April 14th 2020

MDM Fuzzy Match Deep Dive

Augustin Chan achan@informatica.com
Development Architect, MDM ACE Team
Agenda

- Necessary Background
- Match Job Internals
- Match Pair Processing Details
- Match Batch Distribution
- Understanding the Cleanse Log
- Performance Tips
- Q&A

*Note: All logs and screenshots are from MDM 10.3 GA*
A Tale of Two Records

<table>
<thead>
<tr>
<th>PARTY_ROWID</th>
<th>DISPLAY_NAME</th>
<th>ADDRESS_LINE_1</th>
<th>ADDRESS_ROWID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 966</td>
<td>AUTOMOTION CORPORATION</td>
<td>225 BRAE BLVD</td>
<td>1021</td>
</tr>
<tr>
<td>2 991</td>
<td>AUTOMOTION</td>
<td>1740 BROADWAY</td>
<td>1046</td>
</tr>
<tr>
<td>3 991</td>
<td>AUTOMOTION</td>
<td>1740 BROADWAY</td>
<td>1069</td>
</tr>
<tr>
<td>4 991</td>
<td>AUTOMOTION</td>
<td>1740 BROADWAY</td>
<td>1068</td>
</tr>
<tr>
<td>5 991</td>
<td>AUTOMOTION</td>
<td>1740 BROADWAY</td>
<td>1053</td>
</tr>
</tbody>
</table>
Necessary Background
Fuzzy Keys Example

Example:

Automotion Corporation

→ Character level cleaning

AUTOMOTION CORPORATION

→ Edit List processing

AUTO MOTION

→ “Phonetics”

AT MATAN

→ Key Building

Key:

UYV>$F$$ MOTION AUTO
LUU>$WVA AUTO MOTION
LUVBC$$ AUTO
LUVBCGVA AUTOMOTION AUTO

Based on:
## Hub Console – Key Level

### Match/Merge Setup Details

<table>
<thead>
<tr>
<th>Properties</th>
<th>Paths</th>
<th>Match Columns</th>
<th>Match Rule Sets</th>
<th>Primary Key match rules</th>
<th>Match Key Distribution</th>
<th>Merge Settings</th>
</tr>
</thead>
</table>

#### Fuzzy Match Key
- **Key Type**: Organization Name
- **Key Width**: Extended
- **Path Component**: Root (Party)

### Match Columns

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Column Type</th>
<th>Path Component</th>
<th>Source Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address_Pan1</td>
<td>Fuzzy</td>
<td>Address</td>
<td>Address</td>
</tr>
<tr>
<td>Address_Pan2</td>
<td>Fuzzy</td>
<td>Address</td>
<td>Address</td>
</tr>
<tr>
<td>Attribute1</td>
<td>Fuzzy</td>
<td>Electronic Address</td>
<td>Party Electronic Address</td>
</tr>
<tr>
<td>Ex_Address_Type</td>
<td>Exact</td>
<td>Party Address Rel</td>
<td>Party Address Rel</td>
</tr>
<tr>
<td>Ex_Birthdate</td>
<td>Exact</td>
<td>Root</td>
<td>Party</td>
</tr>
<tr>
<td>Ex_Electronic_Address</td>
<td>Exact</td>
<td>Electronic Address</td>
<td>Party Electronic Address</td>
</tr>
<tr>
<td>Ex_Generation</td>
<td>Exact</td>
<td>Root</td>
<td>Party</td>
</tr>
<tr>
<td>Ex_Party_Type</td>
<td>Exact</td>
<td>Root</td>
<td>Party</td>
</tr>
<tr>
<td>Ex_Telecom</td>
<td>Exact</td>
<td>Telecom</td>
<td>Party Phone</td>
</tr>
<tr>
<td>Id</td>
<td>Fuzzy</td>
<td>Root</td>
<td>Party</td>
</tr>
<tr>
<td>Organization_Name</td>
<td>Fuzzy Match Key</td>
<td>Root</td>
<td>Party</td>
</tr>
<tr>
<td>Person_Name</td>
<td>Fuzzy</td>
<td>Root</td>
<td>Party</td>
</tr>
</tbody>
</table>

### Match Column Contents – Source Table: Party

- **Available columns:** Birthdate, DUNNS Number
- **Selected columns:** Display Name
Name3 Workbench Keys
### STRP Table Keys

```sql
select * from c_party_strp where rowid_object in ( '991', '966')
```

<table>
<thead>
<tr>
<th>SSA_KEY</th>
<th>ROWID_OBJECT</th>
<th>DATA_ROW</th>
<th>DATA_COUNT</th>
<th>SSA_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVU$&gt;VVA</td>
<td>966</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION CORPORATION</td>
</tr>
<tr>
<td>LVVBC$$-</td>
<td>966</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION CORPORATION</td>
</tr>
<tr>
<td>LVVBCFVA</td>
<td>966</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION CORPORATION</td>
</tr>
<tr>
<td>UYV&gt;$E$$</td>
<td>966</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION CORPORATION</td>
</tr>
<tr>
<td>LVU$&gt;VVA</td>
<td>991</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION</td>
</tr>
<tr>
<td>LVVBC$$-</td>
<td>991</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION</td>
</tr>
<tr>
<td>LVVBCFVA</td>
<td>991</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION</td>
</tr>
<tr>
<td>UYV&gt;$E$$</td>
<td>991</td>
<td>1</td>
<td>1</td>
<td>AUTOMOTION</td>
</tr>
</tbody>
</table>

