TYPO3.Flow - Feature # 56036

Status:	New	**Priority:**	Should have
Author:	Benjamin Heek	**Category:**	
Created:	2014-02-17	**Assigned To:**	
Updated:	2014-03-03	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Optimize autoloading		

Description

The TYPO3\Flow\Core\Autoloader is checking where to find a class every request.

It's much faster to store the founded path for a class in something like a `class path cache` variable.

It made my time per request go from 140ms to 70ms.

For Example:

```
    /**
       * @var \TYPO3\Flow\Cache\Frontend\PhpFrontend
       */
       protected $classesCache;

       public function __construct() {
            $this->classesPaths = apc_exists('classespaths') ? apc_fetch('classespaths') : array();
       }

       public function __destruct() {
            apc_store('classespaths', $this->classesPaths);
       }

       /**
       * Loads php files containing classes or interfaces found in the classes directory of
       * a package and specifically registered classes.
       *
       * @param string $className Name of the class/interface to load
       * @return boolean
       */
       public function loadClass($className) {
           if ($className[0] === '\\') {
               $className = substr($className, 1);
           }
           if (isset($this->classesPaths[$className])) {
               require($this->classesPaths[$className]);
               return TRUE;
           }

           // Loads any known proxied class:
           if ($this->classesCache !== NULL && $this->classesCache->requireOnce(str_replace('\\', '_', $className)) !==
  FALSE) {
```

```
                $this->classesPaths[$className] = $this->classesCache->getBackend()->getCacheDirectory() . str_replace('\\',
'_', $className) . '.php';
                return TRUE;
            }
```

History

#1 - 2014-02-26 16:25 - Rafael Kähm

Hello Benjamin,

cool idea, but your example will crash multiple PHP-FPM pools, if they are started with single FPM.

If somebody needs that and uses multiple Flow setups with FPM pools, then each fpm-pool must be started separately.

Also:

```
php-fpm --daemonize -c /etc/php/fpm/php.ini --fpm-config /etc/php/fpm/php-firstSetup.conf
php-fpm --daemonize -c /etc/php/fpm/php.ini --fpm-config /etc/php/fpm/php-secondSetup.conf
...
```

#2 - 2014-03-03 11:15 - Kobbe van Daatselaar

Hi Rafael,

what Benjamin tries to say is that caching the successfully loaded classes and using a require instead of a include causes a large performance improvement.

The way it should be cached can be discussed...