الأكاديمية العربية الدولية
Arab International Academy

النظم الموزعة

distributed systems

م. خليل المحمد

كلية العلوم – ماجستير علم البيانات

الأكاديمية العربية الدولية
Arab International Academy

المحاور

الأكاديمية العربية الدولية
Arab International Academy

المخرجات المتوقعة من المحاضرة

1. فهم مفهوم النظم الموزعة: الإلمام بتعريفها وأهميتها مقارنة بالأنظمة التقليدية.
2. معرفة المكونات الأساسية: التعرف على مكونات النظم الموزعة مثل النواة، الخوادم، والعمليات.
3. إدراك التحديات والحلول: استيعاب التحديات التقنية مثل التزامن وإدارة الأخطاء، والطرق المستخدمة للتعامل معها.
4. فهم نماذج المعمارية الموزعة: القدرة على التمييز بين الأنماط المختلفة مثل العميل/الخادم، النظراء/النظراء، والنظم المتمركزة حول البيانات.
5. التعرف على التطبيقات العملية: إدراك استخدامات النظم الموزعة في المجالات المختلفة مثل الحوسبة السحابية، إنترنت الأشياء، والخدمات السحابية.
6. تطوير المهارات التقنية: اكتساب المعرفة بالأدوات واللغات المستخدمة في تصميم وتطوير النظم الموزعة مثل Hadoop و Kubernetes.

1. مقدمة إلى النظم الموزعة

تعريف النظم الموزعة

النظم الموزعة هي مجموعة من الحواسيب المستقلة التي تبدو للمستخدم وكأنها نظام واحد متكامل. يتم تصميم هذه النظم بحيث تتشارك الموارد والوظائف، مع العمل على تحسين الأداء والاعتمادية. تشمل هذه النظم شبكات الحواسيب، قواعد البيانات الموزعة، والحوسبة السحابية.

أهمية النظم الموزعة وتطبيقاتها

تعتبر النظم الموزعة حجر الأساس للكثير من التطبيقات الحديثة التي تعتمد على الاتصال والمشاركة، مثل:

- **الخدمات السحابية:** مثل Google Drive و Dropbox.
- **التطبيقات الاجتماعية:** مثل Facebook و WhatsApp.
- **أنظمة التجارة الإلكترونية:** مثل Amazon و eBay.
- **البنية التحتية للإنترنت:** مثل DNS وخوادم المحتوى.
- **تطبيقات إنترنت الأشياء (IoT):** لتنسيق عمل الأجهزة المتصلة.

1. مقدمة إلى النظم الموزعة

التحديات في تصميم النظم الموزعة

- **التزامن :**ضمان عمل الأجزاء المختلفة من النظام بتناغم.
- **الأمان :**حماية البيانات أثناء النقل وفي التخزين.
- **الاتساق :**التأكد من أن جميع العقد تتشارك نفس البيانات في الوقت المناسب.
- **إدارة الأعطال :**التعامل مع الأعطال المفاجئة دون توشقف النظام.
- **الأداء :**تحسين زمن الاستجابة وتقليل التأخير في نقل البيانات.

أبرز خصائص النظم الموزعة:

- **الاستقلالية :**تعمل كل عقدة (حاسوب) بشكل مستقل ولكن تتفاعل مع العقد الأخرى.
- **التواصل :**يتم تبادل البيانات بين العقد عبر شبكات مثل الإنترنت أو الشبكات المحلية.
- **التنسيق :**تحتاج العقد إلى تنسيق العمل لضمان تحقيق الهدف المشترك.
- **الشفافية :**يجب أن تبدو النظم الموزعة للمستخدم وكأنها نظام واحد غير معقد.

1. مقدمة إلى النظم الموزعة

تطبيقات النظم الموزعة

النظم الموزعة تُستخدم في مجموعة واسعة من التطبيقات التي أصبحت جزءًا لا يتجزأ من حياتنا اليومية، ومنها:

- **الحوسبة السحابية:** مثل Amazon Web Services (AWS) وMicrosoft Azure، والتي توفر خدمات تخزين البيانات وتشغيل التطبيقات عن بُعد.
- **أنظمة قواعد البيانات الموزعة:** تُستخدم في المؤسسات الكبيرة لتخزين وإدارة البيانات عبر مواقع متعددة مثل Google Spanner.
- **الشبكات الاجتماعية:** أنظمة مثل Facebook و Twitter تعتمد على النظم الموزعة لمعالجة كميات هائلة من البيانات في الوقت الفعلي.
- **التطبيقات المصرفية والمالية:** تُستخدم لإدارة البيانات المصرفية عبر فروع متعددة في مواقع جغرافية مختلفة.
- **الألعاب الجماعية عبر الإنترنت:** تعتمد على النظم الموزعة لضمان تجربة لعب سلسة بين المستخدمين في جميع أنحاء العالم.
- **أنظمة إنترنت الأشياء (IoT):** تُستخدم في ربط وتحليل البيانات القادمة من أجهزة الاستشعار المنتشرة في المدن الذكية أو المنازل الذكية.
- **محركات البحث:** مثل Google وBing، التي تعتمد على النظم الموزعة لتحليل وفهرسة الويب.

2. معمارية النظم الموزعة

معمارية النظم الموزعة تشير إلى التصميم الهيكلي الذي يحدد كيفية تنظيم المكونات المختلفة للنظام الموزع، وكيفية تفاعلها مع بعضها البعض لتحقيق الأهداف المرجوة. الهدف الأساسي هو تقديم أداء عالٍ، وقابلية للتوسع، ومرونة في التعامل مع الأخطاء، مع الحفاظ على التناسق والتوافر.

العناصر الرئيسية في معمارية النظم الموزعة:

1. **العقد (Nodes):** الأجهزة أو المكونات التي تعمل كوحدات مستقلة (مثل الخوادم أو أجهزة المستخدم).
2. **الشبكة (Network):** الوسيط الذي يربط العقد معًا لتبادل البيانات والمعلومات.
3. **بروتوكولات الاتصال:** القواعد التي تحدد كيفية تبادل البيانات بين العقد.
4. **آليات إدارة الموارد:** كيفية توزيع وإدارة الموارد المشتركة بين العقد المختلفة مثل المعالج أو الذاكرة.

