نظرة عامة على ريديس الجزء الأول

الريديس (Redis) هو عبارة عن مخزن لتخزين البيانات على شكل أزواج من مفتاح-قيمة Key-Value في الذاكرة الرئيسية In-Memory وهو المخزن الأكثر شيوعاً الذي يستخدم هذه التقنية. كما أن جميع شركات تكنولوجيا المعلومات في أنحاء العالم تستخدم هذه التقنية، فخدمة Elastic Cache التي تقدمها شركة أمازون تدعم هذه تقنية ريدس مما يجعلها واحدة من أقوى وأشهر تقنيات تخزين البيانات. وسوف نقدم في هذا المقال شرحاً مختصراً عن بنية هذه التقنية.

ماذا نقصد بـ In-Memory, Key-Value store؟

إن **Key-Value storage** عبارة عن نظام تخزين يتم فيه تخزين البيانات على شكل أزواج من المفاتيح والقيم **Key-Value pairs**،

تخزّن هذه الأزواج في الذاكرة الرئيسية **RAM** وهذا ما نقصده بـ **In-Memory**، وبهذا يمكننا القول أن تقنية ريدس تخزن البيانات في الذاكرة الرئيسية على شكل أزواج من المفاتيح والقيم.

يكون المفتاح في هذه التقنية عبارة عن محرف **String**، أما القيمة فيمكن أن تكون محرفاً أو قائمة **List** أو مجموعة أو مجموعة مخزنة أو مزيجاً منهم. كما في المثال التالي:

```
name="narayan"
["profession=["web", "mobile
```

المفتاح في حالتنا هذه هو كل من الاسم والمهنة، وتوجد قيمة كل مفتاح على يمينه.

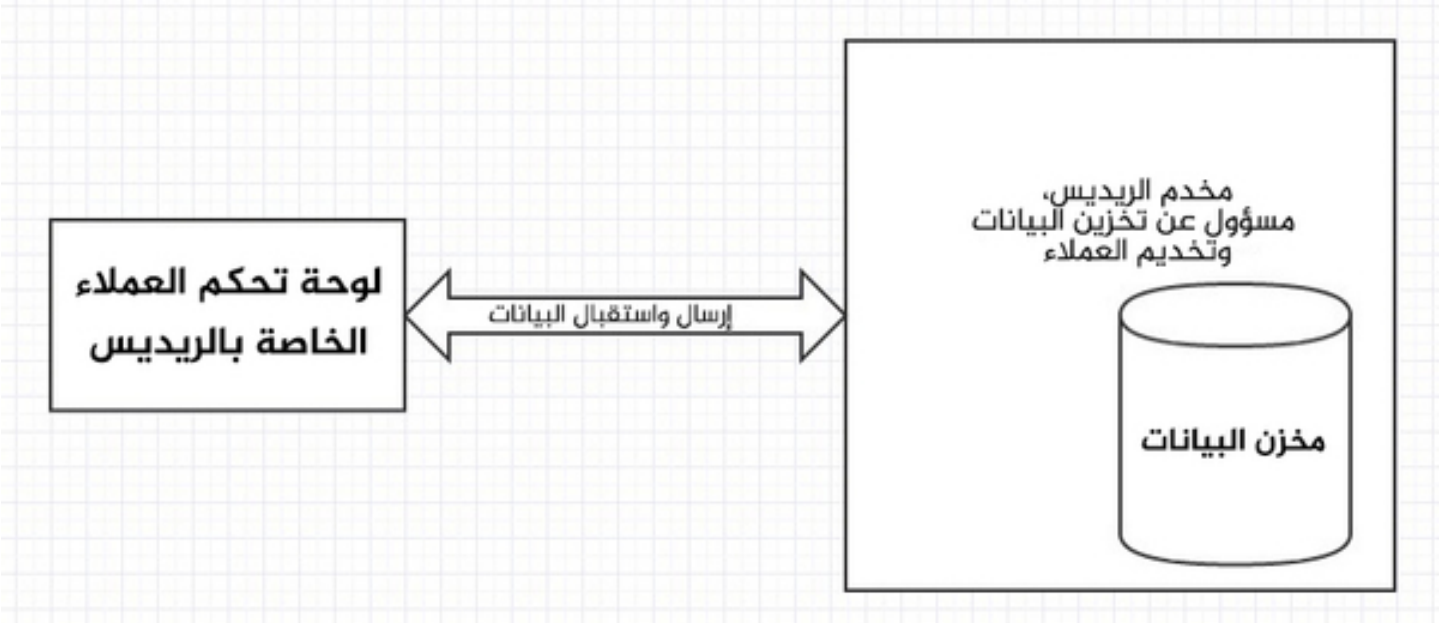
محاسن ومساوئ استخدام الريديس في نظم إدارة قواعد البيانات DBMS