© Informatica. Proprietary and Confidential.
Hub Console – Search Level

<table>
<thead>
<tr>
<th>Match Rule Set</th>
<th>Name</th>
<th>Search Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy, with, Exact</td>
<td>Typical</td>
<td></td>
</tr>
</tbody>
</table>

**Match Rules**

<table>
<thead>
<tr>
<th>Auto</th>
<th>Type</th>
<th>Accept L</th>
<th>Purpose/Level</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Exact</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Fuzzy</td>
<td>0</td>
<td>Address (Typical)</td>
<td>Address_Type, Ex_Party_Type, Ex_Telecom</td>
</tr>
<tr>
<td>No</td>
<td>Fuzzy</td>
<td>0</td>
<td>Resident (Typical)</td>
<td>Address_Part1 (Fuzzy), Ex_Party_Type, Ex_Telecom, Organization_Name (Fuzzy)</td>
</tr>
</tbody>
</table>
Search Ranges

• A range is a pair of 8 character strings
• Can be thought of as the fuzziness around a key
  - Give me all keys between 'UYV>$E$$' and 'UYV>$EZZ'
• Ranges are not persisted in any table!
• Some ranges can be seen from ThreadMonitor, or Match Summary in cleanse log
• MDM generates ranges at runtime with an ssa call
Name3 Workbench – Search Ranges
RangerWorker Summary – Top 10 Range Comparisons

[Ranger0] [INFO ] com.siperian.mrm.match.RangerWorker:
Top 10 Range Comparisons counts
Ranger0 Comparison Max Range 0 = 10 Q:2 DB:5 between 'UYV>$E$$' and 'UYV>$EZZ'
Ranger0 Comparison Max Range 1 = 8 Q:2 DB:4 between 'LVVBCFV>' and 'LVVBCFVB'
Ranger0 Comparison Max Range 2 = 0 Q:2 DB:0 between 'UYV>BGGC' and 'UYV>BGGF'
Ranger0 Comparison Max Range 3 = 0 Q:2 DB:0 between 'UYV>>VVG' and 'UYV>>VVJ'
Ranger0 Comparison Max Range 4 = 0 Q:2 DB:0 between 'UYV>$FV>' and 'UYV>$FVB'
Ranger0 Comparison Max Range 5 = 0 Q:2 DB:0 between 'UYV>$$$'$ and 'UYV>$$$/'
Ranger0 Comparison Max Range 6 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Comparison Max Range 7 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Comparison Max Range 8 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Comparison Max Range 9 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Total Ranges Processed  = 6
Ranger0 Total Candidates        = 14
Ranger0 Total Matches           = 1
Matcher Summary :total_calls: 14 SSA Matches: 14

‘Candidates’ really means candidate comparisons done by this thread (Ranger0).
SSA Matches = SSA calls
Range Queries and DB Counts

```
select * from c_party_strp where ssa_key between 'UYV>$E$' and 'UYV>$EZZ'

select * from c_party_strp where ssa_key between 'LVYBCFV>' and 'LVYBCFVB'

select * from c_party_strp where ssa_key between 'UYV>BGGC' and 'UYV>BGGF'

select * from c_party_strp where ssa_key between 'UYV>WVG>' and 'UYV>WVJ'

select * from c_party_strp where ssa_key between 'UYV>$FV>' and 'UYV>$FVB'

select * from c_party_strp where ssa_key between 'UYV>$F$' and 'UYV>$F$/'
```
Match Job Internals
If `cmx.server.match.distributed_match=1` then the job will be distributed across registered Process Servers.
Match Job Execution Overview

- **Tokenize** – If COMPLETE_STRIP_RATIO or STRIP_CTAS_DELETE_UPPER_LIMIT or STRIP_CTAS_DELETE_RATIO are exceeded, the entire STRP is rebuilt, with an exclusive lock on BO which prevents puts and merges! STRIP_CTAS merges existing and newly tokenized records into new STRP.
MDM Match Process – Data Prep, Range Gen

- **TokenizeWorker** – Generates fuzzy keys based on dirty records in Base Object. (ssan3_get_keys_encoded)
- **RangerProducer** – Reads data from _STRP and packages them into RangerNodeTransports containing 100 rangerNodes each (1 STRP row per rangerNode)
- **RangerWorker** – runRangeGen() reads these RangerNodeTransports fromProducerQ and calls ssan3_get_ranges_encoded to assign all search ranges within work range to their rangerNodes. These processed rangerNodeTransports are placed onto the ToRangeSorterQ for sorting.

For Distributed matching, only Ranges that fall within the work range for that Process Server are processed by the downstream SortManager and placed on the work queue.