مميزات معمارية النظم الموزعة:

- **الشفافية:** مثل شفافية الموقع والوصول، حيث يمكن للمستخدم الوصول إلى الموارد دون معرفة مكانها.
- **قابلية التوسع:** القدرة على إضافة عقد جديدة بسهولة.
- **التوافر العالي:** استمرار عمل النظام حتى في حالة فشل بعض المكونات.

2. معمارية النظم الموزعة

2.1 مفاهيم المعمارية الموزعة

- **الشفافية:(Transparency)**

قدرة النظم الموزعة على إخفاء تعقيد النظام عن المستخدمين، وتشمل:

- **شفافية الوصول :**تمكين المستخدم من الوصول إلى الموارد دون القلق حول مكانها أو كيفية الوصول إليها.
- **شفافية الموقع :**إخفاء موقع الموارد أو الخدمات في النظام.
- **شفافية النسخ :**إدارة نسخ البيانات بشكل تلقائي دون تدخل المستخدم.
- **شفافية التوازي :**السماح بتنفيذ العمليات بشكل متزامن.
- **شفافية التعطل :**الحفاظ على استمرارية العمل حتى في حالة حدوث فشل في بعض الأجزاء.

- **الاعتمادية:(Reliability) :** ضمان استمرارية النظام في تقديم الخدمات بشكل دقيق وموثوق، مع القدرة على التعامل مع الأخطاء والفشل.
- **التدرجية:(Scalability) :** قدرة النظام على النمو أو التوسع من خلال إضافة موارد أو تحسين الأداء مع زيادة عدد المستخدمين أو حجم البيانات.
- **التزامن:(Concurrency) :** تمكين العديد من العمليات من العمل معًا في نفس الوقت دون تعارض أو تأثير سلبي على الأداء.

2. معمارية النظم الموزعة

- **الاتصال بين العمليات:(Inter-Process Communication - IPC) :** توضيح كيفية تبادل البيانات والرسائل بين المكونات المختلفة للنظام الموزع باستخدام بروتوكولات واتصالات محددة.
- **التوحيد القياسي:(Standardization) :** اعتماد معايير وبروتوكولات مشتركة لتسهيل التفاعل بين الأنظمة المختلفة.
- **إدارة الموارد:(Resource Management) :** التعامل مع تخصيص الموارد وضمان استخدامها بكفاءة لتحقيق الأهداف المرجوة.

2.2 أنواع المعمارية في النظم الموزعة

- **معمارية الخوادم/العملاء:(Client/Server Architecture)**
  - **الوصف:**
    في هذا النوع، يتم تقسيم النظام إلى خادم مركزي يقدم الخدمات وعميل يقوم بطلب هذه الخدمات.
  - **المزايا :** سهولة الصيانة بسبب مركزية الخادم. إدارة الموارد بشكل مركزي.
  - **العيوب :** وجود نقطة فشل واحدة (الخادم). ضعف التدرجية عند زيادة عدد العملاء.
  - **أمثلة:** تطبيقات البريد الإلكتروني، أنظمة قواعد البيانات.

2. معمارية النظم الموزعة

- **معمارية النظراء/النظراء:(Peer-to-Peer - P2P)**
  - **الوصف:** لا يوجد خادم مركزي، وكل جهاز في الشبكة يعمل كنظير يمكنه العمل كعميل وخادم في نفس الوقت.
  - **المزايا :** لا توجد نقطة فشل واحدة. توزيع عبء العمل بين جميع العقد.
  - **العيوب :** إدارة الشبكة تصبح أكثر تعقيدًا. الأمن أقل مقارنة بالأنظمة المركزية.
- **معمارية متعددة الطبقات:(Multi-Tier Architecture)**
  - **الوصف:** يتم تقسيم النظام إلى طبقات، مثل طبقة العرض(Presentation Layer) ، وطبقة منطق الأعمال (Business Logic Layer)، وطبقة البيانات.(Data Layer)
  - **المزايا :** تحسين قابلية الصيانة. فصل المسؤوليات بين الطبقات.
  - **العيوب :** زيادة التعقيد. أداء أقل بسبب التفاعلات بين الطبقات.
  - **أمثلة:** تطبيقات الويب الحديثة.

2. معمارية النظم الموزعة

- **معمارية الخدمة الموجهة:(Service-Oriented Architecture - SOA)**
  - **الوصف:** تعتمد على خدمات مستقلة تقدم وظائف محددة يمكن الوصول إليها من خلال بروتوكولات محددة) مثلSOAP ، .REST).
  - **المزايا :** سهولة إعادة استخدام الخدمات. المرونة في التحديث والتوسيع.
  - **العيوب :** زيادة تعقيد إدارة الخدمات. يتطلب تصميمًا دقيقًا للتفاعل بين الخدمات.

أمثلة: التطبيقات السحابية، الخدمات المصرفية عبر الإنترنت.

- **معمارية الأحداث الموجهة:(Event-Driven Architecture)**
  - **الوصف :**تعتمد على معالجة الأحداث حيث يتم تنفيذ العمليات بناءً على الأحداث التي يتم إنشاؤها.
    - **المزايا :** استجابة فورية للأحداث. مرونة في التصميم.
    - **العيوب :** صعوبة تصحيح الأخطاء. يتطلب تصميمًا معقدًا.
  - **أمثلة:** أنظمة الحجز الفوري، تطبيقات الرسائل الفورية.

