Best practices with development of enterprise-scale SharePoint solutions

PAOLO PIALORSI, PIASYS.COM

paolo@pialorsi.com - @PaoloPia

SPC ADRIATICS 2014

SPONSORS

PLATINUM SPONSOR

deroso solutions GmbH
an FGND group company

GOLD SPONSORS

Documentation Toolkit for SharePoint

tecmasters.net
solutions for the .net Enterprise

K2

eVision

lobodms

SPONSORS

Microsoft

AvePoint®

NINTEX

A&M

About Me

- Project Manager, Consultant, Trainer
- More than 40 Microsoft certification exams passed, including MC(S)M
- Focused on SharePoint since 2002
- Author of 10 books about XML, SOAP, .NET, LINQ, and SharePoint
- Speaker at main IT conferences worldwide

- http://www.piasys.com/

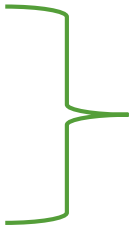
Agenda

- What's an enterprise-scale solution?
- Best Practices for …
  - Server-side code
  - Workflows
  - Public Facing Web Sites

What's an enterprise-scale solution?

What makes a solution enterprise-scale?

- Some technical numbers that matter
  - Number of concurrent users
  - Number of transactions x day
- Some technical facts that matter, as well
  - Business continuity requirements
  - Scalability requirements
  - Etc.
- Some other facts that still matter
  - Overall size of the customer
    - Network and infrastructure topology
  - Overall complexity of the solution
    - Number of use cases to support
  - Team Project size
  - Etc.



Determine Overall Throughput

How to calculate Throughput

- Throughput = number of operations per second (RPS)
- Input Variables
  - Total Number of Users
  - Concurrency Rate (%)
  - Request Rate
- Easy Output
  - Total Number of Users * Concurrency Rate / Request Rate
- More Realistic and Complex Output
  - Based on usage clusters
  - And on concurrency peak

Cluster	Req/Hour	Distribution
Light	20	10%
Typical	36	70%
Heavy	60	15%
Extreme	120	5%

Let's make an example!

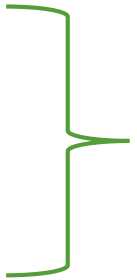
- Input Variables
  - Total Number of Users: 10.000
  - Concurrency Rate (%): 10%
  - Request Rate: 36/hour = 1 every 100 seconds
- Easy Output
  - 10.000 * 10% / 100 = 10 RPS
- Complex Output
  - Usage Peak: 33%
  - 42.200 / 3600 = 11,72 RPS
  - Realistic Throughput = 3,87 RPS

Cluster	Req/Hour	Distribution	# Users	# Req/Hour
Light	20	10%	1.000	2.000
Typical	36	70%	7.500	25.200
Heavy	60	15%	1.500	9.000
Extreme	120	5%	500	6.000
				42.200

THE BIGGEST IS THE THROUGHPUT, THE MORE YOU HAVE TO TAKE CARE OF QUALITY OF CODE … BUT YOU SHOULD ALWAYS TAKE CARE OF IT!

Our scope in this session

- Here we focus on top best practices for enterprise-scale solutions
- What kind of best practices?
  - Better Performances
  - Better Scalability
  - High Availability and Business Continuity
  - Stability of Code
  - Security of Code
- What kind of solutions are we thinking about?
  - Custom Development Solutions
  - Custom Workflow Solutions
  - Public Facing Web Sites



"Server-side" solutions

Let's see the top 15 rules!

Rule #1: Properly Dispose unmanaged data

- Remember to properly release unmanaged resources that you have created
  - Leverage the *using* keyword in that case
- Avoid releasing unmanaged resources that you did not created

- If you don't adhere to this rule
  - Your code will be unstable
  - Sooner or later you will crash the hosting app pool
  - Because of memory leaks … performances could degrade, as well

- Use SPDisposeCheck and/or SPCAF
- Rule applies to: *SPSite*, *SPWeb*, *SPLimitedWebPartManager*, etc.

PROPERLY DISPONING UNMANAGED RESOURCES

Rule #2: Create *SPWeb/SPSite* only if needed

- Don't create *SPWeb/SPSite* objects too much freely
- Create them only in case of real need
- Otherwise leverage *SPContext/SPControl*
  - This will keep memory lighter and will be less error prone (mind Dispose)

- If you don't adhere to this rule
  - Memory allocation will increase uselessly
  - Performances will degrade
  - Code will not scale properly

Rule #3: Request only the data that you need

- Don’t ever iterate over full *SPListItemCollection* of a list/library
- Always query items using a specific view or a CAML query
- Remember to page data, instead of quering the full resultset

- If you don’t adhere to this rule
  - Performances will be bad and sad
    - Mainly in production, with huge numbers of users/items/documents/requests
  - Code will not scale properly
    - When the size of data grows … the performances decrease

QUERYING DATA WITH CAML THE RIGHT WAY

Rule #4: Don’t abuse of LINQ to SharePoint

- Instead of LINQ to SharePoint queries
  - Which let you pay for the LINQ QueryProvider work, under the cover
- Prefer well-defined CAML queries with paging
- Avoid two-stage queries in LINQ to SharePoint
  - Involve superset queries through CAML + LINQ to Objects in memory
- For read-only LINQ to SharePoint queries
  - Use view projections and turn off ObjectTracking
- Eventually simply use LINQ to SharePoint to discover CAML queries

- If you don’t adhere to this rule
  - Your code will be slower than expected
  - Code will not scale properly
  - Solution ALM will be painful from a maintenance perspective

Rule #5: Handle exceptions properly

- Catch and handle only those exceptions that you anticipate and can manage
- Avoid simply catching all exceptions using a *catch all* block or an empty *catch* block
- Always log exceptions and issues in the ULS log, for centralized management

- If you don't adhere to this rule
  - Your code will be unstable
  - The behavior of your solutions will be unpredictable

PROPERLY HANDLING EXCEPTIONS

Rule #6: SharePoint isn't a DBMS!

