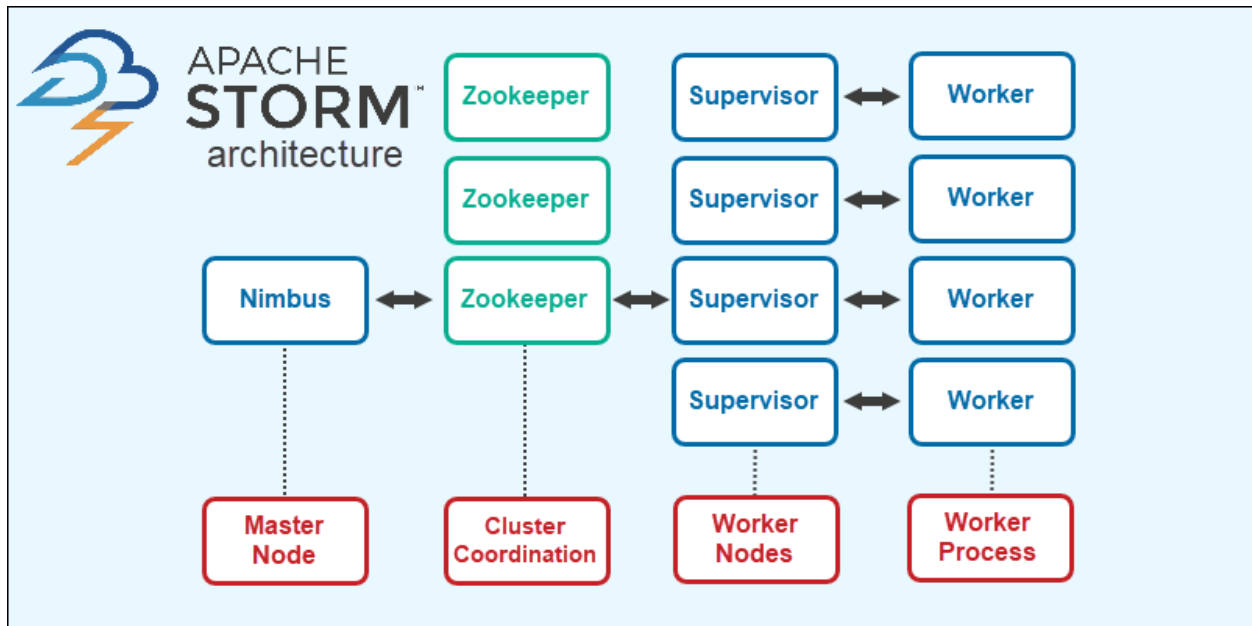
Hình 15. Kiến trúc của Apache Kafka Streams

- **Apache Samza:** Apache Samza là một khung xử lý luồng nguồn mở cung cấp môi trường điện toán phân tán để xử lý và phân tích dữ liệu theo thời gian thực.



Hình 16. Kiến trúc của Apache Samza

- **Apache Storm:** Apache Storm là một hệ thống tính toán thời gian thực phân tán nguồn mở cung cấp công cụ xử lý phân tán để xử lý và phân tích luồng.



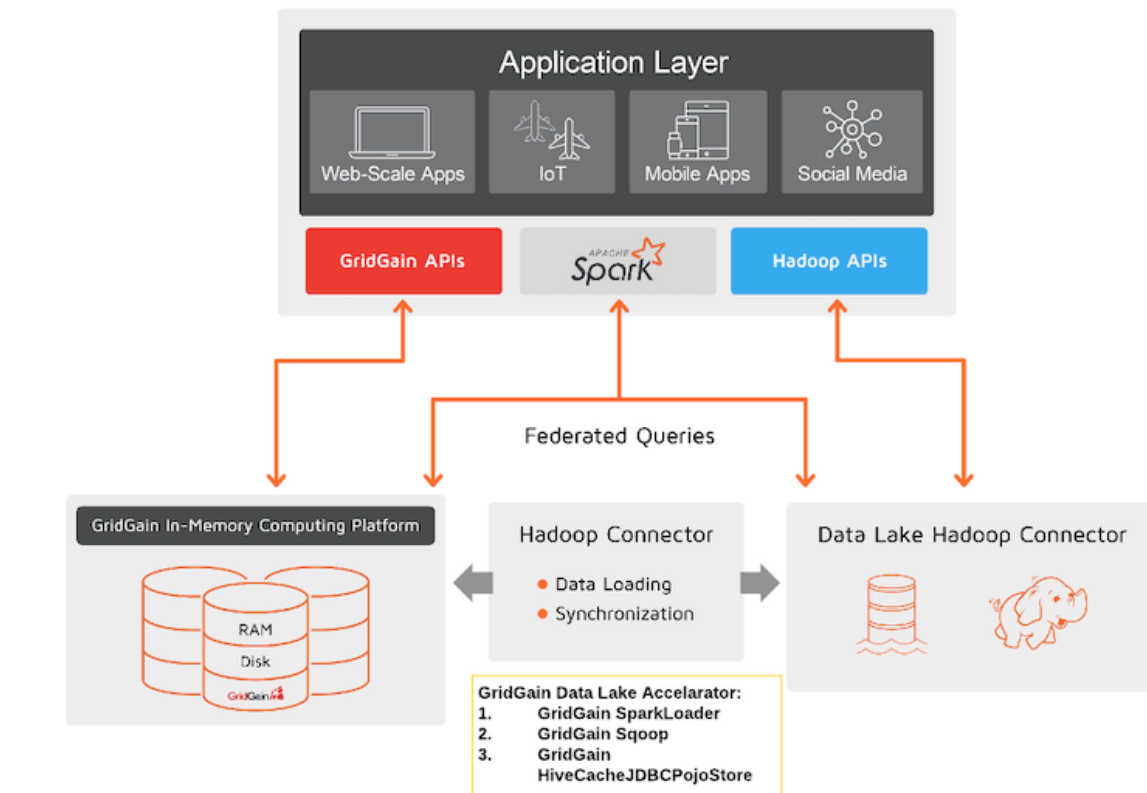
Hình 17. Kiến trúc của Apache Storm

Mỗi sản phẩm này đều có những tính năng, điểm mạnh và điểm yếu riêng và việc lựa chọn sản phẩm phù hợp tùy thuộc vào trường hợp sử dụng cụ thể và yêu cầu của ứng dụng.