تقوم نظم إدارة قواعد البيانات **Database Management systems** بتخزين كل البيانات ضمن الذاكرة الثانوية مما يبطئ عمليتي القراءة والكتابة، أما الريديس تخزن كل البيانات على الذاكرة الرئيسية مما يسرع عمليتي القراءة والكتابة بشكلٍ كبير.

وبسبب محدودية الذاكرة الرئيسية (أقل حجماً وأكثر تكلفة مقارنةً مع وسائط التخزين الثانوية)، فلا نستطيع تخزين الملفات الكبيرة أو البيانات الثنائية، بل فقط البيانات النصية صغيرة الحجم التي تتطلب الولوج والإدخال والتعديل بسرعة فائقة، وفي حال قمنا بكتابة بيانات أكبر حجماً من المساحة المتوفرة في الذاكرة، فسنتلقى رسالة تفيد بوجود خطأ ما.

مم تتكون بنية الريديس؟

تتكون بنية الريديس من إجرائيتين أساسيتين هما العميل **Redis client** والمخدم **Redis Server**.



يمكن أن يكون كلاهما ضمن حاسوب واحد أو في حواسيب مختلفة، يشكل المخدم الجزء الأكبر من الريدس، حيث يتولى ترتيب البيانات ضمن الذاكرة الرئيسية، ويتعامل مع جميع أنواع الإدارات. أما الزبون، فيمكن أن يكون لوحة التحكم الخاصة بالعميل أو أي لغة برمجة خاصة ببرمجة الواجهات بشرط أن تدعم تقنية الريديس.

إذاً فالريديس تخزن كل شيء ضمن الذاكرة الرئيسية كما رأينا ولكنها ذاكرة موقتة سريعة الزوال، وبالتالي سنفقد المعلومات المخزنة فور إعادة تشغيل المخدم أو جهاز الحاسب، ولهذا نحتاج طريقةً تضمن لنا ثبات المعلومات **datastore persistence**.

ما هي طرق ثبات البيانات persistance؟

هناك ثلاثة طرق لضمان ثبات المعلومات: **RDB** و **AOF** و**SAVE**.

1. آلية عمل تقنية قواعد البيانات العلائقية **Relational Database** أو **RDB**
تقوم هذه التقنية بنسخ جميع المعلومات المخزنة في الذاكرة الرئيسية إلى ذاكرة التخزين الثانوية (الذاكرة الدائمة). ويحدث هذا ضمن فترة زمنية محددة، ولهذا فإن حدوث فقدٍ في البيانات التي تم إدخالها بعد آخر عملية نسخ أمر وارد فعلاً.

2. تقنية إلحاق الملفات **Append only file** أو **AOF**
في هذه التقنية يتم الوصول إلى جميع عمليات الكتابة التي يستقبلها المخدم، ولهذا فكل البيانات المخزنة قد أصبحت دائمة.
المشكلة هنا أن الـ **AOF** تقوم بالتخزين على القرص بعد كل عملية كتابة مما يجعلها مكلفة زمنياً، بالإضافة إلى أن الملفات التي يتم تخزينها باستخدام هذه التقنية أكبر من الملفات المخزنة باستخدام **RDB**.