أمثلة على معمارية النظم الموزعة

1. **نظام البريد الإلكتروني:(Email System)**
   - **الوصف:** يعتمد على معمارية الخادم/العميل .(Client/Server) يتم إرسال الرسائل من جهاز العميل عبر بروتوكولات مثل SMTP إلى خادم البريد، ومن هناك يتم توزيعها إلى مستلمين عبر IMAP أو.POP3
   - **التطبيقات الشائعة:** Gmail، Outlook، Yahoo Mail.
2. **شبكات مشاركة الملفات:(File Sharing Networks)**
   - **الوصف:** تعتمد على معمارية النظراء/النظراء .(Peer-to-Peer - P2P) في هذه الشبكات، يتشارك المستخدمون الملفات مباشرة دون الحاجة إلى خادم مركزي.
   - **التطبيقات الشائعة:** BitTorrent، LimeWire.
3. **التجارة الإلكترونية:(E-Commerce Systems)**
   - **الوصف:** تعتمد على معمارية متعددة الطبقات .(Multi-Tier Architecture) تتضمن هذه الأنظمة طبقات منفصلة للتعامل مع واجهة المستخدم، منطق الأعمال، وقواعد البيانات.

التطبيقات الشائعة: Amazon، eBay، Shopify

أمثلة على معمارية النظم الموزعة

4. **منصات البث:(Streaming Platforms)**
   - **الوصف:**
     تعتمد على معمارية الخدمات المصغرة .(Microservices Architecture) يتم تقسيم النظام إلى خدمات صغيرة مثل إدارة الحسابات، توفير المحتوى، معالجة المدفوعات.
   - **التطبيقات الشائعة:**
     YouTube. ،Spotify ،Netflix
5. **أنظمة الحجز الفوري:(Real-Time Booking Systems)**
   - **الوصف:**
     تعتمد على معمارية الأحداث الموجهة .(Event-Driven Architecture) يتم تحديث حالة الحجوزات أو التذاكر بشكل فوري بناءً على الأحداث.
   - **التطبيقات الشائعة:**
     أنظمة حجز الرحلات الجوية والفنادق مثل Booking.com و.Skyscanner

3. الاتصال في النظم الموزعة

3.1 مفهوم الاتصال في النظم الموزعة: الاتصال في النظم الموزعة يشير إلى تبادل البيانات والمعلومات بين العقد (Nodes) المختلفة.

3.2 آليات الاتصال في النظم الموزعة

1. **الاتصال المتزامن:(Synchronous Communication) :** في هذا النوع من الاتصال، يجب أن يتم التبادل بين العقد في الوقت الفعلي. يحتاج كل طرف في الاتصال إلى انتظار رد من الطرف الآخر قبل أن يتمكن من المضي قدمًا في عملية معينة.
2. **الاتصال غير المتزامن:(Asynchronous Communication) :** في الاتصال غير المتزامن، يمكن للعقد إرسال الرسائل دون انتظار رد فوري. يتم معالجة الردود في وقت لاحق عندما تكون العقد مستعدة.
3. **الاتصال البثي:(Broadcast Communication) :** في هذا النوع من الاتصال، يتم إرسال البيانات إلى جميع العقد في الشبكة. يعتبر مفيدًا عندما تحتاج جميع العقد في النظام إلى تلقي نفس المعلومات في وقت واحد.
4. **الاتصال الموجه:(Unicast Communication) :** هذا النوع من الاتصال يشير إلى إرسال البيانات من عقدة إلى أخرى بشكل فردي. تُستخدم هذه التقنية عندما تكون الرسائل موجهة إلى عقدة واحدة فقط.

3. الاتصال في النظم الموزعة

3.3 بروتوكولات الاتصال في النظم الموزعة

1. **بروتوكول:TCP/IP :** يُعتبر البروتوكول الأساسي للاتصال في معظم النظم الموزعة، حيث يوفر طريقة موثوقة وفعالة لنقل البيانات بين العقد.
   - **الميزات :**يوفر ضمانات لتسليم البيانات في ترتيبها الصحيح ويمكن استخدامه عبر شبكات الإنترنت.
2. **بروتوكولات:(Remote Procedure Call) RPC :** تمكن هذه البروتوكولات العقد من تنفيذ الإجراءات عن بُعد كما لو كانت تُنفذ محليًا. يتم استخدام RPC بشكل واسع في النظم الموزعة لتسهيل عملية الاتصال بين الخوادم والعملاء.
   - **الميزات :**تسهل بناء تطبيقات النظم الموزعة المعقدة من خلال تقديم واجهة متسقة للوصول إلى الخدمات عن بُعد.
3. **بروتوكولات الرسائل:(Message-Oriented Middleware - MOM) :** تستخدم هذه البروتوكولات في الأنظمة التي تتطلب تبادل رسائل بين العقد. توفر هذه الأنظمة طبقة من الوساطة بين العقد بحيث يمكن إرسال رسائل أو أحداث بدون الحاجة إلى معرفة الحالة الداخلية للعقدة المستقبلية.
   - **الميزات :**توفر مرونة عالية في الاتصال بين العقد وتدعم كلًا من الاتصال المتزامن وغير المتزامن.

3. الاتصال في النظم الموزعة

3.4 تحديات الاتصال في النظم الموزعة

1. **التأخير في الشبكة:(Network Latency)**
   - وجود فترات تأخير بين إرسال واستقبال البيانات بسبب المسافات الجغرافية أو ازدحام الشبكة قد يؤثر على أداء النظام بشكل عام.
2. **الفشل في العقد:(Node Failures)**
   - في النظم الموزعة، قد تتعرض العقد إلى الفشل، مما يؤدي إلى فقدان الاتصال بين بعض الأجزاء في النظام. من الضروري التعامل مع هذه الحالات باستخدام آليات للتعافي من الفشل.
3. **التزامن والتناسق:(Synchronization and Consistency)**
   - ضمان التزامن بين العقد المختلفة بحيث تكون البيانات متناسقة بين جميع العقد في وقت واحد يعتبر تحديًا كبيرًا في النظم الموزعة.
4. **الأمان:(Security) :**
   - ضمان أمان الاتصال بين العقد أمر حيوي. قد تتعرض البيانات أثناء النقل للهجمات مثل التنصت أو التلاعب. يجب استخدام تقنيات مثل التشفير لضمان الأمان في الشبكة.