Boxes in ORANGE are multi-threaded
RangerProducer – Prepare for Range Generation

- Reads _STRP rows where PREFERRED_KEY_IND = 1 and packages them into 100 rangerNode chunks inside RangerNodeTransports (1 row per rangerNode)
RangerWorker Range Generation and SortManager

RangerWorker Range Generation

Automotion Corporation
UYV>$E$$  UYV>$EZZ
UYV>$$$$  UYV>$$$/
UYV>$FV>  UYV>$FVB
UYV>>VVG  UYV>>VVJ
UYV>BGGC  UYV>BGGF
LVVBCFV>  LVVBCFVB

2 Records in match batch generate 12 total Ranges

SortManager packaging RangerNodes

UYV>$E$$  UYV>$EZZ
  Automotion Corporation
  Automotion
UYV>$$$$  UYV>$$$/
  Automotion Corporation
  Automotion
UYV>$FV>  UYV>$FVB
  Automotion Corporation
  Automotion
UYV>>VVG  UYV>>VVJ
  Automotion Corporation
  Automotion
UYV>BGGC  UYV>BGGF
  Automotion Corporation
  Automotion
LVVBCFV>  LVVBCFVB
  Automotion Corporation
  Automotion

RangerNodes placed onto work queue to be processed by RangerWorkers

Each RangerNode contains max 3000 search records (max_records_per_ranger_node cleanse prop)

Sorted into 6 RangerNodes
Ranges vs RangerNodes

RANGE GENERATION

[2020-03-28 00:49:58,170] [Ranger0] [DEBUG] com.siperian.mrm.match.RangerWorker: Total Records read = 2
[2020-03-28 00:49:58,170] [Ranger0] [DEBUG] com.siperian.mrm.match.RangerWorker: Total Ranges created = 12 (Range+Record combinations)
[2020-03-28 00:49:58,170] [Ranger0] [DEBUG] com.siperian.mrm.match.RangerWorker: Time Range Generation = 90 ms

SORTING+PACKAGING

[2020-03-28 00:49:58,175] [RangeSorter] [DEBUG] com.siperian.mrm.match.SortManager: Sort of 12 Records Done:8ms
[2020-03-28 00:49:58,175] [RangeSorter] [DEBUG] com.siperian.mrm.match.SortManager: ship the sorted ranges in memory back to the workers to match with
[2020-03-28 00:49:58,266] [RangeSorter] [DEBUG] com.siperian.mrm.match.SortManager: Time to distribute, from memory, 12 ranges with 6 candidates = 91 ms (RangerNodes)
MDM Match Process – Sorting, Matching

- **SortManager** – Extracts the rangeGen() rangerNodes from transports, sorts them, and creates new transports each with a single rangerNode that contains up to 3000 match records in their matchNodeArray (cleanse property `max_records_per_ranger_node`).
- **RangerWorker** – Performs Fuzzy match processing (`ssan3_match_encoded`) and exact string comparisons for exact fields. Threads = number of cleanse threads.
- **MatchGatherer** – gathers and removes duplicate match rows produced by the RangerWorkers, persists them to temp file for loading into mtch table.
Match Processing Details
RangerNodes, Rulesets, and Match Pairs
RangerNode Contents

Each element of the MatchNodeArray has the STRP data for its rowid:

Rowid 966: a050AUTOMOTION CORPORATION
b000c000d000g012organizationh000j004BILLk011PARK RIDGE 1014225 BRAE BLVD m01007656-1870n000o0135551212

Rowid 991: a050AUTOMOTION
b000c000d000g012organizationh000j004SHIPk009NEW YORK 10141740 BROADWAY m01010019-4315n000j004SHIPk009NEW YORK 10141740 BROADWAY m01010019-4315n000j004BILLk0010141740 BROADWAY m000n000j004BILLk009NEW YORK 10141740 BROADWAY m01010019-4315n000o0135551212 o0137771111 o0135557890
RangerNode Processing Details

Search Range: e.g., 'UYV>$E$$' - 'UYV>$EZZ'

Match Node Array: Up to 3000 Search Records from match batch which generate that range

And other stuff.....

Outer Loop Range Query for File Records:

```
SELECT ROWID_OBJECT, DATA_COUNT, SSA_DATA, DATA_ROW FROM " + stripTableName + " WHERE SSA_KEY BETWEEN ? AND ? + " AND INVALID_IND = 0" + " GROUP BY ROWID_OBJECT, DATA_COUNT, SSA_DATA, DATA_ROW " + " ORDER BY ROWID_OBJECT, DATA_ROW";
```

<table>
<thead>
<tr>
<th>RangerNode Search Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>991</td>
</tr>
<tr>
<td>966</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Rowid</th>
<th>File SSA_KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1042</td>
<td>UYV&gt;$E$$</td>
</tr>
<tr>
<td>1044</td>
<td>UYV&gt;$E$$</td>
</tr>
<tr>
<td>966</td>
<td>UYV&gt;$E$$</td>
</tr>
<tr>
<td>991</td>
<td>UYV&gt;$E$$</td>
</tr>
<tr>
<td>971</td>
<td>UYV&gt;$EBM</td>
</tr>
</tbody>
</table>

Comparison Matrix (8 Actual Comparisons):

<table>
<thead>
<tr>
<th></th>
<th>1042</th>
<th>991</th>
</tr>
</thead>
<tbody>
<tr>
<td>991</td>
<td></td>
<td></td>
</tr>
<tr>
<td>966</td>
<td></td>
<td></td>
</tr>
<tr>
<td>966</td>
<td></td>
<td></td>
</tr>
<tr>
<td>966</td>
<td></td>
<td></td>
</tr>
<tr>
<td>991</td>
<td></td>
<td></td>
</tr>
<tr>
<td>991</td>
<td></td>
<td></td>
</tr>
<tr>
<td>966</td>
<td></td>
<td></td>
</tr>
<tr>
<td>971</td>
<td></td>
<td></td>
</tr>
<tr>
<td>991</td>
<td></td>
<td></td>
</tr>
<tr>
<td>966</td>
<td></td>
<td></td>
</tr>
<tr>
<td>966</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fuzzy Matching on Distinct ‘Ruleset Nodes’