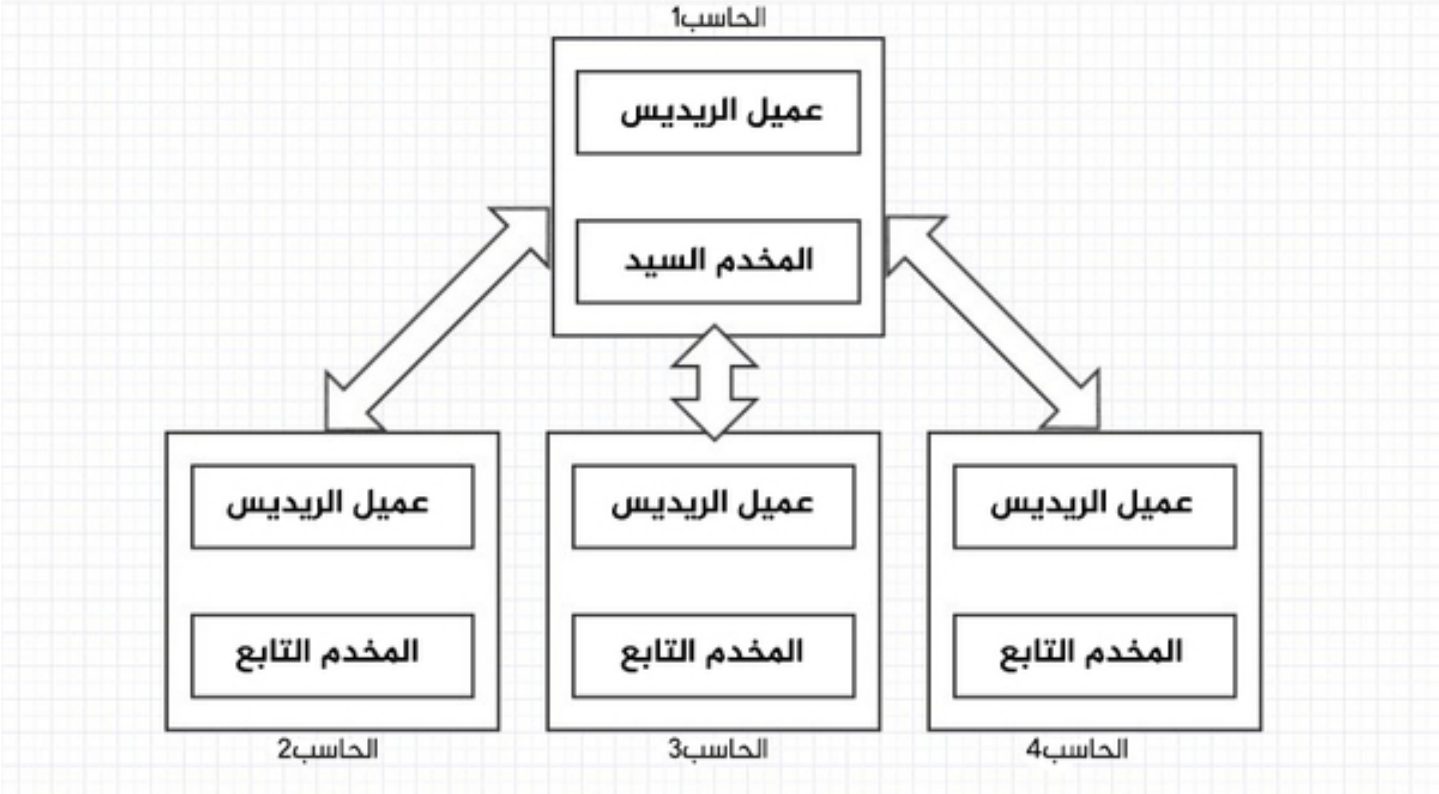
3. أمر الحفظ **SAVE**
يمكننا إجبار المخدم على صنع نسخة من البيانات في أي وقتٍ نريد، وذلك باستخدام أمر الحفظ **SAVE** من لوحة التحكم.
ولضمان نتيجة أفضل يمكننا استخدام كل من **AOF** و **RDB** معاً.