CHƯƠNG 3: ỨNG DỤNG CỦA GRIDGAIN

Khả năng tích hợp tính toán trong bộ nhớ và hệ thống phân tán của GridGain làm cho nó trở thành một giải pháp đa dạng cho nhiều ngành công nghiệp và trường hợp sử dụng. Dưới đây là một số ví dụ về cách GridGain được sử dụng:

1. Tăng tốc Data Lake Hadoop.



Hình 18. Ứng dụng của GridGain trên Data Lake Hadoop

Tăng tốc Data Lake Hadoop là quá trình cải thiện hiệu suất và hiệu quả của việc xử lý dữ liệu trong một Data Lake Hadoop. Hadoop là một nền tảng tính toán phân tán phổ biến dùng để xử lý dữ liệu lớn, sử dụng Hadoop Distributed File System (HDFS) để lưu trữ các tập dữ liệu lớn trên nhiều node trong một cụm.

Tuy nhiên, việc xử lý lượng lớn dữ liệu trong Hadoop có thể mất thời gian do cần đọc và ghi dữ liệu từ đĩa, cùng với sự phức tạp của việc quản lý các nhiệm vụ xử lý dữ liệu trên một cluster lớn.

GridGain là một nền tảng tính toán trong bộ nhớ có thể được sử dụng để tăng tốc xử lý dữ liệu trong Data Lake Hadoop bằng cách cung cấp một lớp in-memory data grid (IMDG) trên Hadoop. Lớp IMDG lưu trữ dữ liệu thường xuyên được truy cập trong bộ nhớ, giảm nhu cầu đọc và ghi dữ liệu từ đĩa. Điều này có thể cải thiện đáng kể hiệu suất và hiệu quả của việc xử lý dữ liệu, đặc biệt là trong các trường hợp xử lý dữ liệu thời gian thực hoặc gần thời gian thực.

GridGain cũng cung cấp các tính năng bổ sung như bộ nhớ đệm phân tán, phân tích thời gian thực, học máy và hỗ trợ cơ sở dữ liệu SQL và NoSQL, có thể được sử dụng để nâng cao khả năng của một Data Lake Hadoop. Bằng cách tích hợp GridGain vào Hadoop, tổ chức có thể tận dụng sức mạnh của cả hai nền tảng để tăng tốc xử lý và phân tích dữ liệu, đồng thời giảm độ phức tạp và cải thiện khả năng mở rộng.

2. Các dịch vụ tài chính.

GridGain được sử dụng trong các dịch vụ tài chính để quản lý rủi ro theo thời gian thực, phát hiện gian lận và tuân thủ. Nó giúp xác định các vấn đề tiềm ẩn và sự bất thường trong thời gian thực, cho phép các tổ chức tài chính đưa ra quyết định nhanh chóng và sáng suốt.

3. Thương mại điện tử và bán lẻ.

Trong ngành thương mại điện tử và bán lẻ, GridGain được sử dụng để quản lý hàng tồn kho theo thời gian thực, đề xuất sản phẩm và phát hiện gian lận. Nó giúp cung cấp trải nghiệm mua sắm được cá nhân hóa cho khách hàng và ngăn chặn các giao dịch gian lận.

4. Viễn thông.

GridGain được sử dụng trong viễn thông để giám sát và tối ưu hóa mạng theo thời gian thực. Nó giúp xác định các sự cố mạng trong thời gian thực và nhanh chóng

giải quyết chúng để giảm thiểu thời gian ngừng hoạt động và đảm bảo hiệu suất mạng tối ưu.

5. Chăm sóc sức khỏe.

Trong ngành chăm sóc sức khỏe, GridGain được sử dụng để phân tích dữ liệu theo thời gian thực, hỗ trợ ra quyết định lâm sàng và theo dõi bệnh nhân. Nó giúp cải thiện kết quả của bệnh nhân và cung cấp dịch vụ chăm sóc tốt hơn.

6. Năng lượng và tiện ích.

GridGain được sử dụng trong lĩnh vực năng lượng và tiện ích để quản lý lưới điện theo thời gian thực, bảo trì dự đoán và tối ưu hóa tài sản. Nó giúp tối ưu hóa việc phân phối năng lượng và giảm thiểu thời gian ngừng hoạt động.

7. Internet of Things (IoT).

GridGain được sử dụng trong IoT để xử lý, phân tích và ra quyết định dữ liệu theo thời gian thực. Nó giúp quản lý và xử lý lượng lớn dữ liệu do thiết bị IoT tạo ra và cho phép đưa ra quyết định theo thời gian thực. [5]



Hình 19. Một số ứng dụng của GridGain trong thực tiễn [8]

CHƯƠNG 4: CÀI ĐẶT GRIDGAIN

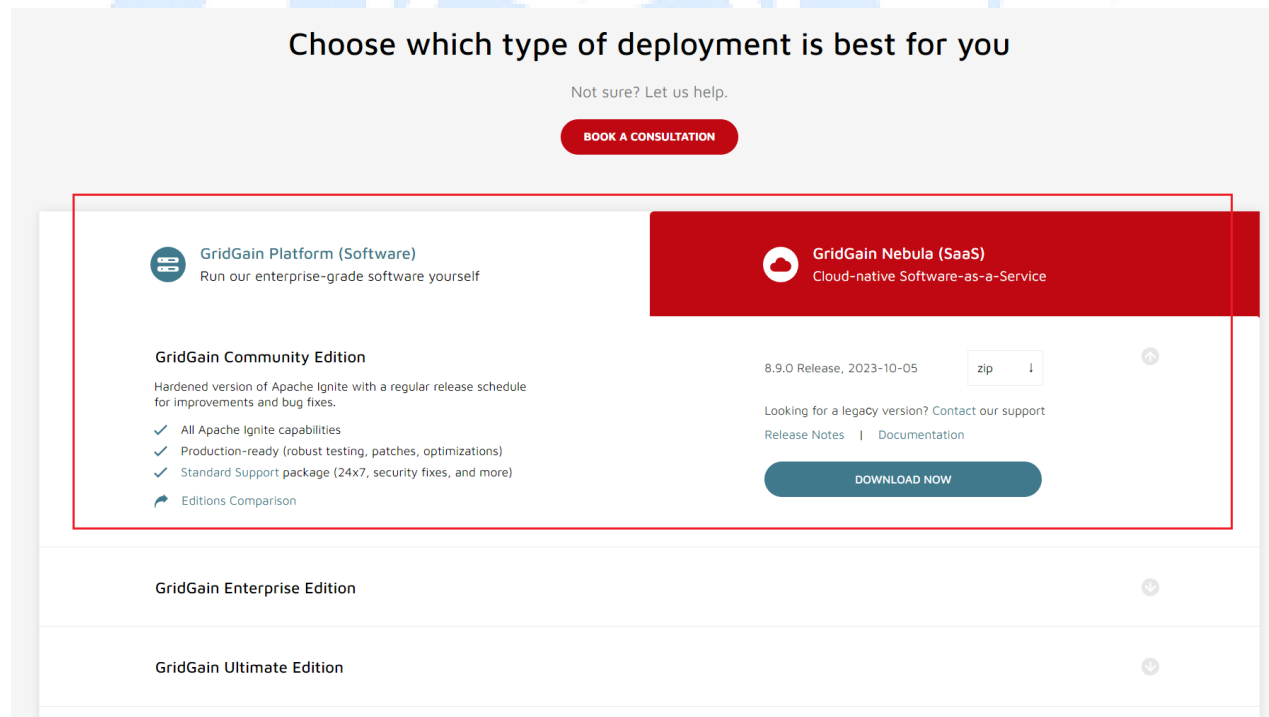
Để cài đặt GridGain, ta làm theo các bước sau:

- **Bước 1:** Kiểm tra yêu cầu hệ thống.