[2020-04-06 15:23:42.459] [Ranger0] [DEBUG]
com.siperian.mrm.match.RangerWorker: Matcher Rules: Ruleset 'Fuzzy_with_Exact' has 3 rule(s), Search Call:false

Exact Rule :1 AutoMerge Ind :false Asymetrical Ind:false
  Node Num :1 Exact Match, Match Column:'Ex_Party_Type'
  Match Column Id:6 Anti Match Ind:false
  Node Num :0 Exact Match, Match Column:'Ex_Address_Type' Match Column Id:9 Anti Match Ind:false
  Node Num :2 Exact Match, Match Column:'Ex_Telecom' Match Column Id:14 Anti Match Ind:false

Fuzzy Rule :2 AutoMerge Ind :false Asymetrical Ind:false
  Node Num :1 Exact Match, Match Column:'Ex_Party_Type'
  Match Column Id:6 Anti Match Ind:false
  Node Num :0 Exact Match, Match Column:'Ex_Address_Type' Match Column Id:9 Anti Match Ind:false
  Node Num :2 Exact Match, Match Column:'Ex_Telecom'
  Match Column Id:14 Anti Match Ind:false
  Node Num :3 SSA Matching on 'Address_Part1 Address_Part1' Column Id:11
  'Organization_Name Organization_Name' Column Id:0
  Match Level:Typical Geocode Radius:0 Match Purpose:Address

Fuzzy Rule :3 AutoMerge Ind :false Asymetrical Ind:false
  Node Num :1 Exact Match, Match Column:'Ex_Party_Type'
  Match Column Id:6 Anti Match Ind:false
  Node Num :2 Exact Match, Match Column:'Ex_Telecom'
  Match Column Id:14 Anti Match Ind:false
  Node Num :4 SSA Matching on 'Address_Part1 Address_Part1' Column Id:11
  'Person_Name Person_Name' Column Id:5
  Match Level:Typical Geocode Radius:0 Match Purpose:Resident

Total Nodes :12 Actual Nodes:5
Ruleset Optimization for Fuzzy Match Pair Evaluation

- A ruleset (rs) node is either an exact match field (Java String.regionMatches) or an SSA Purpose and its fuzzy fields (ssa match call).
- MDM determines the distinct set of rs nodes across all rules in ruleset. Fuzzy fields are associated with their Purpose.
- Fuzzy rules are evaluated in order for a given search+file record match pair.
- For each fuzzy rule - exact rs nodes are evaluated first.
  - If any rs node evaluates as false - current rule is non-match. Other fuzzy rules with this rs node are removed from further evaluation.
  - If any rs node evaluates as true – store true result in case this rs node is present in a subsequent fuzzy rule.
- Any Exact rules are processed in a later phase.
Internal Match Record (aka SearchNode/FileNode)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rowid '991'</td>
<td></td>
</tr>
<tr>
<td>ORGANIZATION_NAME</td>
<td>0, 0, AUTOMOTION</td>
</tr>
<tr>
<td>EX_GENERATION</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>DATE</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>ID</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>EX_BIRTHDATE</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>PERSON_NAME</td>
<td>0, 0, AUTOMOTION</td>
</tr>
<tr>
<td>EX_PARTY_TYPE</td>
<td>0, 0, Organization</td>
</tr>
<tr>
<td>ATTRIBUTE1</td>
<td>1, 0, 0, SIP null</td>
</tr>
<tr>
<td>EX ELECTRONIC_ADDRESS</td>
<td>1, 0, 0, SIP null</td>
</tr>
<tr>
<td>EX_ADDRESS_TYPE</td>
<td>2, 0, 0, 0, SHIF</td>
</tr>
<tr>
<td></td>
<td>2, 0, 0, 1, SHIF</td>
</tr>
<tr>
<td></td>
<td>2, 0, 0, 2, BILL</td>
</tr>
<tr>
<td></td>
<td>2, 0, 0, 3, BILL</td>
</tr>
<tr>
<td>ADDRESS_PART2</td>
<td>3, 0, 0, 0, 0, NEW YORK</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 1, NEW YORK</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 2, SIP null</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 3, NEW YORK</td>
</tr>
<tr>
<td>ADDRESS_PART1</td>
<td>3, 0, 0, 0, 0, 1740 BROADWAY</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 1, 1740 BROADWAY</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 2, 1740 BROADWAY</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 3, 1740 BROADWAY</td>
</tr>
<tr>
<td>POSTAL AREA</td>
<td>3, 0, 0, 0, 10019-4315</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 1, 10019-4315</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 2, SIP null</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 3, 10019-4315</td>
</tr>
<tr>
<td>POSTAL_SUB3</td>
<td>3, 0, 0, 0, SIP null</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 1, SIP null</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 2, SIP null</td>
</tr>
<tr>
<td></td>
<td>3, 0, 0, 3, SIP null</td>
</tr>
<tr>
<td>EX TELECOM</td>
<td>4, 0, 0, 3, 0, 5551212</td>
</tr>
<tr>
<td></td>
<td>4, 0, 0, 3, 1, 7771111</td>
</tr>
<tr>
<td></td>
<td>4, 0, 0, 3, 2, 5557890</td>
</tr>
<tr>
<td>TELEPHONE_NUMBER</td>
<td>4, 0, 0, 3, 0, 5551212</td>
</tr>
<tr>
<td></td>
<td>4, 0, 0, 3, 1, 7771111</td>
</tr>
<tr>
<td></td>
<td>4, 0, 0, 3, 2, 5557890</td>
</tr>
</tbody>
</table>