التزامن في نقل البيانات

1. **التزامن (Synchronization) في النظم الموزعة:** التزامن في نظم الاتصالات الموزعة يعنى التنسيق بين عمليات مختلفة (مثل الخوادم أو الأجهزة) لضمان إجراء العمليات بشكل صحيح ومتسق عبر النظام، حتى وإن كانت هذه العمليات تعمل في بيئات أو مواقع جغرافية مختلفة.

2. **أهمية التزامن في نقل البيانات:**

- **التنسيق بين العمليات :**يساعد التزامن في ضمان أن العمليات التي تجري في النظام الموزع تعمل معًا بشكل متسق، مما يقلل من حدوث الأخطاء أو التصادمات بين البيانات.
- **ضمان التوقيت الصحيح :**التزامن يضمن أن البيانات التي يتم إرسالها واستلامها تكون في الوقت المناسب، مما يعزز من كفاءة النظام.
- **تحقيق الاستقرار :**إذا لم يكن هناك تزامن، يمكن أن يحدث تعارض بين العمليات أو فشل في حفظ البيانات بشكل صحيح، مما يؤدي إلى تدهور أداء النظام.

3. **أنواع التزامن في نقل البيانات:**

- **التزامن التام :(Hard Synchronization)** في هذا النوع، يتم تحديد التوقيت بالضبط بين العقد الموزعة. يتم نقل البيانات أو إتمام العمليات فقط عندما تكون جميع العقد جاهزة.
- **التزامن المرن :(Soft Synchronization)** في هذا النوع، يتم السماح بمرونة أكبر في التوقيت بين العقد. قد تكون بعض العمليات متزامنة ولكن ليس بشكل صارم، مما يتيح للعقد العمل بشكل مستقل إلى حد ما.

التزامن في نقل البيانات

4. طرق التزامن في نقل البيانات:

- **التزامن باستخدام الأقفال (Locks):** تستخدم هذه الطريقة لتنظيم الوصول إلى الموارد المشتركة، مثل قاعدة بيانات أو ملف، بحيث لا يمكن للعمليات المتعددة الوصول إليها في نفس الوقت. يتم استخدام أقفال لضمان أن كل عملية تحصل على وصول حصري إلى الموارد.
  - **أقفال الكتابة (Write Locks):** تمنع العمليات الأخرى من الكتابة على نفس المورد.
  - **أقفال القراءة (Read Locks):** تسمح بقراءة البيانات بواسطة العديد من العمليات ولكن تمنع الكتابة.
- **التزامن باستخدام الرسائل (Message Passing):** في هذه الطريقة، يتم إرسال رسائل بين العقد المختلفة للتنسيق بين العمليات. يمكن أن تحتوي الرسائل على إشارات حول جاهزية العقد أو معلومات حول حالة العمليات، مما يساعد في التنسيق بين العمليات.
- **التزامن باستخدام الساعات (Clock Synchronization):** يتطلب التزامن بين العقد في النظام الموزع مزامنة الساعات بين العقد المختلفة بحيث تتمكن كل عقدة من معرفة الوقت الحقيقي للآخرين. مثال على ذلك هو **خوارزمية نِستَر (NTP)** التي تستخدم لتزامن الوقت بين الأنظمة المختلفة.
- **التزامن باستخدام قاعدة البيانات (Database Synchronization):** في نظم الاتصالات الموزعة، قد تكون قاعدة البيانات هي المصدر الرئيسي لتخزين البيانات. وبالتالي، يتم تزامن البيانات عبر قاعدة البيانات الموزعة لضمان أن جميع النسخ متسقة.

التزامن في نقل البيانات

5. مشاكل التزامن في نقل البيانات:

- **التصادمات (Collisions):** عندما تحاول عدة عمليات الكتابة أو القراءة من نفس المورد في وقت واحد، يمكن أن يحدث تصادم بين العمليات، مما يؤدي إلى أخطاء أو بيانات غير متسقة.
- **الحجب (Deadlocks):** يحدث الحجب عندما تقوم عمليتان أو أكثر بإغلاق مورد معين ولا تستطيع أي منهما الاستمرار بسبب انتظار الأخرى.
- **السباقات (Race Conditions):** يحدث السباق عندما تعتمد عملية على التوقيت والآلية التي يتم بها التزامن، مما يؤدي إلى حدوث أخطاء إذا كانت العمليات تعمل بسرعة غير متوقعة أو لا يتم التنسيق بينها بالشكل الصحيح.

6. حلول التزامن في نقل البيانات:

- **استخدام الخوارزميات الموزعة:** يمكن تطبيق خوارزميات موزعة مثل خوارزمية **Lamport's Logical Clocks** أو **Vector Clocks** لتنسيق العمليات وتحديد ترتيب الأحداث في النظم الموزعة.
- **خوارزميات تسوية التوقيت (Clock Synchronization Algorithms):** مثل خوارزمية **Berkeley** أو **NTP**، التي تستخدم لمزامنة الوقت بين العقد الموزعة بشكل فعال.

التكرار (Replication): استخدام التكرار لنسخ البيانات عبر العقد المختلفة، حيث يمكن لكل عقدة التحقق من البيانات بشكل مستقل وتحديثها إذا كانت هناك أي تغييرات

إدارة الخطأ والموثوقية في الاتصال في النظم الموزعة

1. **مفهوم إدارة الخطأ في النظم الموزعة:** إدارة الخطأ تشير إلى استراتيجيات وتقنيات يتم استخدامها للكشف عن الأخطاء والتعامل معها في النظم الموزعة.

2. **أنواع الأخطاء في النظم الموزعة:**

- **أخطاء الاتصال:(Communication Errors) :** تحدث عند فقدان الاتصال بين العقد بسبب مشكلات في الشبكة مثل التأخير الكبير أو انقطاع الاتصال.
- **أخطاء في البيانات:(Data Errors) :** تحدث عند تلف أو فقدان البيانات أثناء النقل بين العقد. يمكن أن يؤدي ذلك إلى فقدان المعلومات أو استقبال بيانات غير صحيحة.
- **أخطاء في التزامن:(Synchronization Errors) :** تحدث عند حدوث عدم توافق في التوقيت بين العقد الموزعة، مما يؤدي إلى تعارض بين العمليات أو تنفيذ غير دقيق.
- **أخطاء النظام:(System Failures) :** تتضمن فشل الخوادم أو المحطات القاعدية أو الأنظمة التي تتحكم في تدفق البيانات. يمكن أن يؤدي ذلك إلى انهيار النظام إذا لم يتم التعامل معه بشكل صحيح.

