Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!

وثيقة إحاطة: عملية حل القواعد في Pega Platform

ملخص تنفيذي

حل القواعد (Rule Resolution) في Pega Platform، هو خوارزمية بحث أساسية ومستمرة التشغيل مصممة لتحديد أنسب مثيل قاعدة لتنفيذه في أي موقف معين أثناء تشغيل التطبيق. تعمل هذه الخوارزمية بشكل دائم في الخلفية للحفاظ على سلامة القواعد وعلاقاتها بالفئات.

تتكون العملية من سلسلة خطوات دقيقة تبدأ بالتحقق من الذاكرة المؤقتة (Cache) لتسريع الأداء. إذا لم تكن القاعدة مخزنة مؤقتاً، يقوم النظام بتصفية القواعد من قاعدة البيانات بناءً على "الغرض" (اسم القاعدة ونوعها). بعد ذلك، يتم تجاهل القواعد غير المتاحة، أو التي لا تنتمي إلى قائمة مجموعات القواعد (RuleSet) الخاصة بالمستخدم، أو التي تقع خارج التسلسل الهرمي للفئات.

بعد مرحلة التصفية، يتم ترتيب القواعد المرشحة المتبقية. تحظى القواعد الظرفية (Circumstanced rules) بالأولوية القصوى، يليها الترتيب حسب التسلسل الهرمي للفئات، ثم حسب مجموعة القواعد وإصدارها. في النهاية، يتم التحقق من عدم حظر القاعدة المختارة والتأكد من أن المستخدم الحالي يمتلك الصلاحيات اللازمة لاستخدامها قبل تطبيقها. من الجدير بالذكر أن هذه العملية تنطبق حصرياً على مثيلات القواعد التي تبدأ بـ Rule-، وتستثني فئات البيانات مثل Data- و Work- و System-.

١. مقدمة: جوهر حل القواعد

حل القواعد هو الخوارزمية الأساسية التي يستخدمها نظام Pega للبحث عن المثيل الأكثر ملاءمة من قاعدة ما وتطبيقه في موقف معين. هذه العملية ليست حدثاً منعزلاً، بل هي خوارزمية ثابتة تعمل باستمرار في خلفية النظام للحفاظ على تكامل القواعد وعلاقاتها بالفئات.

تعمل هذه الخوارزمية في سيناريوهات متعددة:

- عند إنشاء قاعدة وحفظها.
- عند تسجيل دخول المستخدم وتحميل مجموعات القواعد الخاصة به.
- عند تشغيل نوع حالة (Case Type) إلخ، ومنطق المعالجة، واجهات المستخدم (مثل والوصول إلى أي خطوة تتطلب استدعاء قواعد).

على الرغم من أن النهج الأولي للخوارزمية قد يبدو غير فعال - حيث تبدأ باستعلام واسع من قاعدة البيانات بدلاً من استخدام فلاتر دقيقة - إلا أن هذه الطريقة أثبتت كفاءتها العالية على مدار أكثر من 35 عاماً، حيث تنجز المهمة في أقل من ثانية في معظم الحالات. يمكن تصور العملية على أنها رحلة عبر التسلسل الهرمي للفئات لـ "جمع" كل القواعد المحتملة في سلة واحدة، ثم فرزها بدقة لاختيار الأنسب.

٢. نطاق وتطبيق حل القواعد

من الأهمية بمكان فهم أن عملية حل القواعد لا تنطبق على جميع المثيلات في النظام. يقتصر نطاقها بشكل صارم على المثيلات التي تنتمي إلى فئات يبدأ اسمها بـ Rule-.

تشمل الأمثلة على القواعد الخاضعة للحل:

- Rule-Obj-Property (الخصائص)
- Rule-HTML-Harness (واجهات Harness)
- Rule-Obj-DataTransform (تحويلات البيانات)
- جميع القواعد الموجودة في قسم واجهة المستخدم (UI).