Trước khi cài đặt GridGain, ta cần phải kiểm tra xem hệ thống đã đáp ứng đủ yêu cầu để chạy GridGain hay chưa, bao gồm các yêu cầu sau:

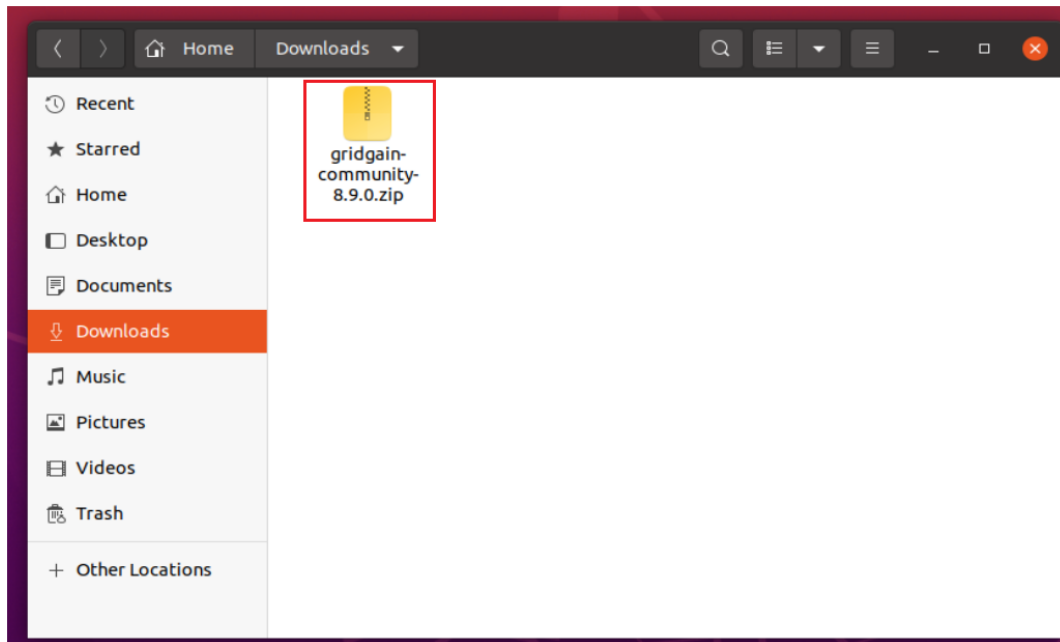
- Linux (phiên bản bất kỳ), Mac OSX (10.6 trở lên), Windows (XP trở lên), Windows Server (2008 trở lên), Oracle Solaris, z/OS.
- JDK: Oracle JDK 8, 11 hoặc 17, Open JDK 8, 11 hoặc 17, IBM JDK 8, 11 hoặc 17.
- ISA: x86, x64, SPARC, PowerPC.
- **Bước 2:** Tải xuống GridGain.

GridGain có 3 phiên bản: Community Edition (CE), Enterprise Edition (EE) và Ultimate Edition (UE). Ở đây, chọn phiên bản **Community Edition (CE)** để tải xuống.



Hình 20. Tải xuống phần mềm GridGain phiên bản Community Edition

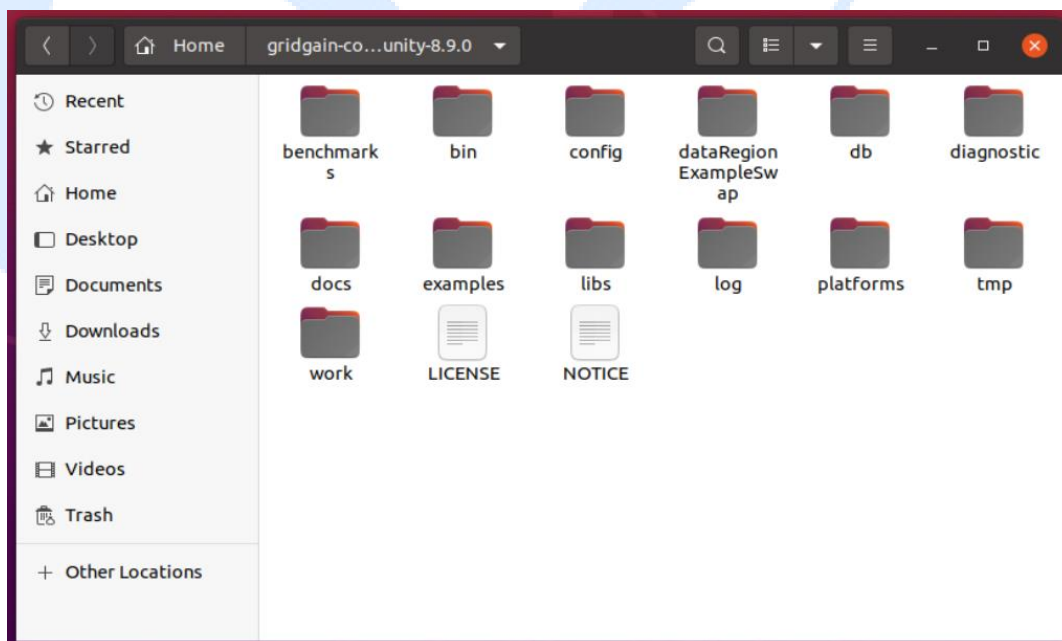
Sau khi tải xong ta được file “gridgain-community-8.9.0.zip” (T11/2023)



Hình 21. Tập tin nén *gridgain-community-8.9.0.zip*

- **Bước 3:** Giải nén tập tin.