إدارة الخطأ والموثوقية في الاتصال في النظم الموزعة

3. آليات إدارة الخطأ في النظم الموزعة:

- **إعادة الإرسال التلقائي (Automatic Retransmission):** في حال فقدان البيانات أو حدوث خلل في الاتصال، يمكن إعادة إرسال البيانات تلقائيًا لضمان عدم فقدان المعلومات. تستخدم هذه التقنية بشكل شائع في بروتوكولات النقل مثل **TCP**.
- **اكتشاف الأخطاء (Error Detection):** تعتمد النظم الموزعة على تقنيات الكشف عن الأخطاء مثل **Checksum** و**Cyclic Redundancy Check (CRC)** للكشف عن الأخطاء التي قد تحدث أثناء نقل البيانات بين العقد.
- **إعادة بناء البيانات (Data Recovery):** بعد اكتشاف الخطأ، يتم تطبيق تقنيات مثل **التكرار** (Replication)لضمان عدم فقدان البيانات عن طريق إنشاء نسخ متعددة للبيانات على عقد مختلفة. يمكن أن تساعد هذه النسخ في استعادة البيانات عند حدوث أخطاء في أي جزء من النظام.
- **التكرار (Replication):** باستخدام التكرار، يتم تخزين نسخة من البيانات على أكثر من عقدة واحدة في النظام الموزع. إذا فشلت إحدى العقد، يمكن استرجاع البيانات من العقد الأخرى لضمان استمرارية الخدمة.
- **البروتوكولات المتسامحة مع الأخطاء (Fault-Tolerant Protocols):** تهدف هذه البروتوكولات إلى استيعاب الأخطاء دون التأثير الكبير على أداء النظام. على سبيل المثال، بروتوكولات مثل **Paxos** و**Raft** تهدف إلى ضمان الاستمرارية والتنسيق بين العقد حتى في وجود أخطاء.

إدارة الخطأ والموثوقية في الاتصال في النظم الموزعة

4. ضمان الموثوقية في نقل البيانات:

موثوقية النقل في النظم الموزعة تعتمد على ضمان وصول البيانات بشكل صحيح من المصدر إلى الوجهة بدون فقدان أو تشويه.

- **التأكيد على استلام البيانات (Acknowledgement):** إحدى الطرق المستخدمة لضمان الموثوقية هي استخدام التأكيدات (Acknowledgements)من العقد المستقبلة التي تؤكد أن البيانات قد تم استلامها بنجاح. في حال لم يتم استلام التأكيد في وقت محدد، يتم إرسال البيانات مرة أخرى.
- **النسخ المتعدد (Multiple Replication):** تستخدم العديد من الأنظمة الموزعة النسخ المتعددة للبيانات لضمان موثوقية أكبر. إذا فشلت إحدى النسخ أو العقد، يتمكن النظام من استخدام النسخ الأخرى لاستعادة البيانات.
- **ضمان الاتصال الثابت (Stable Connection):** تعد الحفاظ على الاتصال الثابت بين العقد أساسية لضمان الموثوقية، حيث أن الانقطاع المفاجئ أو التأخير الكبير في الاتصال قد يسبب فقدان البيانات أو حدوث تعارضات.

الأكاديمية العربية الدولية
Arab International Academy

أسئلة

1. ما هو تعريف النظم الموزعة؟
2. لماذا تعتبر النظم الموزعة مهمة وما هي تطبيقاتها؟
3. ما هي الأنواع الرئيسية لمعمارية النظم الموزعة؟
4. ما هي بروتوكولات الاتصال الشائعة في النظم الموزعة؟
5. ما هي التحديات في تصميم النظم الموزعة؟

الأجوبة

ج1: النظم الموزعة هي أنظمة تتكون من عدة عقد متصلة بشبكة تعمل بشكل متزامن لتحقيق هدف مشترك، مما يتيح مشاركة الموارد والخدمات.

ج2: أهميتها تكمن في تقديم أداء عالٍ، وتوفير موارد مشتركة، وزيادة التوافر. تستخدم في الحوسبة السحابية، تطبيقات الإنترنت، وإنترنت الأشياء.

ج3:

- خوادم/عملاء: تعتمد على تفاعل الأجهزة العميلة مع الخوادم.
- النظراء/النظراء: جميع الأجهزة متساوية في المهام.
- المعمارية الهجينة: تجمع بين النوعين.

ج4:

- **HTTP:** لنقل البيانات عبر الويب.
- **RPC:** لاستدعاء الإجراءات عن بُعد.
- **RMI:** لاستدعاء الأساليب في الكائنات البعيدة.

ج5:

- التزامن بين العقد.
- معالجة الأخطاء وضمان الموثوقية.
- التعامل مع التداخل وضمان الأمان.

النماذج في النظم الموزعة

- **النموذج العمودي:(Vertical Model)**
  - يعتمد هذا النموذج على تقسيم الأنظمة إلى طبقات متعددة، حيث يتم فصل مكونات النظام الموزع إلى طبقات متعددة تتعامل مع الوظائف المختلفة. هذه الطبقات قد تشمل طبقة الشبكة، طبقة الاتصال، وطبقة البيانات، مما يساهم في عزل الوظائف وتقليل التعقيد.
- **النموذج الأفقي:(Horizontal Model)**
  - في هذا النموذج، يتم التعامل مع عقد متعددة (موزعة) تؤدي نفس الوظيفة، مثل الخوادم الموزعة التي تؤدي نفس الدور في معالجة البيانات أو تخزين المعلومات.
- **نموذج العميل والخادم:(Client-Server Model)**
  - يعتمد هذا النموذج على فصل الوظائف بين العميل والخادم. العميل هو الجهة التي تطلب الخدمات بينما الخادم هو الذي يقدم هذه الخدمات. يمكن أن يكون الخادم عبارة عن عقدة خدمية توفر المعالجة أو التخزين، بينما يعمل العميل كواجهة تفاعلية للمستخدم.
- **نموذج النظراء:(Peer-to-Peer Model)**
  - في هذا النموذج، كل عقدة (جهاز أو كمبيوتر) في الشبكة تعمل كعميل وخادم في ذات الوقت. العقدة يمكنها أن تطلب الخدمات وأيضاً تقديم الخدمات لعقد أخرى في الشبكة.

