IO optimalizations, deployment, security

Dominik Pinter, dominikp@kentico.com

Kentico

Agenda

CMS.IO:

- Optimalization
- Temporary files
- Object locking
- File access
- Debug

Deployment:

- Compilation, debugging
- On-premise vs. Off-premise

Security:

- Web security
- C# code security

CMS.IO optimalization - caching

```
public override bool Exists(string path)
{
        return Provider.GetFileFromPath(path) != null;
}
```

```
public FileClass GetFile(string id)
{
        try
        {
           return Service.Files.Get(id).Fetch();
        }
        catch (Exception)
        {
           return null;
        }
}
```

CMS.IO - caching

- What?
  - Metadata
  - File contents
- How?
  - RequestStockHelper
  - Local file system
- Techniques
  - E-tag
  - Download only what is really needed

CMS.IO - caching

```
public bool Exists()
{
      if (RequestStockHelper.Contains(Key + "|Exists"))
      {
         return ValidationHelper.GetBoolean(RequestStockHelper.GetItem(Key + "|Exists"), false);
      }

      GetObjectMetadataRequest request = new GetObjectMetadataRequest();
      request.BucketName = BucketName;
      request.Key = Key;

      try
      {
         GetObjectMetadataResponse response = S3Client.GetObjectMetadata(request);
         response.Dispose();

         RequestStockHelper.Add(Key + "|Exists", true);
         RequestStockHelper.Add(Key + "|Length", response.ContentLength);
         RequestStockHelper.Add(Key + "|FetchMetadata", true);

         return true;
       }
…
```

CMS.IO - caching

```
public CMS.IO.Stream GetObjectContent(IS3ObjectInfo obj)
{
        // First try to get data from cache
        string cachePath = PathHelper.CachePath + GetPathFromObjectKey(obj.Key, false);
        string eTagFilename = cachePath + ".etag";

        if (CMS.IO.File.Exists(cachePath))
        {
                string eTag = System.IO.File.ReadAllText(eTagFilename).Trim();
                if (eTag == obj.ETag)
                {
                        // Return cached object
                }
         }

        // Download object
        // Save it to cache
}
```

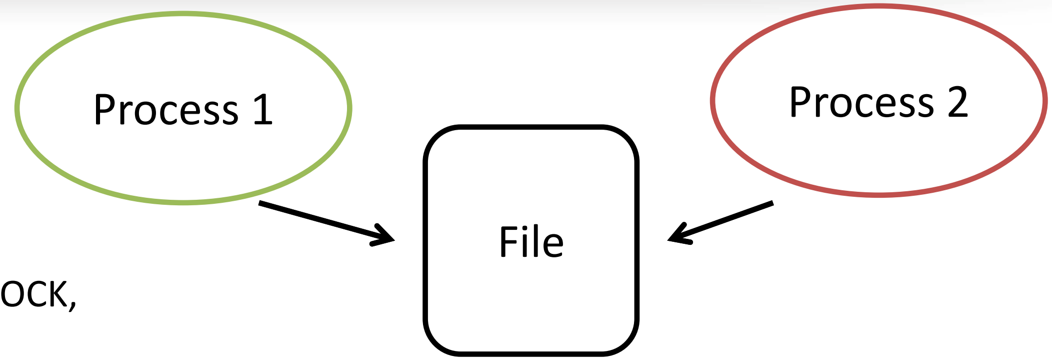
CMS.IO optimalization – temporary files

```
private System.IO.FileStream _innerStream;

public override void Write(byte[] buffer, int offset, int count)
{
        _innerStream.Write(buffer, offset, count);
}

public override void Dispose()
{
        _innerStream.Close();

        if (fileAccess != FileAccess.Read && fileMode != FileMode.Open)
        {
                _fileService.FileWriteContent(_validPath.ObjectName, _validPath.Parent,
                System.IO.File.ReadAllText(_validPath.ObjectName));
        }
}
```

CMS.IO - locking

```
public void Lock()
{
    if (Provider.ObjectExists(this))
    {
        SetMetadata(S3ObjectInfoProvider.LOCK,
            "True", true, false);
     }
}
```

```
public void UnLock()
{
     if (Provider.ObjectExists(this))
     {
         SetMetadata(S3ObjectInfoProvider.LOCK,
             "False", true, false);
     }
}
```



```
public void PutTextToObject(IS3ObjectInfo obj,
string content)
{
            if (!obj.IsLocked)
            {
                        obj.Lock();

                        // Do action

                        obj.UnLock();

            RemoveRequestCache(obj.Key);
            }
}
```

CMS.IO – public access

Komentář z jednoho z projektů v metodě GetFileUrl:

```
/* opsano z Amazon solution - nemate nekdo implementaci techto metod
abych pochopil co delaj? podle jejich jmen to moc poznat nejde (i kdyz by
melo)
            string url = AmazonHelper.GetDownloadPath(relative);
            string hash =
ValidationHelper.GetHashString(URLHelper.GetQuery(url), false);
            url = URLHelper.AddParameterToUrl(url, "hash", hash);

            return URLHelper.ResolveUrl(url);
*/
```

CMS.IO – public access

1. Implement file access: request.CannedACL = S3CannedACL.PublicRead;

2. Implement method for resolving URLs:

```
public override string GetFileUrl(string path, string siteName)
 {
    string objKey = PathHelper.GetObjectKeyFromPath(path);
    if (ExistsInS3Storage(path))
    {
       if (AmazonHelper.PublicAccess)
       {
             return String.Format("{0}/{1}", AmazonHelper.EndPoint, objKey).ToLowerCSafe();
       }

       string relative = "~\\" + objKey.Replace('/', '\\');

       string url = AmazonHelper.GetDownloadPath(relative);
       string hash = ValidationHelper.GetHashString(URLHelper.GetQuery(url), false);
       url = URLHelper.AddParameterToUrl(url, "hash", hash);
       return URLHelper.ResolveUrl(url);
    }

    return null;
 }
```

CMS.IO - debug

- Helps you with optimalizations
  - 1 storage operation per 1 request
  - Content is downloaded only once

- CMSSiteManager->Administration->System->Debug->IO
- Live site

- using CMS.IO
  - Directory.LogDirectoryOperation()
  - File.LogFileOperation()

- IOOperationType.General
  - Use prefixes to identify storage operations

Deployment – On-premise vs. Off-premise

- On-premise = standard server
  - webhosting
- Off-premise = cloud
  - IaaS vs. PaaS vs. SaaS
  - Multi servers environment
  - Shared storage
  - Content delivery network

Security: web security

Golden rule: sanitize users input!

GetXXFile.aspx:

```
protected void Page_Load(object sender, EventArgs e)
{
    string hash = QueryHelper.GetString("hash", string.Empty);
    string path = QueryHelper.GetString("path", string.Empty);
    if (ValidationHelper.ValidateHash("?path=" + path, hash, false))
    {
    }
}
```

Q&A

?

Thank you

Kentico
CMS for ASP.NET

http://www.kentico.com
http://devnet.kentico.com
dominikp@kentico.com