النسخ الاحتياطي واستعادة البيانات Backup And Recovery

لا يوفر الريدس أي تقنية للنسخ الاحتياطي أو لاستعادة البيانات، ولهذا سيتم فقد البيانات في حال تعطل أي قرص صلب أو إن حدثت أي مشكلة أخرى، وللاحتيال على هذه المشكلة، علينا استخدام برمجيات نسخ احتياطي أخرى من أجل استعادة البيانات، أما إن كنت تستخدم الريديس ضمن بيئة عمل تكرارية **replicated environment** فأنت لا تحتاج إلى تلك البرامج الإضافية.

النسخ المطابقة

في هذه التقنية يتم استخدام عدة حواسيب من أجل تصحيح الأخطاء وإمكانية الوصول إلى البيانات. حيث تتشارك عدة حواسيب نفس البيانات بحيث إذا تعطل حاسوب ما تبقى كل البيانات متاحة.
توضح الصورة أدناه أساسيات هذه التقنية:



2

السيد **master** والتوابع **slaves** هي مخدمات من نوع ريدس ذاتية التشكل. تحوي التوابع على نفس البيانات التي يحتويها السيد، ويحوي المخدم السيد على أكبر عدد ممكن من التوابع، وفي حال إدخال تابع جديد إلى بيئة العمل يقوم السيد بمزامنة البيانات معه تلقائياً.

يتم إعادة توجيه جميع الاستعلامات إلى المخدم السيد، حيث يقوم بدوره بتنفيذ هذه العمليات. وعند حدوث عملية كتابة، يقوم السيد بنسخ البيانات الجديدة المدخلة إلى جميع التوابع. وفي حال إجراء عدد كبير من عمليات الترتيب والقراءة يقوم السيد بتوزيع هذه العمليات على التوابع بحيث يتم معالجة عدد كبير منها في وقتٍ واحد.

وفي حال فشل التابع فستستمر البيئة بالعمل، وعند عودة التابع للعمل مجدداً يقوم السيد بإرسال التحديثات التي حصلت إليه. في حال انهيار المخدم السيد وفقد كل بياناته، عندها يتم تحويل جهاز تابع إلى جهاز سيد عوضاً عن احضار سيد جديد. في حال استخدام سيد جديد فسنفقد كل البيانات، حيث أن السيد الجديد لن يحوي أي بيانات وبالتالي سوف يقوم بحذف بيانات التوابع الأخرى. في حال حصول عطل ما في السيد بدون أن تتضرر البيانات على القرص، فيكفي أن نقوم بإعادة تفعيل السيد ليعود كل شيء إلى العمل مجدداً.

تساعدنا عملية النسخ في حال حدوث خلل ما أو في حال تعطل أي تجهيزة أخرى، كما يساعد في تنفيذ عدة عمليات تصنيف وقراءة **read/sort queries** في نفس الوقت.