Rowid '966'
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORGANIZATION_NAME</td>
<td>0, 0, AUTOMOTION CORPORATION</td>
</tr>
<tr>
<td>EX_GENERATION</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>DATE</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>ID</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>EX_BIRTHDATE</td>
<td>0, 0, SIP null</td>
</tr>
<tr>
<td>PERSON_NAME</td>
<td>0, 0, AUTOMOTION CORPORATION</td>
</tr>
<tr>
<td>EX_PARTY_TYPE</td>
<td>0, 0, Organization</td>
</tr>
<tr>
<td>ATTRIBUTE1</td>
<td>1, 0, 0, SIP null</td>
</tr>
<tr>
<td>EX ELECTRONIC_ADDRESS</td>
<td>1, 0, 0, SIP null</td>
</tr>
<tr>
<td>EX_ADDRESS_TYPE</td>
<td>2, 0, 0, 0, BILL</td>
</tr>
<tr>
<td>ADDRESS_PART2</td>
<td>3, 0, 0, 0, PARK RIDGE</td>
</tr>
<tr>
<td>ADDRESS_PART1</td>
<td>3, 0, 0, 0, 225 BREA BLVD</td>
</tr>
<tr>
<td>POSTAL AREA</td>
<td>3, 0, 0, 0, 07656-1870</td>
</tr>
<tr>
<td>POSTAL_SUB3</td>
<td>3, 0, 0, 0, SIP null</td>
</tr>
<tr>
<td>EX TELECOM</td>
<td>4, 0, 0, 0, 0, 5551212</td>
</tr>
<tr>
<td>TELEPHONE_NUMBER</td>
<td>4, 0, 0, 0, 0, 5551212</td>
</tr>
</tbody>
</table>
Fuzzy Rules - Exact Multi-Field Handling in MDM

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX_TELECOM</td>
<td>5557890</td>
</tr>
<tr>
<td>EX_TELECOM</td>
<td>5551212</td>
</tr>
<tr>
<td>EX_TELECOM</td>
<td>7771111</td>
</tr>
<tr>
<td>EX_TELECOM</td>
<td>2225555</td>
</tr>
<tr>
<td>EX_TELECOM</td>
<td>5551212</td>
</tr>
</tbody>
</table>

We first check if file record string is null, then we check if string lengths are the same before we try to find string match

All combinations will be matched until an exact string match is found:

• 5557890 to 2225555
• 5551212 to 2225555
• 7771111 to 2225555
• 5557890 to 5551212
• 5551212 to 5551212 – exact match found. stop

We will do all 6 comparisons only if we cannot stop early with 100%.
SSA Layout for Fuzzy Match

**Rowid 991**

a050AUTOMOTION  
b000c000d000g012Organizationh000j004SHIPk009NEW YORK 10141740 BROADWAY  
m01010019-4315n000j004SHIPk009NEW YORK 10141740 BROADWAY m01010019-4315n000j004BILLk00010141740 BROADWAY m000n000j004BILLk009NEW YORK 10141740 BROADWAY m01010019-4315n000o0135551212 o0137771111 o0135557890

\[=\text{Address Part1,222,28,Address Part2,196,18,Telephone Number,598,26,Address Part1,336,28,Address Part2,310,18,Telephone Number,632,26,Address Part1,432,28,Telephone Number,666,26,Address Part1,526,28,Address Part2,500,18}\]

**NOTE:** Layout string lengths are doubled because `cmx.server.match.server_encoding=1`

**Match Call Pseudo-Code:**

```python
ssa.match(searchNode, searchLayout, fileNode, fileLayout)
```
Multi-Field Handling for Match in SSA

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address #1 1740 Broadway</td>
<td>1 Fern St</td>
</tr>
<tr>
<td>Address #2 6555 Quince Rd Ste 400</td>
<td>1740 Broadway</td>
</tr>
<tr>
<td>Address #3 12012 N Mo Pac Expy</td>
<td>109 E Main St</td>
</tr>
</tbody>
</table>

All combinations will be matched until a 100 pct match is found:
- 1740 Broadway to 1 Fern St
- 6555 Quince Rd Ste 400 To 1 Fern St
- 12012 N Mo Pac Expy To 1 Fern St
- 1740 Broadway to 1740 Broadway – 100 pct match, we stop

We will do all 9 comparisons only if we cannot stop early with 100%. Only 1 combination needs to match for the records to match.
### SSA Multi-Field Matching in Workbench

<table>
<thead>
<tr>
<th>Record 1</th>
<th>Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address #1</strong> 1740 Broadway</td>
<td>1 Fern St</td>
</tr>
<tr>
<td><strong>Address #2</strong> 6555 Quince Rd Ste 400</td>
<td>1740 Broadway</td>
</tr>
<tr>
<td><strong>Address #3</strong> 12012 N Mo Pac Expy</td>
<td>109 E Main St</td>
</tr>
</tbody>
</table>

![Database Matching Example](image)

- Session: 3984589
- System: default
- Population: demo
- PURPOSE: Address MATCH_LEVEL: Typical

**Response Messages**: 0

**Search Data**:
- `*Address_Part1*1740 Broadway`  
- `*Address_Part1*6555 Quince Rd Ste 400`  
- `*Address_Part1*12012 N Mo Pac Expy`

**File Data**:  
- `*Address_Part1*1 Fern St`  
- `*Address_Part1*1740 Broadway`  
- `*Address_Part1*109 E Main St`

- Hex

**Decision**: A
**Score**: 100
Subtype Match = Matching with Subsets of Child Data

Matching is done for each common subtype in the match pair. Matching stops when a match is found. Only Fuzzy rules can have subtypes.