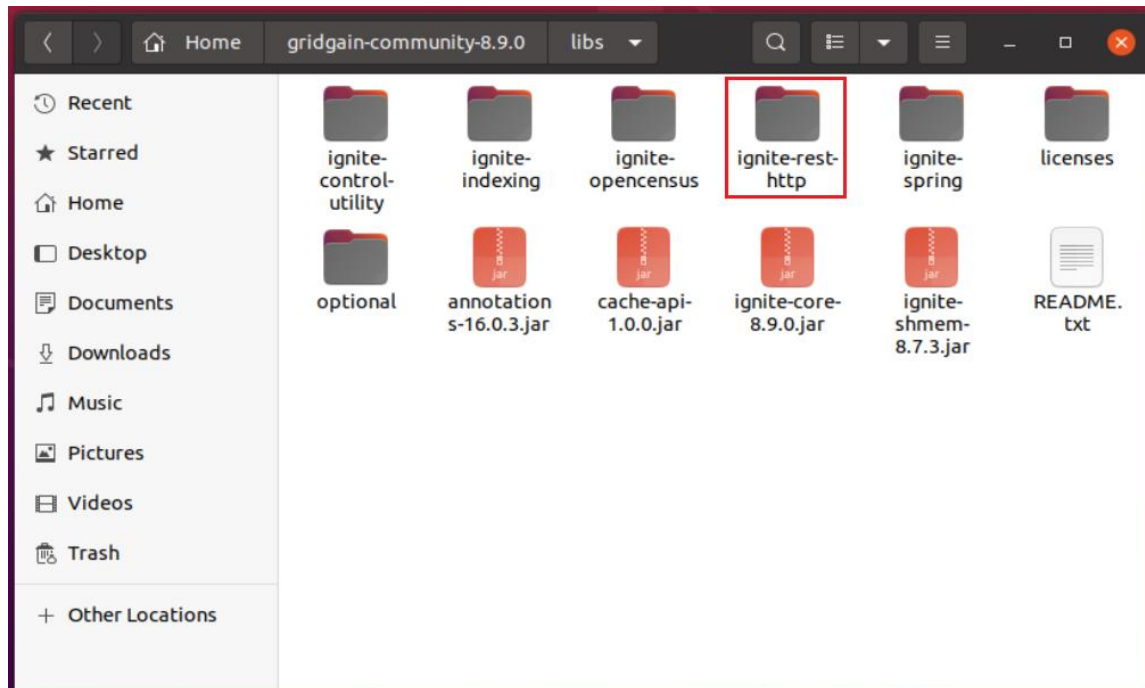
Ta giải nén tập tin zip đã tải vào thư mục bạn muốn sử dụng. Ví dụ: ~/gridgain-community-8.9.0



Hình 22. Các thư mục hiển thị sau khi giải nén tập tin *gridgain-community-8.9.0.zip*

- **Bước 4:** Di chuyển thư mục để kích hoạt thư viện.

Di chuyển thư mục `ignite-rest-http` từ `{gridgain}/libs/optional` đến `{gridgain}/libs` để kích hoạt thư viện Ignite REST cho cụm. Thư viện này được sử dụng bởi GridGain Nebula cho nhu cầu giám sát và quản lý cụm.

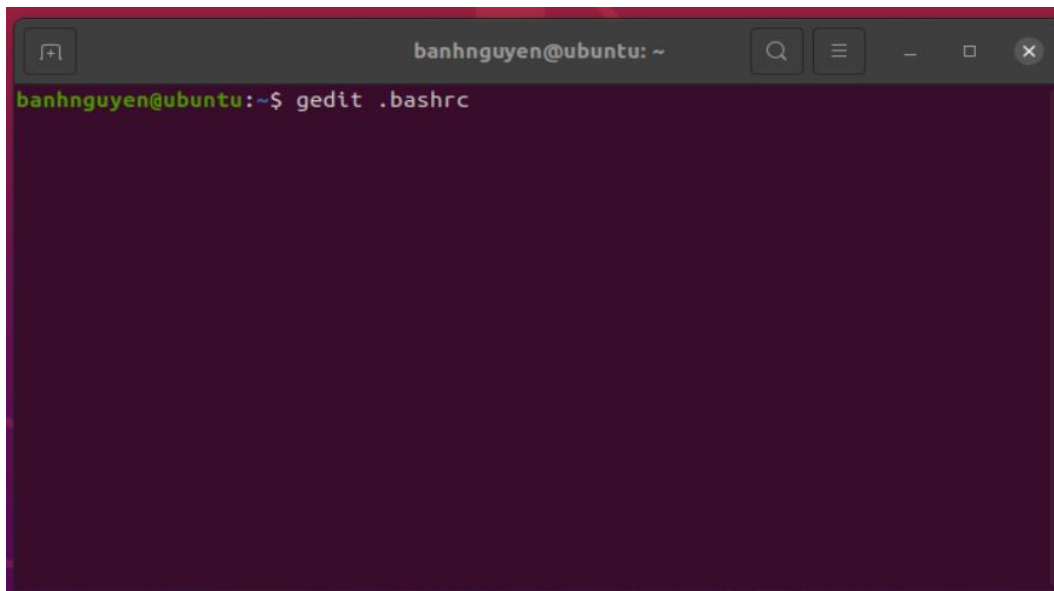
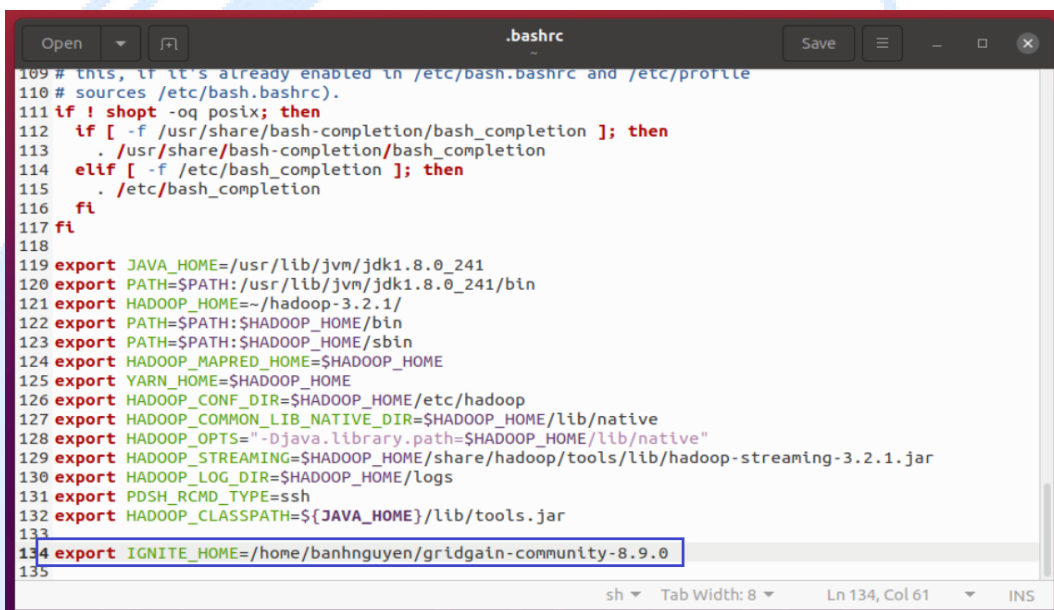


Hình 23. Di chuyển thư mục `ignite-rest-http` đến đường dẫn `{gridgain}/libs`

- **Bước 5:** Thiết lập biến môi trường.