حفظ البيانات المنسوخة في ريديس

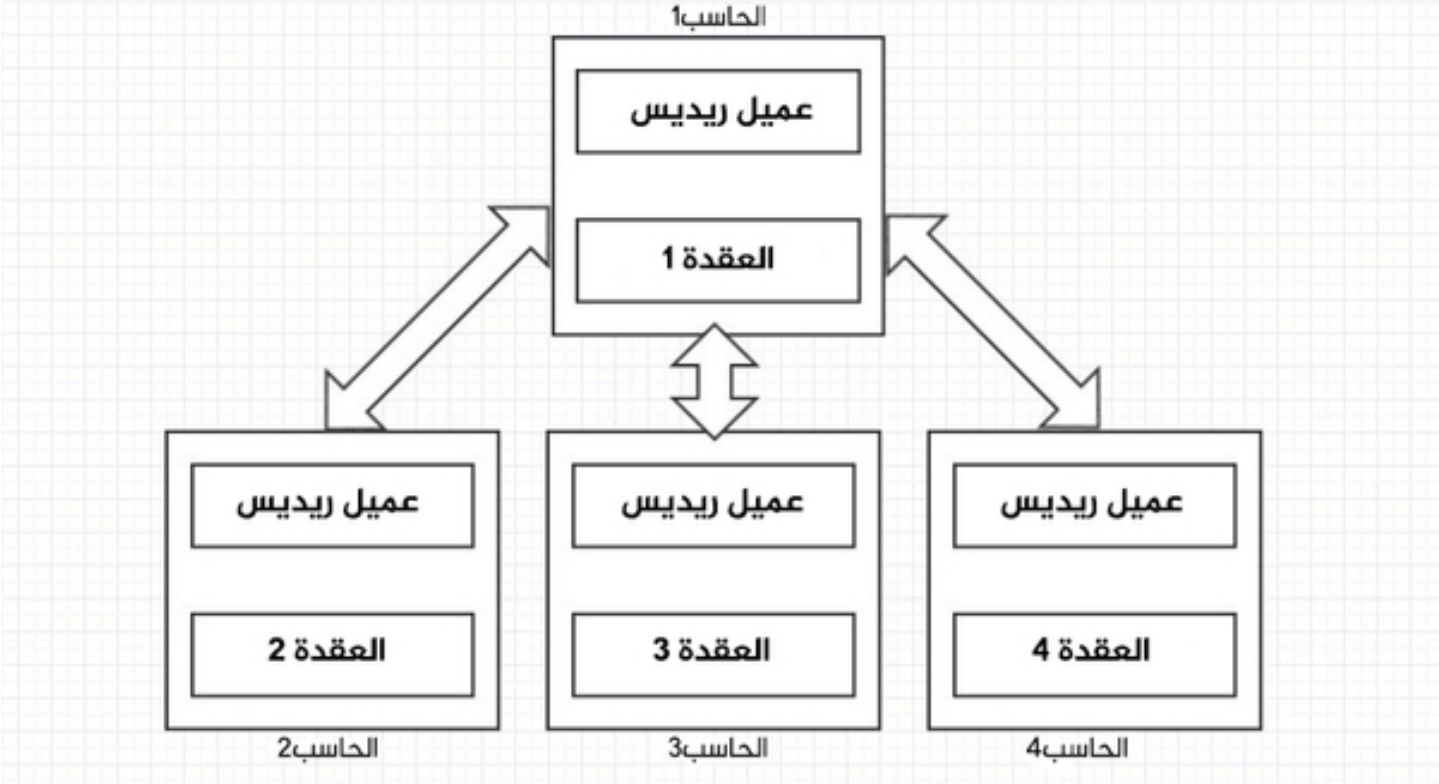
استخدام تقنية الثبات **Persistance** ضمن تقنية النسخ المطابق **Replication** رأينا كيف يمكننا تجنب المشاكل غير المتوقعة عن طريق صنع نسخ مطابقة **Replication** من البيانات. ولكن البيانات ما زالت معرضة لخطر الفقد الكامل في حال حدوث انهيار في منظومة التغذية الكهربائية، حيث أن هذه هي الطريقة الوحيدة التي يتم فيها فقد البيانات بشكل كامل، حيث تكون جميع البيانات مخزنة على الذاكرة الرئيسية ولهذا نحتاج إلى استخدام تقنية الثبات **persistance** في هذه الحالة أيضاً.

لذلك نقوم بتهيئة السيد أو أي تابع آخر لتخزين البيانات في الذاكرة الثانوية باستخدام أي من التقنيتين **AOF**أو **RDB**. والآن وبعد عودة النظام على العمل بشكل كامل، نقوم بضبط المخدم الذي حفظنا عليه البيانات كمخدم سيد، وباستخدام كل من تقنيتي النسخ المطابق والثبات، تكون بياناتنا محمية بشكل كامل من الأعطال المفاجئة.