- Avoid using SharePoint as a DBMS replacement!
- SharePoint isn't a transactional resource manager
- SharePoint cannot participate in distributed transactions
  - Or at least cannot do that with consciousness ☺ …
- Consider using a polyglot system (SharePoint + DBMS + noSQL)
  - Handle with care data persistence and data integrity
  - Make your choice based on what you need to persist

- If you don't adhere to this rule
  - Sooner or later your solution will fail to keep data integrity
  - Your code will be potentially unstable
  - The behavior of your solutions will be unpredictable

MANAGING A POLYGLOT SYSTEMS, WHICH INCLUDES SHAREPOINT

Rule #7: Mind throttling rules

- Keep in mind the list view threshold settings
  - 5.000 items for regular users
  - 10.000 items for site administrators
  - Daily time window for overriding
  - Unlimited for Local Administrators in local/remote server session
- Overridable through Object Model
  - But it is not a good idea to exceed limits!
  - Restrict the scope of any overriding to the minimum
- Consider List View Lookup Threshold, as well
- Partition data results through custom views
- Use indexed columns to avoid full table scan
- Impacts on large workflow solutions (history list and task list)

MANAGING THROTTLING OVERRIDES

Rule #8: Avoid too many item-level permissions

- Assign permissions at the highest possible level
- Break permission inheritance as infrequently as possible
- Don’t create tons of items with item-level permissions
- Prefer folders to group items and their security rules via groups membership
  - In SharePoint 2013 continuous crawl mitigates security trimming issues
- For lists of items (!documents) use ReadSecurity and WriteSecurity permission levels

Rule #9: Avoid "elevated" code via *SPSecurity*

- *SPSecurity.RunWithElevatedPrivileges* is generally speaking a "bad idea"
  - Consider changing the implementation
  - Or at least impersonate "minimal" principals using
    - *SPSite(*, SPUserToken)* constructor

- If you don't adhere to this rule
  - You could have potential security issues
  - Your code could be unsafe for the target farm

Rule #10: Properly use *AllowUnsafeUpdates*

- Avoid using *AllowUnsafeUpdates* as much as you can
- And use it properly in case you really need it
  - Remember to use a *try* … *finally* block of code to reset it
  - Especially when you share *SPWeb* with others
- Remember to put *FormDigest* in your custom pages
  - And manage update/delete/insert actions via POST
  - Avoid update/delete/insert actions via GET, as much as you can

WRITING SAFE AND SECURE CODE IN SHAREPOINT

Rule #11: Mind workflow numbers

- Keep in mind List View Threshold and Trottling for history and tasks lists
- Be careful with item level permissions, as already stated
- Dependant tasks and task-level permissions are not supported out of the box by the Workflow Manager tools
  - You have to manually create custom actions for that

Rule #12: Caching, caching, caching!

- Avoid useless network roundtrips
- Keep in cache frequently accessed and rarely changing data
  - SharePoint Objects
  - Lookup data
  - Images, blob files, etc.
- Out of the box there are
  - Object Cache
  - Output Cache
  - BLOB Cache
- Use them to improve performances and response time
  - In particular in Public Facing Web Sites, but not only

Rule #13: Try to always use CQWP/CSWP

- In Public Facing Web Sites always try to use
  - Content by Query Web Part
  - Content by Search Web Part
- These web parts do a good caching of results
  - Thus, improving the overall perceived time
- You only have to focus on XSLT or HTML Templates
- Fallback to *DataFormWebPart* or *XsltWebPart*
- Lastly, if you can't avoid it, fallback to custom a WebPart

Rule #14: Don’t use Session or in-memory data

- Avoid storing sessions or in-memory data on a single server
  - It wouldn’t scale across multiple servers
- Only cached data that can stay in-memory of a single server

- If you don’t adhere to this rule
  - The solutions will not scale horizontally
  - You will have to configure NLB affinity (very bad idea!)
  - The overall availability of the solutions will be reduced

Rule #15: Avoid single points of failure

- Create code ready to be executed on multiple servers
  - Avoid relying on web.config or app.config directly
  - Avoid looking for specific files on the file system
  - Use only shared resources accessed through a provider
    - To abstract from the physical location of data/content
- Mind data concurrency across multiple servers
  - Provide concurrency checks and optimistic concurrency

- If you don’t adhere to this rule
  - Your solution will not scale
  - You will have to activate affinity, which will impact availability, as well

questions?

HTTP://WWW.SHAREPOINT-REFERENCE.COM/BLOG/

@PAOLOPIA

thank you.

SHAREPOINT AND PROJECT CONFERENCE ADRIATICS
ZAGREB, 10/14/2014 - 10/15/2014