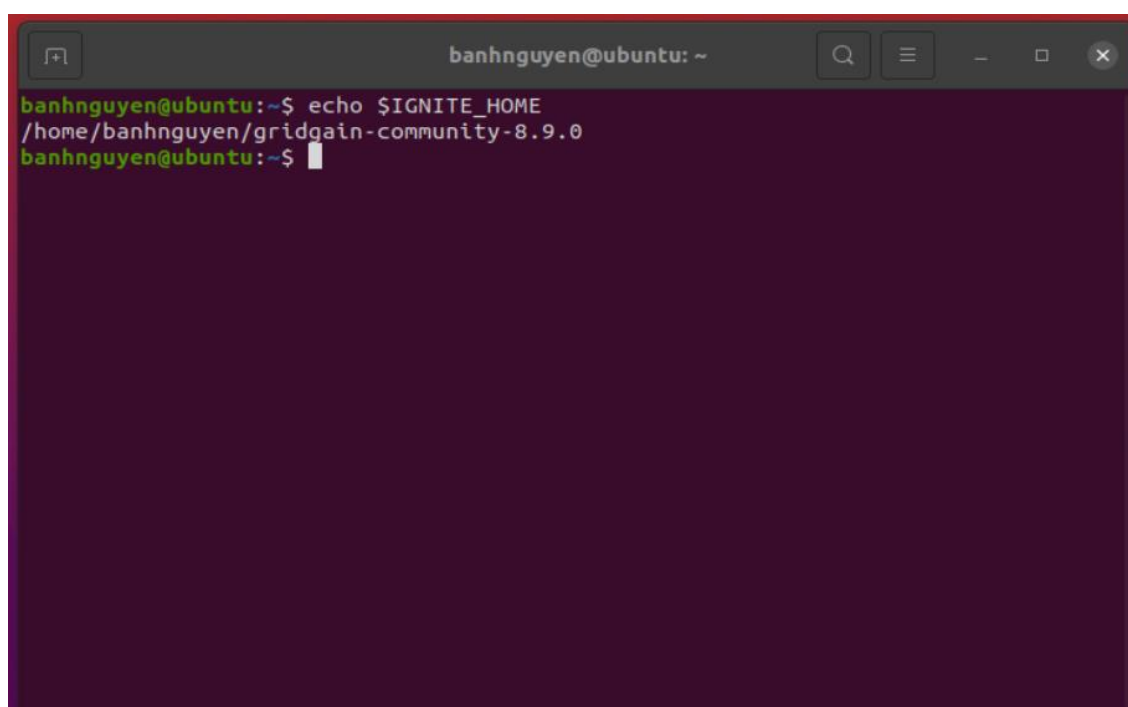
Thêm biến môi trường `IGNITE_HOME` để chỉ đến thư mục cài đặt GridGain. Sau đó, mở file `.bashrc` và thêm dòng lệnh sau:

```
export IGNITE_HOME=/home/{username}/gridgain-community-8.9.0
```

Hình 24. Mở tệp `.bashrc`

Hình 25. Thêm dòng lệnh tạo biến môi trường

Nhập lệnh `source .bashrc` để tải lại file, tiếp tục thoát terminal và bật lại. Nhập `echo $IGNITE_HOME`, nếu có trả kết quả về thì việc thêm biến môi trường thành công.

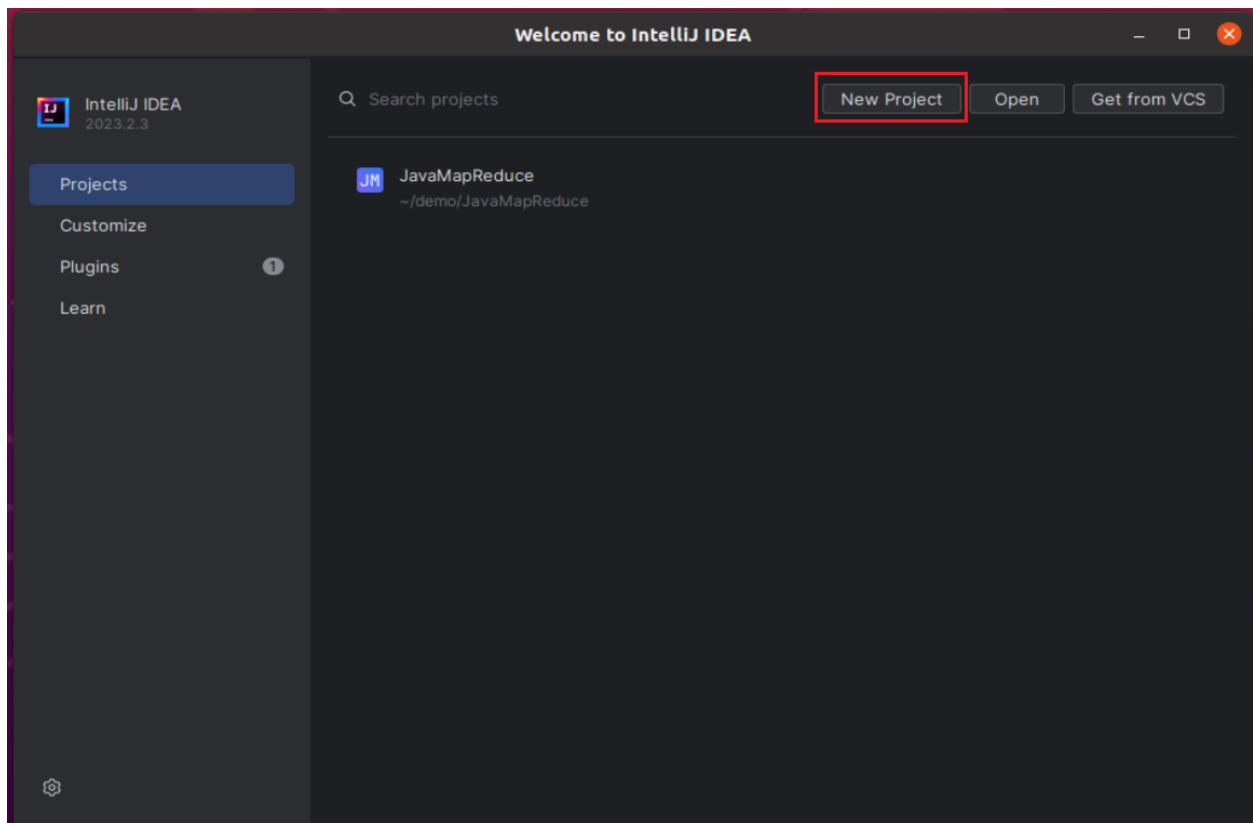


```
banhnguyen@ubuntu: ~  
banhnguyen@ubuntu:~$ echo $SIGNITE_HOME  
/home/banhnguyen/gridgain-community-8.9.0  
banhnguyen@ubuntu:~$
```