تقنية العنقدة في الريديس

العنقدة **Clustering** هي تقنية تمكننا من تقسيم البيانات بين عدة حواسيب. ميزتها الأساسية أنها تمكننا من تخزين كمية أكبر من البيانات على شكل عناقيد وذلك لأنها مجموعة من الحواسيب المرتبطة مع بعضها.

فلنفرض أن لدينا مخدم ريديس بذاكرة بحجم 64 غيغابايت، فعندها يمكننا تخزين 64 غيغابايت من البيانات كحد أقصى، أما إن استخدمنا 10 حواسيب تدعم تقنية العنقدة ولكل حاسوب ذاكرة بحجم 64 غيغابايت فعندها سنستطيع تخزين 640 غيغابايت من البيانات.



3

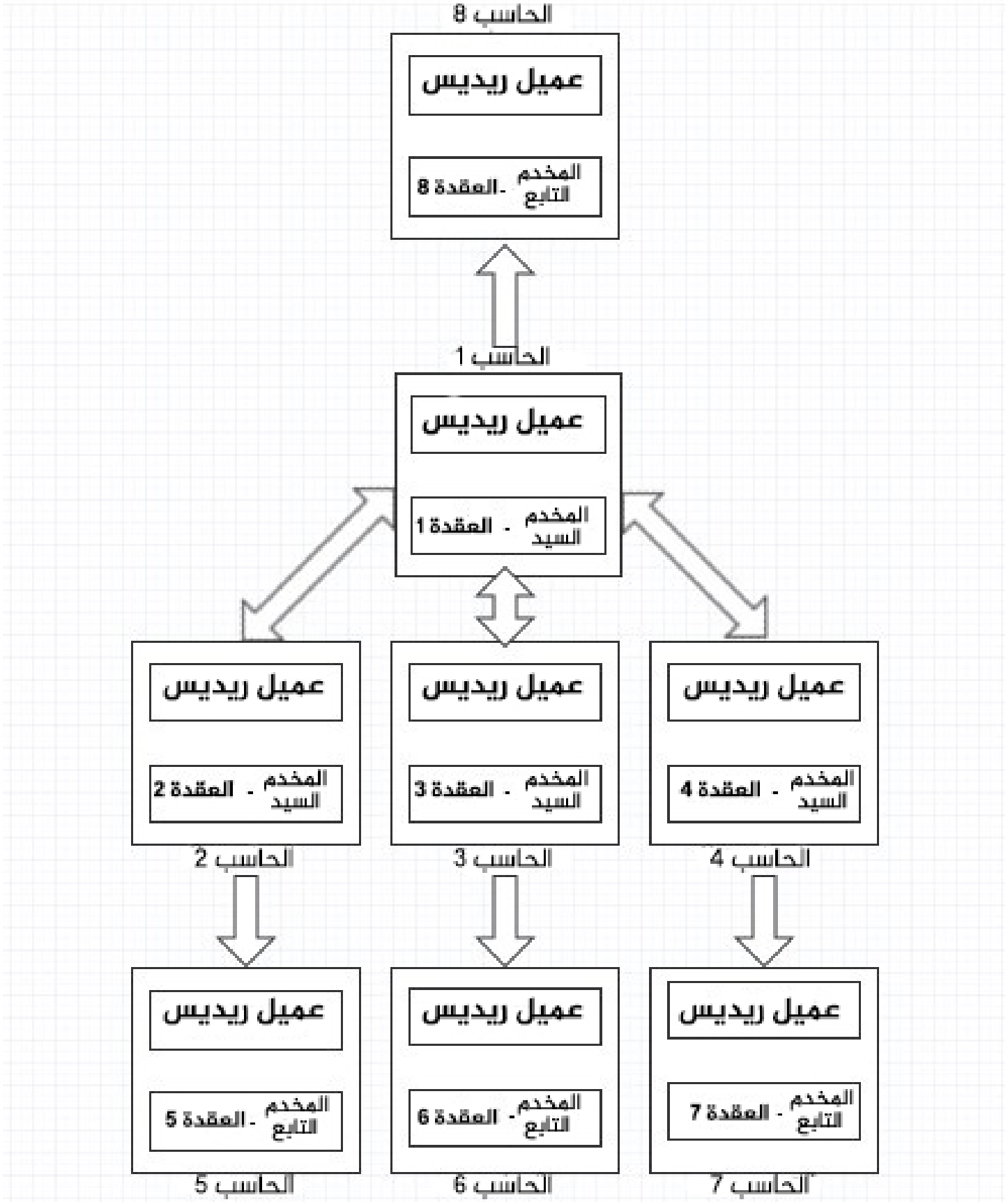
نرى في الصورة أنه تم تقسيم البيانات إلى 4 عقد، كل عقدة تمثل مخدم ريديس تم ضبطه كعقدة عنقود، وفي حال فشل أحد العقد فسوف يتوقف العنقود بشكل كامل.

