Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

PHP versus Perl, primera cata (27157 lectures)

Per **Ricardo Galli Granada**, ***gallir*** (http://mnm.uib.es/gallir/)

Creado el 07/03/2001 00:00 modificado el 07/03/2001 00:00

Como dice[1] John Lim, discutir de lenguajes de programación es como discutir de vinos: a unos les gusta seco, a otros les gusta afrutados. Otros no entienden la discusión, a otros no les gusta nada el vino. En este artículo discuto 10 diferencias importantes entre la programación en Perl y PHP.

Aunque las sintaxis de ambos lenguajes tienen cosas parecidas heredadas del C y el primer prototipo de PHP estuvo hecho en Perl, actualmente tienen casi nada que ver, el Perl nació como un lenguaje de *scripting* para tareas de UNIX, el PHP fue diseñado específicamente para web.

Si tuviera que ponerlo en pocas líneas "reglas" diría:

1. La integración de PHP con páginas HTML es muy fácil y directo. Puedes "embeber" (empotrar...) PHP en una página HTML ya terminada sin ningún problema. Esto significa que hay una clara separación de código y datos.
2. El PHP debería ser comparado con el mod_perl (o al menos con CGI::SpeedyCGI), ya que son interpretados por un módulo del apache y no por un intérprete externo, que genera mas sobrecarga (*overhead*) al necesitar un fork/exec para ejecutar el intérprete. Programar con el mod_perl es algo más complicado que programar con Perl como CGI externo directamente, sobre todo por la persistencia y visibilidad (*scope*) de las variables que uses. Por ejemplo, las variables globales pueden ser accedidos por todos los programas que usen el mod_perl, lo que obliga a un control muy estricto de las variables globales. Mirar en http://perl.apache.org/guide/porting.html[2] para mas detalles. Hace un par de años, con programas hechos para Perl 5.004 tuve bastantes problemas de que se ejecuten correctamente con el mod_perl, al final lo abandoné porque los programas eran complejos y me iba a llevar mucho trabajo pasarlos el mod_perl.
3. El PHP, desde el punto de vista de *script* para webs, tiene prácticamente las mismas posibilidades y potencia que el Perl. Inclusive (y muy importante) es _mucho_ mas sencillo en la definición de interfaces (p.e. Perl:DBI vs. bases de datos en PHP) a costa de perder algo de flexibilidad.
4. Desde el punto de vista del trabajo "cooperativo" con los diseñadores, es mucho mas cómodo el PHP, ya que los diseñadores/maquetadores pueden hacer el html normal y luego los programadores meten el código PHP necesario. Yo considero esto como una de las mayores ventajas del PHP, ya que para hacer sitios web necesitas siempre de diseñadores.
5. El PHP es relativamente nuevo y cada día salen módulos y extensiones (por ejemplo, PHP-GTK), está creciendo bastante y hay mogollón de ejemplos de código, quizás no tanto como el perl todavía, pero está creciendo mucho.
6. EL PHP4 (a diferencia del PHP3) tiene una muy buena gestión de *thrashing* (liberación de variables/datos que ya no son referenciados). Este era un problema en PHP3 con los accesos a las base de datos que retornan muchos datos. El PHP4 verifica que los datos ya no son referenciados y libera memoria inmediatamente.
7. El *Zend optimizer* incluido en el PHP4 es muy rápido, e inclusive hay módulos adicionales para optimizar la compilación (gratuito, se puede bajar desde www.zend.com[3]) que es muy bueno cuando los scripts son largos y complejos. Ahora también puedes instalar módulos de cache para evitar tener que interpretar cada vez el código fuente. Hay una versión comercial en Zend (cuesta unas 300.000 Ptas, $1.500, pero es muy buena) y hay otra versión libre (APC) en http://apc.communityconnect.com/[4]
8. La integración del PHP es muy buena con el Apache, por ejemplo, las cabeceras del HTT, campos de formularios o argumentos en los URLs, se pueden acceder directamente como variables globales, sin necesidad de interpretarlos. También permite trabajar con funciones POSIX tal como haces en C sobre UNIX, por ejemplo: semáforos, memoria compartida, acceso a ficheros, funciones de tiempo, etc.
9. Los módulos de acceso a base de datos para Postgres, MySQL, Oracle, ODBC genérico, DB2 e Interbase

funcionan muy bien. En general no hay problemas con las interfaces a bases de datos.
10. Por último, _mi_ esfuerzo de aprendizaje de PHP fue mucho menor que al de Perl.

En general yo usaría PHP para embeber código en HTML y si tuviera que hacer cosas más complejas que no pudiera con el PHP (no se me ocurre nada ahora), usaría el Perl. No me haría tantos problemas: elijo lo más fácil para lo genérico y si tengo que hacer cosas que me resulten más sencillas de programarlo con Perl, lo haría sin pensarlo dos veces: *make it easy, stupid* ;-).

Otros enlaces:

- Comparación de PHP y Perl[1] para páginas dinámicas.
- Comparación de PHP y ASP[5]
- Comparación de PHP y Cold Fusion[6]
- What's so cool about Perl
- Comparación de 5 lenguajes de *scripting*: Perl, PHP, Python, Tcl, Java servelets (NOTA: los ejemplos de PHP son del PHP3, en los ejemplos no muestran ni mencionan las expresiones regulares compatibles con Perl en el PHP4)

--ricardo

Lista de enlaces de este artículo:

1. http://php.weblogs.com/php_versus_perl
2. http://perl.apache.org/guide/porting.html
3. http://%20www.zend.com
4. http://apc.communityconnect.com/
5. http://php.weblogs.com/php_vs_asp
6. http://php.weblogs.com/php_vs_cold_fusion

E-mail del autor: gallir _ARROBA_ uib.es