Hình 26. Tạo biến môi trường thành công

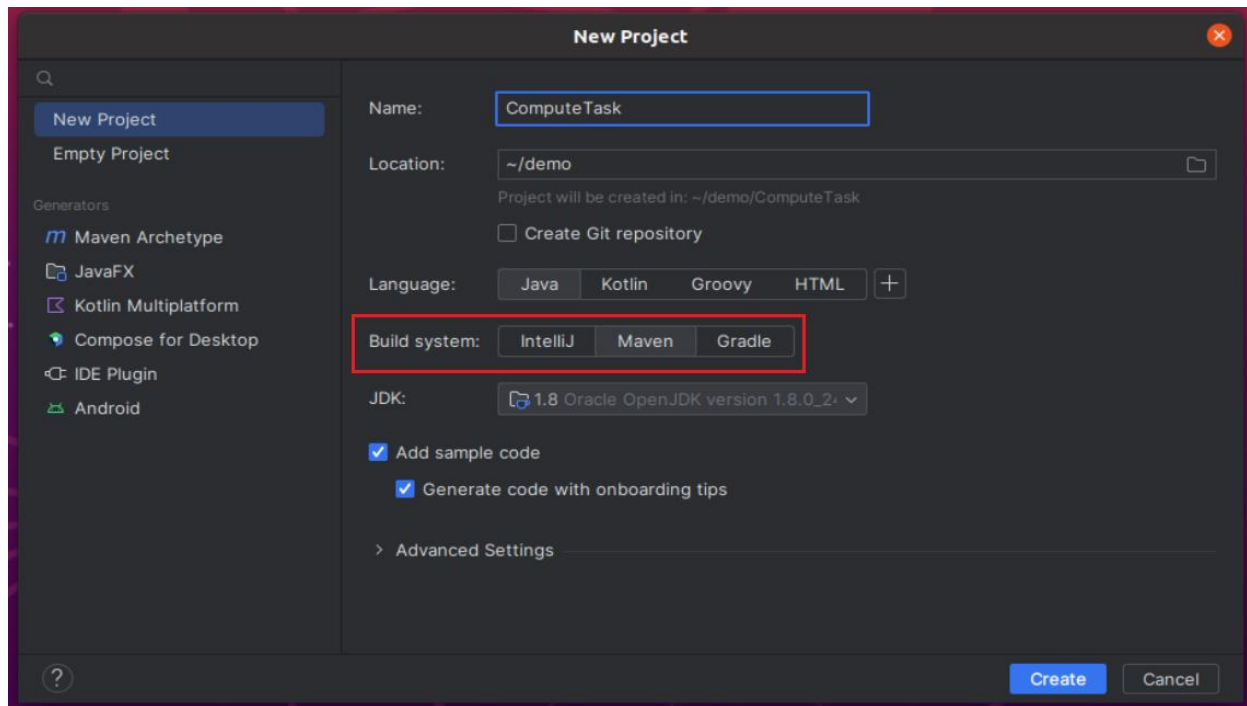
CHƯƠNG 5: MINH HỌA XỬ LÝ DỮ LIỆU LỚN TRÊN GRIDGAIN

- **Bước 1:** Tạo một project mới trên IDE IntelliJ IDEA (có thể sử dụng các IDE tương tự như VSCode, Eclipse, NetBeans, v.v)



Hình 27. Tạo project mới trên IntelliJ IDEA

- **Bước 2:** Đặt tên project với tên `ComputeTask`, ở phần Build System chọn mục Maven



Hình 28. Nhập tên project mới và chọn Build System tương ứng là Maven

- **Bước 3:** Xóa file Main trong thư mục `src/main/java/org/example` và thêm mới file `ComputeTaskExample` với đoạn lệnh như sau:

```
package org.example;

import org.apache.ignite.Ignite;
import org.apache.ignite.IgniteCompute;
import org.apache.ignite.Ignition;
import org.apache.ignite.compute.ComputeJob;
import org.apache.ignite.compute.ComputeJobAdapter;
import org.apache.ignite.compute.ComputeJobResult;
import org.apache.ignite.compute.ComputeTaskSplitAdapter;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
```

```

import java.util.ArrayList;

import java.util.List;

public class ComputeTaskExample {

    private static class CharacterCountTask extends
ComputeTaskSplitAdapter<String, Integer> {

        // 1. Splits the received string into words
        // 2. Creates a child job for each word
        // 3. Sends the jobs to other nodes for processing.

        @Override

        public List<ComputeJob> split(int gridSize, String arg) {

            String[] words = arg.split(" ");

            List<ComputeJob> jobs = new ArrayList<>(words.length);

            for (final String word : words) {

                jobs.add(new ComputeJobAdapter() {

                    @Override

                    public Object execute() {

                        System.out.println(">>> Printing '" + word + "'
on from compute job.");

                        // Return the number of letters in the word.

                        return word.length();

                    }

                });

            }

        }
    }
}

```

```

        return jobs;
    }

    @Override
    public Integer reduce(List<ComputeJobResult> results) {
        int sum = 0;

        for (ComputeJobResult res : results)
            sum += res.<Integer>getData();

        return sum;
    }
}

public static void main(String[] args) {
    Ignite ignite = Ignition.start();
    IgniteCompute compute = ignite.compute();
    String text;

    try(BufferedReader br = new BufferedReader(new
    FileReader("/home/banhnguyen/demo/input.txt"))) {

        StringBuilder sb = new StringBuilder();

        String line = br.readLine();

        while (line != null) {

            sb.append(line);