استخدام تقنيتي الثبات والعنقدة

تخزن البيانات ضمن الذاكرة الرئيسية على شكل عقد **nodes**، وعلينا أن نجعل البيانات الموجودة في العقد ثابتة **persistance**، ويمكننا فعل ذلك باستخدام الطرق التي ذكرناها سابقاً (**AOF** و**RDB**).

العنقدة والنسخ المطابق

فلنفرض أنه حصل عطل في أحد الأقراص، وتلاه انهيار أحد العقد وتسبب ذلك بتوقف كامل العنقود عن العمل بشكل نهائي. وبالتالي، لن نستطيع استعادة العقد طالما أن البيانات قد فُقدت بشكل كامل. ولتفادي ذلك يمكننا القيام بالنسخ الاحتياطي لكل عقدة يدوياً وبشكل دوري، وهذا أمر متعب حقاً، ولذلك يمكننا الاعتماد على عملية النسخ المطابق **replication** لحل هذه المشكلة.


الحاسب 8
عميل ريديس
المخدم التابع - العقدة 8
الحاسب 1
عميل ريديس
المخدم السيد - العقدة 1
عميل ريديس
المخدم السيد - العقدة 2
الحاسب 2
عميل ريديس
المخدم السيد - العقدة 3
الحاسب 3
عميل ريديس
المخدم السيد - العقدة 4
الحاسب 4
عميل ريديس
المخدم التابع - العقدة 5
الحاسب 5
عميل ريديس
المخدم التابع - العقدة 6
الحاسب 6
عميل ريديس
المخدم التابع - العقدة 7
الحاسب 7

قمنا هنا بتحويل كل مخدم عقدة إلى مخدم سيد، وخصصنا تابعاً لكل سيد، وفي حال تعطل أي عقدة (سيد) يقوم العنقود باستخدام التابع للبقاء في حالة العمل.

تجريب الريديس

---

1. لتقم بتجريب الريديس: سوف تساعدك لوحة تحكم العميل الرائعة هذه الخاصة بالريديس في تعلم كيفية استخدام لوحة تحكم العميل عن طريق الإنترنت.
2. بداية سريعة مع الريديس: سوف يساعدك هذا المقال في تنصيب الريديس والبدء بالتعلم.
3. أسئلة وأجوبة: بإمكانك رؤية الأسئلة الأكثر تكراراً حول الريديس في الرابط التالي.

للاطلاع علي الجزء الثاني.

- التاريخ: 2017-04-03
- التصنيف: تكنولوجيا

#لغات البرمجة #علوم الحاسوب #الذاكرة الرئيسية #الريديس #سلسلة الريديس

المصادر

- qnimate

المساهمون

- ترجمة
  - علي مرعي
- مُراجعة
  - ريم المير أبو عجيب
- تحرير
  - روان زيدان
- تصميم
  - محمود سلهب
  - هادي أبو حسون
- نشر
  - مي الشاهد