Each color represents a logical group tied to a subtyped value. Each group will have its own layout for the ssa call.
Match Batch Distribution
Process Server Work Range (scaled to 1000)

[2020-03-31 12:55:45,205] [default task-12] [DEBUG]
com.siperian.mrm.util.distributed.DistManager:

This server is:http://torapp2:8380/cleanse/
Server:Port is torapp1:8380 Match true Cleanse true Match Mode 3 online flag true
Included MatchServer-- Server:Port is torapp1:8380 Node Count 24 node Capability Multiplier 1.0
Server:Port is torapp2:8380 Match true Cleanse true Match Mode 3 online flag true
Included MatchServer-- Server:Port is torapp2:8380 Node Count 24 node Capability Multiplier 1.0
Number of servers :2 Number of Nodes :48.0
Work Range from 1000 for Server 0 is 0 to 499
Work Range from 1000 for Server 1 is 500 to 999
MDM Match Job Distribution – Based on Search Range

- All Search Ranges have their hashCode calculated during Range Generation, scaled to 1000. If the value falls within that nodes UoW range, then the search range is passed to the SortManager.
- Every process server generates all ranges and determines its own ranges concurrently

"UYV>$FV>UYV>$FVB".hashCode() % 1000
(int) 648
Server 1

"UYV>>VVGUYV>>VVJ".hashCode() % 1000
(int) 341
Server 0
Understanding the Cleanse Log
Producer – Reading Key Data from the STRP

[RangerProducer] [DEBUG] com.siperian.mrm.match.RangerProducer: Starting RangerProducer
[RangerProducer] [DEBUG] com.siperian.mrm.match.MatchProperties: No Prefetch setting in properties file. Setting to Default: 1000
[RangerProducer] [DEBUG] com.siperian.mrm.match.RangerProducer: Before SQL Max Memory = 6442450944 Total Memory = 2456813568 Free Memory = 1342555808
[RangerProducer] [INFO ] com.siperian.mrm.match.RangerProducer: Start reading Data from STRP Table, sql is: SELECT /*+ PARALLEL ORDERED USE_HASH (B, A) */ S.ROWID_OBJECT, S.DATA_COUNT, S.SSA_DATA FROM T$MAQ_PARTY B INNER JOIN C_PARTY_STRP S ON (S.ROWID_OBJECT = B.ROWID_OBJECT AND S.PREFERRED_KEY_IND = 1 AND S.INVALID_IND = 0) ORDER BY S.SSA_KEY, S.ROWID_OBJECT, S.DATA_ROW
[RangerProducer] [DEBUG] com.siperian.mrm.match.RangerProducer: After SQL Max Memory = 6442450944 Total Memory = 2456813568 Free Memory = 1342501032
[RangerProducer] [DEBUG] com.siperian.mrm.match.MatchProperties: No ranger_producer_to_ranger_worker_buffer setting in properties file. Setting to Default: 100
[RangerProducer] [INFO ] com.siperian.mrm.match.RangerProducer: Finished fetching data from database. Fetched 3 BOs
[RangerProducer] [DEBUG] com.siperian.mrm.match.RangerProducer: Total Search Candidates: 2
[RangerProducer] [DEBUG] com.siperian.mrm.match.RangerProducer: Processed 2 RangerProducer completed in 0.099 ( 0.099 sec)
RangerWorker – Generating Search Ranges

com.siperian.mrm.match.RangerWorker: Search Level Set for Execution: Narrow
com.siperian.mrm.match.RangerWorker: File loading is used
com.siperian.mrm.match.RangerWorker: Starting Ranger0
com.siperian.mrm.match.RangerWorker: Total Records read = 2
com.siperian.mrm.match.RangerWorker: Total Ranges created = 12
com.siperian.mrm.match.RangerWorker: Time Range Generation = 87 ms

NOTE: Above Search Level log message is incorrect and shows the search level for realtime SearchMatch. Look further up in the log for the Search Level for the ruleset to see the actual search level used for Range Generation, e.g:

[2020-04-01 01:38:03,698] [HTTP-276] [DEBUG]
com.siperian.mrm.match.SSAMeta: MatchRuleSet Fuzzy_Rule_Only Search Level is: Typical
SortManager – Packaging RangerNodes

[2020-03-28 00:49:58,086] [RangeSorter] [DEBUG]
com.siperian.mrm.match.RangeSorter:
RangeSorter: Starting RangeSorter
RangeSorter: Start gathering output from worker threads
RangeSorter: got end of loading, number closed:1
RangeSorter: Finished sort. Time to gather 2 records, 12 ranges = 1585327798167 ms. Start handing back to workers for matching
SortManager: Sort Starting
SortManager: Sort of 12 Records Done:8ms
SortManager: ship the sorted ranges in memory back to the workers to match with
SortManager: Time to distribute, from memory, 12 ranges with 6 candidates = 91 ms
RangeSorter: RangeSorter completed in 0.182 ( 0.182 sec )