```

```

        sb.append(System.lineSeparator());

        line = br.readLine();

    }

    text = sb.toString();

} catch (IOException e) {

    throw new RuntimeException(e);

}

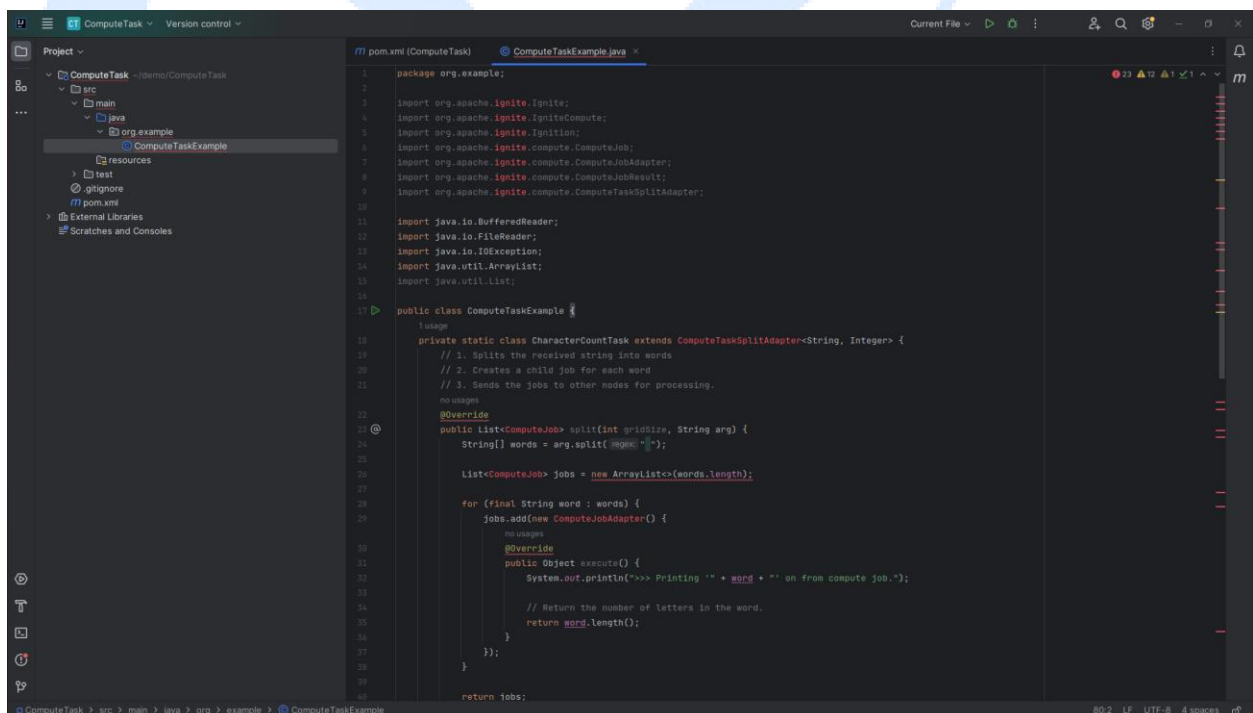
// Execute the task on the cluster and wait for its completion.
int cnt = compute.execute(CharacterCountTask.class, text);

System.out.println(">>> Total number of characters in the phrase
is '" + cnt + "'.");

}

}

```



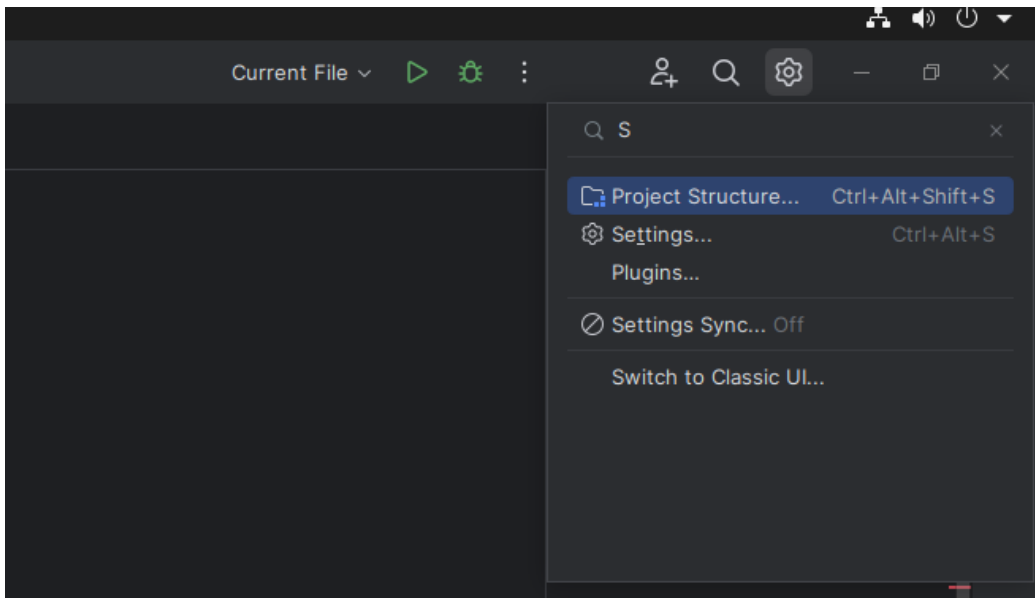
```

1 package org.example;
2
3 import org.apache.ignite.Ignite;
4 import org.apache.ignite.IgniteCompute;
5 import org.apache.ignite.Ignition;
6 import org.apache.ignite.compute.ComputeJob;
7 import org.apache.ignite.compute.ComputeJobAdapter;
8 import org.apache.ignite.compute.ComputeJobResult;
9 import org.apache.ignite.compute.ComputeTaskSplitAdapter;
10
11 import java.io.BufferedReader;
12 import java.io.FileReader;
13 import java.io.IOException;
14 import java.util.ArrayList;
15 import java.util.List;
16
17 public class ComputeTaskExample {
18     // 1. Splits the received string into words
19     // 2. Creates a child job for each word
20     // 3. Sends the jobs to other nodes for processing.
21     noUsage
22     @Override
23     public List<ComputeJob> split(int gridSize, String arg) {
24         String[] words = arg.split("\\s");
25
26         List<ComputeJob> jobs = new ArrayList<>(words.length);
27
28         for (final String word : words) {
29             jobs.add(new ComputeJobAdapter() {
30                 noUsage
31                 @Override
32                 public Object execute() {
33                     System.out.println(">>> Printing '" + word + "' on from compute job.");
34
35                     // Return the number of letters in the word.
36                     return word.length();
37                 }
38             });
39         }
40
41         return jobs;
42     }
43 }