الأنماط في النظم الموزعة

- **نمط الاسترداد من الأخطاء:(Fault Tolerance Pattern)**
  - هذا النمط يهدف إلى توفير طريقة للنظام لمواصلة العمل بشكل صحيح على الرغم من حدوث أعطال أو أخطاء في أجزاء منه. يتم استخدام تقنيات مثل **التكرار** والتوزيع لضمان استمرارية العمل.
  - مثال: استخدام **Paxos** و **Raft**لبروتوكولات الاسترداد من الأخطاء في نظم قاعدة البيانات الموزعة.
- **نمط التوزيع المتوازي:(Parallelism Pattern)**
  - يهدف هذا النمط إلى توزيع العمل عبر عدة عقد في الشبكة بهدف تسريع العمليات وتحسين الأداء. يتم تقسيم المهمة إلى مهام أصغر ثم توزيعها بين عدة عقد، ثم دمج النتائج.
  - مثال **MapReduce** :في أنظمة معالجة البيانات الكبيرة مثل **Hadoop**.
- **نمط التزامن:(Synchronization Pattern) :** في النظم الموزعة، يجب على العقد المختلفة أن تتعاون وتتزامن من أجل إتمام العمليات بشكل صحيح. هذا النمط يساعد في تقليل التداخل وضمان تنفيذ العمليات بالترتيب الصحيح.
  - مثال :**المنطق المتزامن** عبر **المزامنة باستخدام الأقفال** (Locks)أو **الخوارزميات الموزعة** مثل **Paxos**و **Lamport Timestamps**.

الأنماط في النظم الموزعة

- **نمط الاستعلامات:(Query Pattern)**
  - هذا النمط يُستخدم لتنظيم كيفية استرجاع البيانات عبر النظام الموزع بشكل متسق وفعّال. الهدف هو توفير طريقة سهلة لتنفيذ الاستعلامات عبر عدة عقد وضمان الحصول على النتائج بسرعة.
  - مثال :**نمط قواعد البيانات الموزعة** حيث يتم تقسيم البيانات عبر العديد من العقد ويُستخدم خوارزميات معينة لتوزيع الاستعلامات عبر النظام.
- **نمط التخزين الموزع:(Distributed Storage Pattern)**
  - يهدف هذا النمط إلى إدارة البيانات عبر عدة عقد بحيث يتم تخزين البيانات بشكل موزع ولكن يمكن الوصول إليها بشكل شفاف كما لو كانت على عقدة واحدة.
  - مثال :**أنظمة التخزين السحابي** مثل **Amazon S3**، حيث يتم تخزين الملفات في مراكز بيانات مختلفة، ويظل الوصول إلى هذه الملفات موحداً للمستخدم.

النموذج المتمركز حول الكائن Object-based Model

1. مفهوم الكائن في النموذج المتمركز حول الكائن

- **الكائن (Object):** هو الكائن الأساسي في هذا النموذج، والذي يحتوي على بيانات (السمات أو الخصائص) ودوال (الوظائف أو العمليات) تعبر عن السلوكيات.
- في البرمجة الكائنية، يُعتبر كل كائن بمثابة وحدة من البيانات التي تحتوي على خصائص وسلوكيات مترابطة.
- يتم تنظيم الكائنات في هيكل شجري أو هرمي وفقًا للعلاقات بينها.

2. خصائص النموذج المتمركز حول الكائن:

- **التغليف (Encapsulation):** يتم تجميع البيانات والعمليات التي تعمل عليها داخل الكائن. هذا يعزل الكائن عن باقي النظام ويسمح بالتحكم في الوصول إلى البيانات.
- **الوراثة (Inheritance):** يسمح هذا النموذج للكائنات بأن ترث خصائص وسلوكيات الكائنات الأخرى. يساعد في إعادة استخدام الكود وتوسيع النظام بسهولة.
- **التعددية الشكلية (Polymorphism):** يسمح للكائنات بتقديم سلوكيات متعددة استنادًا إلى السياق أو النوع الفرعي للكائن.
- **الاستقلالية (Independence):** كل كائن يمكن أن يعمل بشكل مستقل عن الكائنات الأخرى، مما يساهم في توزيع المهام بين الكائنات عبر الشبكة في نظم موزعة.

النموذج المتمركز حول الكائن Object-based Model

3. كيفية العمل في النظم الموزعة:

- يتم استخدام **البرمجة الموزعة** لتوزيع الكائنات بين العقد في النظام. كل عقدة يمكن أن تحتوي على كائنات متعددة تعمل في الوقت ذاته.
- **التفاعل بين الكائنات** يتم عبر بروتوكولات الاتصال مثل **RMI (Remote Method Invocation)** أو **CORBA (Common Object Request Broker Architecture)** التي تسمح بالكائنات البعيدة بالتفاعل كما لو كانت محلية.
- **النداء البعيد (Remote Invocation)**: عندما يريد كائن من عقدة معينة استدعاء عملية على كائن موجود في عقدة أخرى، يتم استخدام تقنيات مثل RMI أو SOAP لتمرير الطلبات عبر الشبكة.