المثيلات المستثناة من عملية الحل: لا تخضع المثيلات التي تنتمي إلى الفئات التي تبدأ بـ System- أو Work- أو Data- لعملية حل القواعد. وتُعتبر هذه المثيلات "بيانات تكوينية" وليست قواعد قابلة للتنفيذ بنفس الطريقة.

- **Data-:** يشمل ذلك فئات مهمة مثل Data-Admin-Operator-ID (معرّف المشغّل)، ومجموعات الوصول (Access Groups)، وقواعد التطبيق (Application Rules). لا يتم حل هذه العناصر بل يتم التعامل معها كبيانات.
- **Work-:** فئات أنواع الحالات لا تخضع لعملية الحل.
- **System-:** فئات النظام أيضاً مستثناة.

Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!

عملية حل القواعد المكونة من 10 خطوات٣ .

تنقسم العملية إلى مرحلتين رئيسيتين: المرحلة الأولى (الخطوات 1-5) تهدف إلى تصفية وتحديد القواعد المرشحة الصالحة، بينما تركز المرحلة الثانية (الخطوات 6-10) على ترتيب هذه القواعد لاختيار أفضلها.

المرحلة الأولى: التصفية وتحديد القواعد المرشحة

1. **التحقق من الذاكرة المؤقتة (Cache Check):**
   a. أولاً، يتحقق النظام مما إذا كانت القاعدة المطلوبة موجودة بالفعل في الذاكرة المؤقتة. يتم تخزين القواعد التي يتم استدعاؤها بشكل متكرر في الذاكرة لتجنب استعلامات قاعدة البيانات المكلفة.
   b. إذا تم العثور على القاعدة في الذاكرة المؤقتة، يتجاوز النظام الخطوات من 2 إلى 7 وينتقل مباشرة إلى الخطوة 8.
2. **الاختيار حسب الغرض (Selection by Purpose):**
   a. إذا لم تكن القاعدة في الذاكرة المؤقتة، يقوم النظام بتنفيذ استعلام SELECT عام من قاعدة البيانات.
   b. يعتمد الاستعلام على "الغرض"، والذي يتكون من جزأين: **اسم القاعدة** (المعرّف ID) و**نوع القاعدة** (Rule Type).
   c. هذا الاستعلام الأولي واسع النطاق ويعيد مجموعة كبيرة من القواعد المحتملة، حتى لو كانت بعضها لا يتطابق مع التسلسل الهرمي الحالي.
3. **تجاهل القواعد غير المتاحة (Discard Unavailable Rules):**
   a. يقوم النظام بمراجعة القائمة الأولية من القواعد ويتجاهل أي قاعدة تم تعيين حالتها إلى Not Available.
4. **تجاهل القواعد خارج قائمة مجموعة القواعد (Discard Rules Outside the RuleSet List):**
   a. بعد ذلك، يقارن النظام القواعد المتبقية بقائمة مجموعات القواعد الديناميكية (RuleSet List) الخاصة بالمستخدم.
   b. يتم تجاهل أي قاعدة لا تنتمي إلى مجموعة قواعد أو إصدار مدرج في قائمة المستخدم.
5. **تجاهل القواعد خارج التسلسل الهرمي للفئات (Discard Rules Outside the Class Hierarchy):**
   a. أخيراً، يتم تصفية القائمة مرة أخرى لإزالة أي قواعد لا تقع ضمن مسار الوراثة للفئة الحالية.