Candidates = rangerNodes!!! Log statement is misleading.
Opening the SSA Session

[2020-03-28 00:49:58,203] [Ranger0] [INFO ]
com.siperian.mrm.match.SsaBase:
*** SSA Session opened: s_mdt>     Mar 29 2017 16:59:20 10.0.0.100
s_mdt   MDT      1.8.2.11MSVS2008 2014-02-19 18:11:41
Population File = /home/infa/infamdm/hub/cleanse/resources/match/demo
SECTION: E1                               SSA
        SSA-NAME3 00302n3sgxx     E1
YY0031          0000EXPDAT   2014-02-19 18:11:43.397000

Red – SSA Library Version (loaded from cleanse/lib) – 10.0.0.100
SSA Client Jar Version (from siperian-mrm.ear):
    09/25/2018 04:27 PM         200,658 ssan3-10.1.0.jar
ThreadMonitor – Totals and Current Snapshots

[2019-02-10 22:49:58,342] [RangerManager] [INFO ]
com.siperian.mrm.util.threads.ThreadMonitor: Dist:Ranger15 Matching TCan:1891941329

- **TCan**: total number of database candidates retrieved across all rangerNodes processed by this thread
- **Tgr**: Total number of comparisons (exact and ssa) where the search record has a lower rowid than the file record performed across all rangerNodes processed by this thread. Only accurate if “Match Only Previous Rowid Objects” is enabled
- **TSSA**: Total number of ssa comparisons performed across all rangerNodes processed by this thread
- **TM**: Total matches found across all rangerNodes processed by this thread
- **TR**: Total rangerNodes processed by this thread up to this point
- **Cur RI**: The rowid of the db file record currently being processed
- **Cur Range**: The search range of the current rangerNode being processed by this thread at this point in time
- **CompsPerRange**: The number of search records * db comparisons done so far for the current rangerNode
RangerWorker Summary – Top 10 Range Counts

[Ranger0] [INFO ] com.siperian.mrm.match.RangerWorker:
Top 10 Range counts:
Ranger0 Max Range 0 = 5 between 'UYV>$E$$' and 'UYV>$EZZ'
Ranger0 Max Range 1 = 4 between 'LVVBCFV>' and 'LVVBCFVB'
Ranger0 Max Range 2 = 0 between 'UYV>BGGC' and 'UYV>BGGF'
Ranger0 Max Range 3 = 0 between 'UYV>>VVG' and 'UYV>>VVJ'
Ranger0 Max Range 4 = 0 between 'UYV>$FV>' and 'UYV>$FVB'
Ranger0 Max Range 5 = 0 between 'UYV>$$$$' and 'UYV>$$$/'
Ranger0 Max Range 6 = 0 between 'null' and 'null'
Ranger0 Max Range 7 = 0 between 'null' and 'null'
Ranger0 Max Range 8 = 0 between 'null' and 'null'
Ranger0 Max Range 9 = 0 between 'null' and 'null'
RangerWorker Summary – Top 10 Range Comparisons

[Ranger0] [INFO ] com.siperian.mrm.match.RangerWorker:
Top 10 Range Comparisons counts
Ranger0 Comparison Max Range 0 = 10 Q:2 DB:5 between 'UYV>$E$$' and 'UYV>$EZZ'
Ranger0 Comparison Max Range 1 = 8 Q:2 DB:4 between 'LVVBCFV>' and 'LVVBCFVB'
Ranger0 Comparison Max Range 2 = 0 Q:2 DB:0 between 'UYV>BGGC' and 'UYV>BGGF'
Ranger0 Comparison Max Range 3 = 0 Q:2 DB:0 between 'UYV>>VVG' and 'UYV>>VVJ'
Ranger0 Comparison Max Range 4 = 0 Q:2 DB:0 between 'UYV>$FV>' and 'UYV>$FVB'
Ranger0 Comparison Max Range 5 = 0 Q:2 DB:0 between 'UYV>$$$$' and 'UYV>$$$/'
Ranger0 Comparison Max Range 6 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Comparison Max Range 7 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Comparison Max Range 8 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Comparison Max Range 9 = 0 Q:0 DB:0 between 'null' and 'null'
Ranger0 Total Ranges Processed = 6
Ranger0 Total Candidates = 14
Ranger0 Total Matches = 1
Matcher Summary : total_calls: 14 SSA Matches: 14

‘Candidates’ really means candidate comparisons done by this thread (Ranger0).