4. مزايا النموذج المتمركز حول الكائن:

- **إعادة استخدام الكود**: من خلال الوراثة، يمكن إنشاء كائنات جديدة بناءً على كائنات موجودة، مما يقلل من تكرار الكود.
- **إدارة معقدة للبيانات**: يتيح هذا النموذج تمثيل البيانات المعقدة بشكل طبيعي وسلس، مما يجعل من السهل التعامل مع البيانات في نظم موزعة.
- **التفاعل السلس**: يسمح النموذج بتفاعل سلس بين الكائنات المتوزعة على الشبكة، مما يسهل بناء أنظمة موزعة عالية الأداء.
- **التوسع**: يمكن إضافة كائنات جديدة بسهولة دون التأثير على الكائنات الأخرى في النظام.

النموذج المتمركز حول البيانات Data-based Model

1. مفهوم النموذج المتمركز حول البيانات:

- **البيانات** هي العنصر الأساسي الذي يتم التعامل معه في هذا النموذج، ويُعتبر الوصول إلى هذه البيانات وتعديلها هو الهدف الرئيسي.
- يركز هذا النموذج على فصل البيانات عن العمليات التي قد تتم عليها، بحيث يتم توفير قاعدة بيانات مركزية أو موزعة تدير البيانات، بينما يتم تنفيذ العمليات عليها بواسطة أنظمة أو خدمات أخرى.

2. خصائص النموذج المتمركز حول البيانات:

- **التخزين المركزي أو الموزع للبيانات** :يتم تخزين البيانات في قاعدة بيانات مركزية أو موزعة، مما يتيح الوصول إلى البيانات من قبل مكونات النظام المختلفة عبر الشبكة.
- **الوصول غير المتزامن** :يتسم النظام في هذا النموذج بأنه يمكن أن يكون الوصول إلى البيانات غير متزامن، حيث يتمكن العملاء من الوصول إلى البيانات دون الحاجة إلى الانتظار للحصول على استجابة فورية.
- **الاستقلالية** :يمكن أن تكون البيانات مستقلة عن العمليات، مما يسمح بالتطوير المرن للنظام بحيث يمكن تغيير العمليات دون التأثير على البيانات المخزنة.
- **التكرار والنسخ** :من الممكن نسخ البيانات عبر مواقع متعددة لزيادة توافرها وتحقيق التكرار في حالة حدوث مشاكل في بعض العقد.

النموذج المتمركز حول البيانات Data-based Model

3. كيفية العمل في النظم الموزعة:

- يتم تنظيم البيانات بشكل مركزي أو موزع في قاعدة بيانات يمكن الوصول إليها عبر الشبكة.
- **المعاملات الموزعة** :يمكن أن يتم الوصول إلى البيانات بواسطة عملاء متعددين عبر شبكة من العقد الموزعة.
- يتم تبادل البيانات بين العقد باستخدام بروتوكولات مثل **SQL**أو **NoSQL**، وقد يتم استخدام أدوات مثل **MapReduce**لمعالجة البيانات على نطاق واسع.

4. أنواع البيانات في النموذج المتمركز حول البيانات:

- **البيانات الهيكلية** :بيانات منظمة في جداول أو أشكال منسقة) مثل قواعد بيانات.(SQL
- **البيانات غير الهيكلية** :بيانات غير منظمة مثل النصوص والصور والفيديو، التي يمكن تخزينها في نظم قواعد بيانات NoSQL أو أنظمة التخزين المتوزعة.
- **البيانات شبه الهيكلية** :بيانات تمثل مزيجًا بين الهيكلية وغير الهيكلية، مثل **XML**أو .**JSON**

الأنماط الشائعة في النظم الموزعة

1. نمط:MapReduce

- **الفكرة الأساسية** :يقوم **MapReduce**بتقسيم المهمة الكبيرة إلى جزئين رئيسيين: مرحلة الـ **Map**ومرحلة الـ **Reduce**.
  - **مرحلة الـ Map**: في هذه المرحلة، يتم تقسيم البيانات إلى أجزاء صغيرة (Chunks) مع معالجتها بشكل مستقل على الخوادم المختلفة. تُنفذ دالة Mapعلى كل جزء من البيانات، وعادة ما يتم تحويل البيانات إلى شكل.(Key, Value)
  - **مرحلة الـ Reduce**: بعد أن يتم معالجة البيانات في مرحلة الـ **Map**، يتم تجميع النتائج في مرحلة الـ **Reduce**. تقوم دالة Reduce بتجميع القيم المتشابهة بناءً على المفاتيح (Keys) لتوليد النتيجة النهائية.

مثال :إذا كانت لدينا مجموعة من السجلات التي تحتوي على كلمات في مستندات مختلفة، يمكن استخدام **MapReduce**لحساب عدد مرات تكرار كل كلمة في جميع المستندات. في مرحلة الـ Map، يتم تحويل الكلمات إلى شكل (كلمة، 1)، ثم في مرحلة الـ Reduce، يتم تجميع القيم المتشابهة (أي الكلمات) وحساب العدد الإجمالي لكل ك

الأنماط الشائعة في النظم الموزعة

2. نمط إرسال الاستعلامات:(Query Processing)

- يُستخدم هذا النمط لتنفيذ استعلامات معقدة عبر البيانات الموزعة. يُوزع الاستعلام عبر العقد المختلفة لتوزيع عبء العمل، ثم يتم تجميع النتائج مرة أخرى في نقطة واحدة.
- **التحدي** :التحدي الرئيس في هذا النمط هو إدارة تجميع البيانات من العقد المختلفة وضمان توافق النتائج.

3. نمط مدير المهام:(Task Manager Model)

- في هذا النمط، يتم تقسيم المهمة إلى مهام صغيرة، ثم يتم إدارة هذه المهام بشكل مركزي بواسطة مدير المهام.
- **المزايا** :تحسين توزيع المهام بين العقد والمكونات المختلفة في النظام. كما يساعد على تقليل التأخير عند تنفيذ المهام.
- **مثال** :يمكن أن يتعامل هذا النمط مع تطبيقات مثل إدارة مهام معقدة أو معالجة المعاملات في نظم الأعمال الموزعة.