```

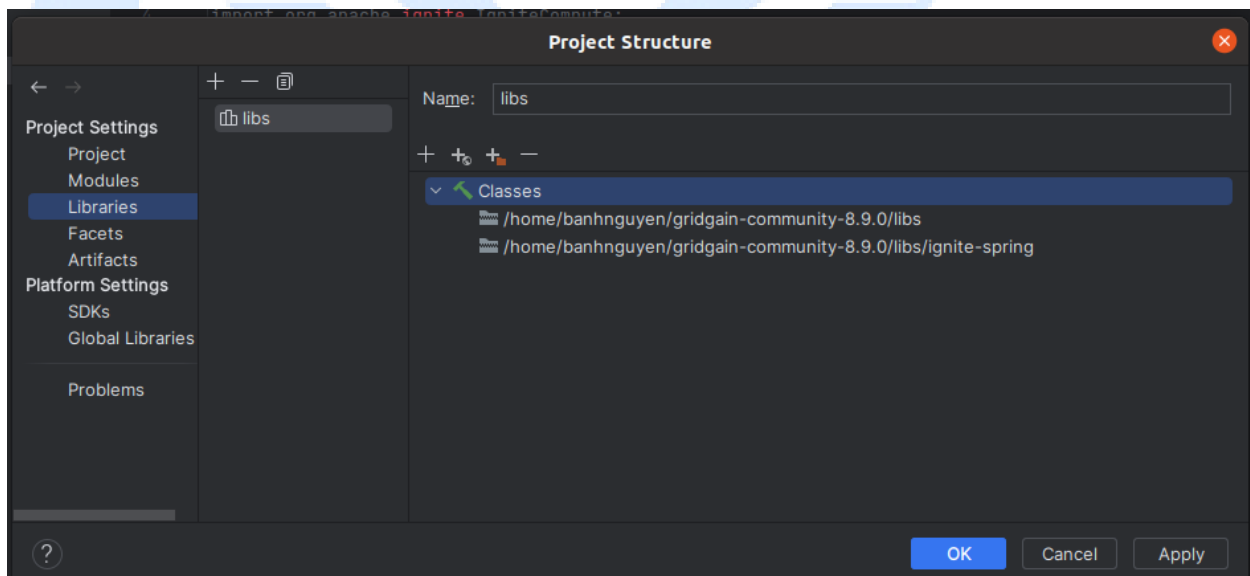
Hình 29. Viết đoạn mã trên để xử lý yêu cầu

- **Bước 4:** Thêm đường dẫn đến các thư viện của GridGain như sau:



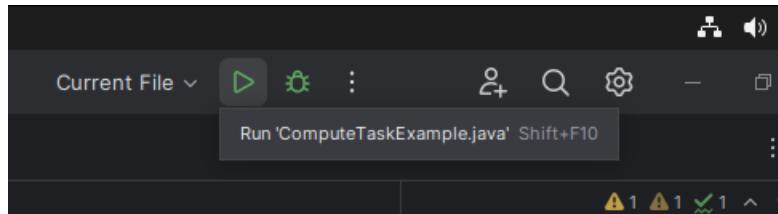
Hình 30. Chọn Project Structure để thêm đường dẫn đến các thư viện

Trong mục Libraries, thêm các thư viện cần thiết để chạy với đường dẫn `{gridgain}/libs` và `{gridgain}/libs/ignite-spring`. Tiếp theo chọn Apply và OK.



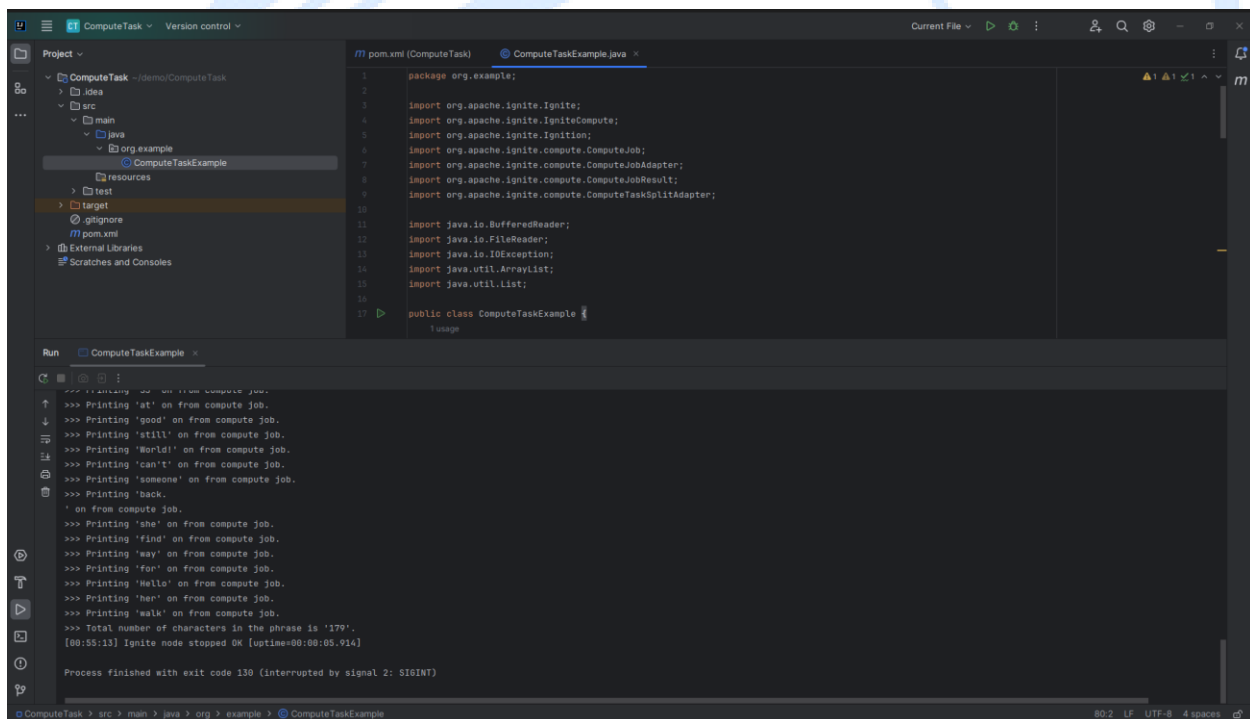
Hình 31. Thêm các thư viện cần thiết để thực thi

- **Bước 5:** Chạy file `ComputeTaskExample.java` trong thư mục project `src/main/java/org/example`



Hình 32. Chạy file `ComputeTaskExample.java`

- **Bước 6:** Đảm bảo rằng file đã được khởi động và thực thi thành công, như trong hình bên dưới.



Hình 33. Kết quả sau khi thực thi chương trình

TÀI LIỆU THAM KHẢO

- [1] GridGain Systems, Inc. All, "Introduction to GridGain," 2017.
- [2] GridGain, "Overview | GridGain Documentation," [Online]. Available: <https://www.gridgain.com/docs/index.html>. [Accessed 6 November 2023].
- [3] Google AI, "Google Bard: A Large Language Model," Google, 2023.
- [4] OpenAI, "OpenAI GPT-3," OpenAI, San Francisco, 2020.
- [5] Mohamed Ashraf K, ChatGPT Co-Authored., "Gridgain - In Memory Computing - White paper," 17 April 2023. [Online]. Available: <https://techyjargon.blogspot.com/2023/04/gridgain-in-memory-computing-white-paper.html>. [Accessed 7 November 2023].
- [6] Hazelcast, "Low Latency Storage," Hazelcast, [Online]. Available: <https://hazelcast.com/products/>. [Accessed 7 November 2023].
- [7] N. Sabharwal, "Apache Geode," LinkedIn, 9 July 2017. [Online]. Available: <https://www.linkedin.com/pulse/apache-geode-neeraj-sabharwal/>. [Accessed 7 November 2023].
- [8] GridGain Systems, "In-Memory Computing Essentials for Software Engineers," GridGain Systems, California, 2020.