These comparison counts can be used to choose a relevant Dynamic Match Analysis Threshold. A DMAT of 8 would still process rangerNode 'LVVBCFV>' and 'LVVBCFVB’ but skip 'UYV>$E$$' and 'UYV>$EZZ'
Fuzzy Match Summary

[2020-03-31 15:23:39,256] [HTTP-219] [INFO ] com.siperian.mrm.match.Ranger:
Total Records to Match : 2
Total DB Records Read : 9
Total Possible Matches : 14
Total Greater : 14
Total SSA : 14
Total Matches : 1
Total Auto Matches : 0
Total Manual Matches : 1
Total Ranges Created : 12
Total Ranges Processed : 6
Total Rejects From Exact : 0
Total Rejects From Ssa : 13
Total Match Calls : 14
Total time used 1 secs

Ranges Created = total search ranges generated across all records in match batch
Ranges Processed = rangerNodes processed across all rangerWorkers
Fuzzy Match Summary Legend

**Total Records to Match**: Number of records in match batch

**Total DB Records Read**: Candidates read from STRP table across all RangerWorker threads

**Total Possible Matches**: Number of candidates evaluated across all RangerWorker threads

**Total Greater**: Candidates whose rowid is greater than the search record’s rowid (only correct if using Match Only Previous Rowid Objects, otherwise same as Possible Matches)

**Total SSA**: Number of SSA Purpose evaluations across all RangerWorker threads

**Total Matches**: # of match rows collected by MatchGatherer, net of any dupes found by MatchGatherer

**Total Auto Matches**: of the Total Matches found, how many are from fuzzy automerge rules

**Total Manual Matches**: of the total matches found, how many are from fuzzy manual merge rules

**Total Ranges Created**: total number of search ranges generated by rangeGen() across all RangerWorkers

**Total Ranges Processed**: total number of rangerNodes processed across all RangerWorkers

**Total Rejects From Exact**: exact comparisons that failed evaluation, including child data (does not count: exact only rules, null matching, or segment matching)

**Total Rejects From Ssa**: fuzzy comparisons that failed ssa purpose evaluation

**Total Match Calls**: Total search rec to file rec comparisons across all RangerWorkers
Begin Exact Match Phase


**Exact Rule:** 1 AutoMerge Ind: false Asymetrical Ind: false
- Node Num: 1 Exact Match, Match Column: 'Ex_Party_Type' Match Column Id: 6 Anti Match Ind: false
  - Node Num: 0 Exact Match, Match Column: 'Ex_Address_Type' Match Column Id: 9 Anti Match Ind: false
  - Node Num: 2 Exact Match, Match Column: 'Ex_Telecom' Match Column Id: 14 Anti Match Ind: false

This rule will be processed with a normal select join clause

[2020-04-06 15:23:42,952] [HTTP-209] [DEBUG] com.siperian.mrm.match.SSAMeta:
Node is Node Num: 1 Exact Match, Match Column: 'Ex_Party_Type' Match Column Id: 6 Anti Match Ind: false
Match Column is Ex_Party_Type Depth 1 Table: C_PARTY_MTIP
Node is Node Num: 0 Exact Match, Match Column: 'Ex_Address_Type' Match Column Id: 9 Anti Match Ind: false
Match Column is Ex_Address_Type Depth 2 Table: C_MT_PARTY_ADDRESS_REL
Node is Node Num: 2 Exact Match, Match Column: 'Ex_Telecom' Match Column Id: 14 Anti Match Ind: false
Match Column is Ex_Telecom Depth 2 Table: C_MT_TELECOM
Improving Performance
Performance Tips

- Use exact fields as much as possible
- Avoid subtype match – try filtered match path as workaround
- If few straggling RangerWorkers finish much later
  - Consider decreasing max_records_per_ranger_node to smooth out uneven rangerNode processing times and improve concurrency
  - Configure Dynamic Match Analysis Threshold if match quality can be sacrificed for performance
- Analyze RangerWorker Summary Top Counts
  - Check unusually large range counts against STRP table
    - If SSA_DATA shows keys from noise, add noise words with Population Override Manager
    - If SSA_DATA shows numerous keys from valid data, adjust frequency table with Population Override Manager to mark this data as ‘common’

**CAUTION:** Any population changes should be well tested before promotion. Reach out to IPS/GCS/ACE for help as needed.
Thank You

achan@informatica.com
Matching on Distinct ‘Ruleset Nodes’

<table>
<thead>
<tr>
<th>Match Rules</th>
<th>Purpose/Level</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>Type</td>
<td>Accept L.</td>
</tr>
<tr>
<td>No</td>
<td>Fuzzy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Fuzzy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ruleset 'Fuzzy_Rule Only' has 2 rule(s), Search Call:false
Fuzzy Rule :1 AutoMerge Ind :false Asymetrical Ind: false
Node Num :1 Segment Match, Match Column:'Ex_Party_Type' Match Column Id:6 Segment Value/s:'Organization'
Node Num :0 Exact Match, Match Column:'Ex_Address_Type' Match Column Id:9 Anti Match Ind: false
Node Num :2 Exact Match, Match Column:'Ex_Telecom' Match Column Id:14 Anti Match Ind: false
Node Num :3 SSA Matching on 'Address_Part1 Address_Part1' Column Id:11 'Address_Part2 Address_Part2' Column Id:10 'Organization_Name Organization_Name' Column Id:0 Match Level: Typical Geocode Radius: 0 Match Purpose: Division

Fuzzy Rule :2 AutoMerge Ind :false Asymetrical Ind: false
Node Num :4 Segment Match, Match Column:'Ex_Party_Type' Match Column Id:6 Segment Value/s:'Person'
Node Num :2 Exact Match, Match Column:'Ex_Telecom' Match Column Id:14 Anti Match Ind: false
Node Num :5 SSA Matching on 'Address_Part1 Address_Part1' Column Id:11 'Person_Name Person_Name' Column Id:5 Match Level: Typical Geocode Radius: 0 Match Purpose: Resident
Total Nodes :10 Actual Nodes: 6