الأنماط الشائعة في النظم الموزعة

4. نمط المعاملات الموزعة:(Distributed Transactions Model)

- يُستخدم هذا النمط في النظم الموزعة التي تتطلب توافقًا بين الأنظمة المختلفة عند تنفيذ المعاملات. يعتمد هذا النمط على آليات مثل بروتوكولات **Two-Phase Commit**و **Three-Phase Commit**.
- **التحدي** :ضمان التزامن عبر جميع العقد لضمان صحة المعاملات في بيئات معقدة ومتوزعة.
- **مثال** :المعاملات المالية أو الحسابات التي تحتاج إلى تنسيق بين عدة قواعد بيانات في نظم موزعة.

5. نمط المراجعة والمزامنة:(Consensus and Synchronization Model)

- يُستخدم هذا النمط في النظم الموزعة لضمان توافق العقد المختلفة في النظام والتأكد من أن العمليات التي تتم على البيانات تتوافق وتتم بشكل متسق.
- **التحديات** :ضمان التزامن بين العقد المتعددة، ويشمل بروتوكولات مثل **Paxos**و **Raft**، التي تهدف إلى حل مشكلات الاتساق والتزامن في البيئات الموزعة.
- **مثال** :التحقق من صحة البيانات أو القرارات في نظم موزعة مثل قواعد بيانات متوازية أو شبكات توزيع للبيانات.

الأنماط الشائعة في النظم الموزعة

6. نمط التخزين الموزع:(Distributed Storage Model)

- يُستخدم هذا النمط في النظم الموزعة لتخزين البيانات عبر العديد من العقد أو الخوادم المتعددة بحيث يمكن الوصول إليها بطريقة موزعة ومرنة.
- **المزايا** :يوفر هذا النمط تحسينات في الأداء والموثوقية، حيث يمكن تخزين البيانات في مواقع متعددة بحيث يكون هناك نسخ احتياطية في حال حدوث فشل في العقد أو الخوادم.
- **مثال** :الأنظمة مثل (Hadoop Distributed File System) **HDFS** أو **Amazon S3** تستخدم هذا النمط لتوزيع البيانات.

7. نمط شبكات من العقد المتوازية:(Peer-to-Peer Networks)

- في هذا النموذج، لا يوجد خادم مركزي؛ بل يتم توزيع المهام بين العقد (أو الأقران) في الشبكة. تُعتبر العقد جميعها متساوية وتتحمل أعباء معالجة البيانات.
- **المزايا** :يُعد هذا النمط مرنًا ويعزز من قدرة التحمل والتوافر العالي. كما يسمح للمستخدمين بالوصول إلى الموارد بسهولة دون الحاجة إلى نقطة مركزية.
- **التطبيقات** :هذا النموذج يستخدم في تطبيقات مثل **BitTorrent** أو **شبكات العمل الجماعي**.

الأنماط الشائعة في النظم الموزعة

8. نمط الخوادم الموزعة:(Distributed Server Model)

- في هذا النموذج، يتم تقسيم الخوادم بحيث تتحمل كل واحدة منها جزءًا من الحمل الوظيفي، مثل الخوادم التي تتعامل مع قاعدة بيانات معينة أو تقدم خدمة خاصة.
- **المزايا** :يمكن تحقيق الأداء العالي من خلال التوزيع الجغرافي والتوافر العالي.
- **مثال** :خوادم الويب الموزعة التي تتعامل مع الطلبات الواردة من مختلف المواقع الجغرافية.

9. نمط البيانات الموزعة:(Distributed Data Model)

- يتعلق هذا النموذج بتوزيع البيانات على عدة خوادم أو عقد، حيث تتوزع البيانات أو تُجزأ بين العقد المختلفة.
- **المزايا** :يتيح إمكانية الوصول إلى البيانات بسرعة أكبر، ويوفر أيضًا استمرارية الخدمة في حالة تعطل أحد الخوادم.
- **التطبيقات** :مثل **Big Data**، حيث يتم توزيع البيانات بشكل موحد بين العديد من الخوادم لتسهيل الوصول إليها.

الأكاديمية العربية الدولية
Arab International Academy

أسئلة

1. ما هو النموذج المتمركز حول الكائن؟
2. ما الذي يميز النموذج المتمركز حول البيانات؟
3. ما هو دور نمط MapReduce في النظم الموزعة؟
4. كيف يتم استخدام النموذج المتمركز حول الكائن في التطبيقات الموزعة؟
5. ما هي الفوائد الرئيسية لاستخدام الأنماط مثل MapReduce؟

الأجوبة

1. هو نموذج يعتمد على تمثيل البيانات والخدمات ككائنات يتم التفاعل معها من خلال واجهات محددة.
2. يركز على إدارة البيانات الموزعة وتبادلها، حيث تكون البيانات هي المحور الرئيسي للعمليات.
3. يوفر طريقة لمعالجة كميات كبيرة من البيانات عن طريق تقسيم المهام إلى مراحل " "Mapو""Reduce، مما يسهل الحوسبة الموزعة.
4. يُستخدم لإنشاء تطبيقات تعتمد على الكائنات، مثل الأنظمة القائمة على RPCأو RMI، حيث تسهّل التفاعل بين العقد.

5.

- تحسين أداء معالجة البيانات الكبيرة.
- تقليل زمن المعالجة عبر التوازي.
- تبسيط التعامل مع البيانات الموزعة.

مصادر خارجية

عنوان الفيديو	الرابط
النظم الموزعة	https://www.youtube.com/watch?v=skbKiPP7lhI&list=PLZfmlNL0D6Iw64rkpsZE0s4EcIV5X5Ado
الأنظمة الموزعة	https://www.youtube.com/watch?v=NJkavF-jBdc&list=PL1X65tGfsa62ipyFCaBc0Fzzlqcr7unfT
النظم الموزعة	https://www.youtube.com/watch?v=RTdpGvrpWOI&list=PLgAqrVq84PDcg55xnbUBHuLS8tWul6-kF

المراجع

- "Distributed Systems: Principles and Paradigms" لـ Andrew S. Tanenbaum.
- "Designing Data-Intensive Applications" لـ Martin Kleppmann.

الأكاديمية العربية الدولية
Arab International Academy

شكرا لكم