المرحلة الثانية: الترتيب والاختيار النهائي

1. **ترتيب القواعد المرشحة المتبقية (Rank Remaining Candidate Rules):**
   a. في هذه المرحلة، تكون لدى النظام قائمة بالقواعد الصالحة فقط. يقوم بترتيبها لتحديد الأفضل وفقاً للتسلسل الهرمي التالي:
      i. **أولاً: القواعد الظرفية (Circumstanced Rules):** تحظى هذه القواعد بالأولوية الأعلى. هناك تسلسل داخلي بين أنواع الظروف المختلفة (مثل الظروف متعددة المتغيرات، وظروف الخصائص، وظروف التاريخ).
      ii. **ثانياً: التسلسل الهرمي للفئات (Class Hierarchy):** القواعد الموجودة في فئات أكثر تحديداً (أقرب إلى فئة العمل الحالية) تحتل مرتبة أعلى من تلك الموجودة في الفئات العامة (مثل فئات Pega الأساسية).
      iii. **ثالثاً: مجموعة القواعد والإصدار (RuleSet and Version):** يتم استخدامها كمعيار ترتيب نهائي.
2. **إضافة القائمة المرتبة إلى الذاكرة المؤقتة (Add Ranked List to Cache):**
   a. يتم إضافة القائمة المرتبة من القواعد المرشحة إلى الذاكرة المؤقتة لتسريع الطلبات المستقبلية لنفس القاعدة.
3. **تحديد المثيل الأنسب والتحقق من التكرارات (Find the Most Applicable Instance & Check for Duplicates):**
   a. يختار النظام القاعدة الموجودة في أعلى القائمة المرتبة.
   b. يتحقق أيضاً من وجود أي "أشقاء" (Duplicates)، وهي قواعد بنفس الاسم والفئة ولكن في مجموعات قواعد مختلفة.

4. **التحقق من أن القاعدة ليست محظورة (Check that the Rule is Not Blocked):**
    a. يتم إجراء فحص حالة ثانٍ للتأكد من أن القاعدة المختارة ليست Blocked.
5. **التحقق من صلاحيات المستخدم (Verify Requester Privileges):**
    a. تمنحه الإذن لاستخدام هذه (Requestor) كخطوة أخيرة حاسمة، يتحقق النظام من أن مجموعة الوصول الخاصة بالمستخدم القاعدة.

بعد اجتياز القاعدة لجميع هذه الفحوصات، يتم تطبيقها وتنفيذها في وقت التشغيل. في حالات نادرة جداً، قد يفشل النظام في تحديد قاعدة واحدة فريدة، مما يؤدي إلى حدوث خطأ.

٤. رؤى رئيسية واقتباسات هامة

رؤية رئيسية	الوصف والاقتباس
الكفاءة غير المتوقعة	على الرغم من أن الخوارزمية تبدأ باستعلام قاعدة بيانات عام وغير محدد، إلا أنها فعالة للغاية. "إنه ينفذ أمر WHERE عام، ويعيد كل هذه القواعد، وبعضها لا يتطابق حتى مع التسلسل الهرمي، وهو ما أعتقد أنه تفعل هذا منذ 35 عامًا. Pegasystems جنوني. لكن"
عملية من مرحلتين	يمكن تقسيم العملية بوضوح إلى مرحلتين متميزتين. "الخطوات الخمس الأولى تحاول في الواقع اكتشاف القواعد الصالحة. والخطوات الخمس الأخيرة تقوم بترتيبها للعثور على الأفضل."
أولوية الذاكرة المؤقتة	يعد التحقق من الذاكرة المؤقتة هو الخطوة الأولى والأكثر أهمية لتحسين الأداء، حيث يسمح بتجاوز معظم الخطوات الأخرى. "إذا كانت (القاعدة) في الذاكرة المؤقتة، فإنها تقفز مباشرة إلى الخطوة رقم ثمانية."
التحقق المتأخر من الصلاحيات	من المثير للاهتمام أن صلاحيات المستخدم يتم التحقق منها في نهاية العملية وليس في بدايتها، وهو قرار تصميمي أثبت فعاليته. "كانت فكرتي الأولى عندما رأيت هذا هي: لماذا لا يتم التحقق من أن الطالب لديه الصلاحية في البداية؟ ... ولكن يبدو أن هذا النهج يعمل وهذا ما يستخدمونه."