
Zero Downtime
Dump/Load
OMEGASERVE®
The last word in uptime protection!

ABOUT THE SPEAKERS

- Mike Furgal
  - Progress OpenEdge expert since 1989
  - Former MDBA director for Progress
- Vladimir Soloviev
  - Senior Progress/QAD and Infor ERP Consultant
  - Managing 100+ (10TB) databases, logistics business
- Nectarios Daloglou
  - President of OmegaServe and Senior DBA
  - Working with Progress & QAD for over 20 years

OMEGASERVE®
The last word in uptime protection!

ABOUT OmegaServe

- Managed DBA Services
  - 24/7 Monitoring & Support
  - Database Reorgs
  - Migrations
  - Health Checks
  - Business continuity planning and implementations
- Specialized in ERPs such as QAD
- Our goal: Maximize Uptime and Performance!
- Official ProTop Reseller

This session will describe the process of using Table Partitioning to be able to dump and load a table with little to no interruption to the application or their users.

MEGASERVE®
The last word in uptime protection!


WARNING!
Your mileage ~~may~~ will vary
Proceed at your own risk!

2016 PUG REVISITED

- Add an integer field to a table with initial value of 0
- Add an index on this new field
- Enable Table Partitioning on the database
- Make a list partition on this field
  - Value of 0 is in the current storage area
  - Value of 1 is in a different storage area
- Run a program to change all the values of the new field to 1 to migrate the data

DEMO

- Create a demo database
- Add 500,000 rows to the database
- Make it fragmented
- Add an integer field “my area”
- Add an index to my area
- Enable Table Partitioning
- Run an application
- Move the data
- Check the results

MEGASERVE®
The last word in uptime protection!

DEMO

MEGASERVE®
The last word in uptime protection!

BENEFITS

- You have control over the rate that rows move to the new Storage Area
- Restartable
- Can be run over the course of weeks during off hours
- Can make a temporary Storage Area, move smaller table to the temporary area, then back to their normal area
- Does not cause an OE Replication rebaseline

OMEGASERVE®
The last word in uptime protection!

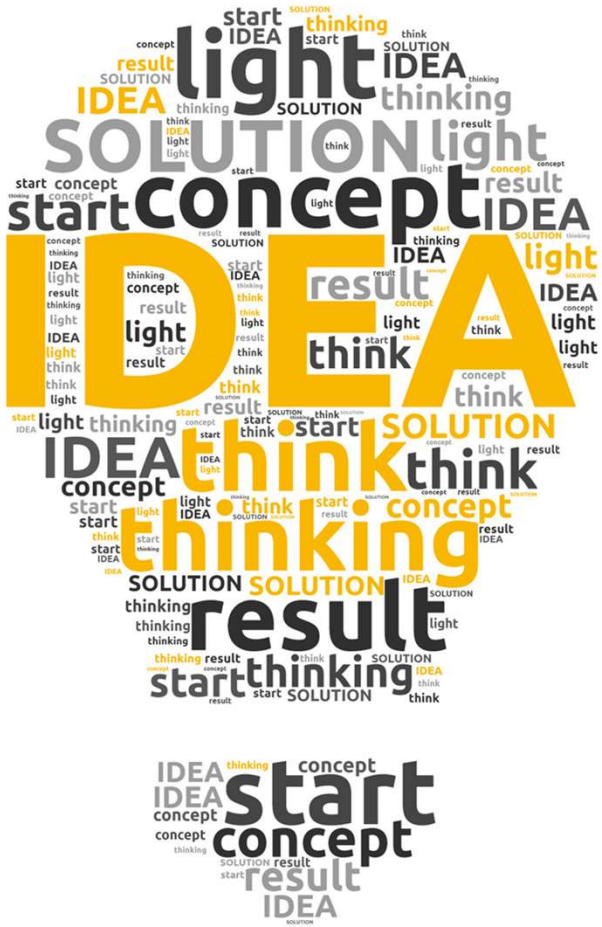
CAVEATS

- Table Partitioning or Advanced Enterprise RDBMS license required
- The new area will now be part of the database
- The old area High Water Mark will not change unless truncated
  - Offline
  - OE Replication rebaseline
- Backups will be larger
- Beware of code like this:

  FIND <table> NO-LOCK WHERE ….

  RUN update_table.p (INPUT RECID(<table>)).

INSPIRED?



V

LANDSCAPE

- 100+ Production databases across 20 servers
- 3000+ Customer users
- Service Level Agreement (SLA)
- Limited downtime for maintenance

WHY DO THIS?

- Improve application performance and user experience
- Reorganize database online
- Repeatable procedure across 100+ database environments
- Avoid dump/load
  - Long outage
  - Record validation
  - Risk factor

MEGASERVE®
The last word in uptime protection!

Eliminate logical scatter

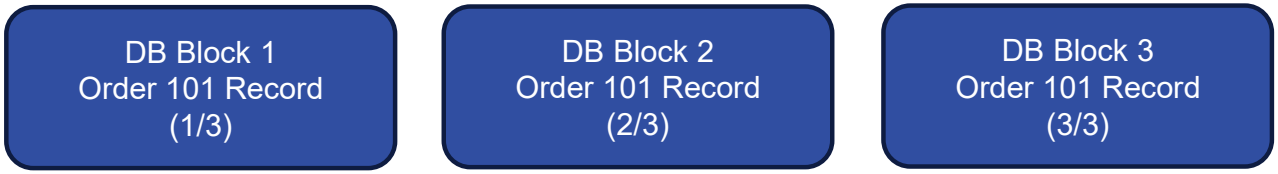
1	250	167	40	407	424	177	124	456	321	286	46	37	390	244	259	202	135	336	359	480	108	83	274	462	394	224	30	341	441	451	95	51	316	172	104	32	374	150	323
219	282	436	23	62	400	75	427	473	458	67	41	125	326	198	309	152	213	284	255	234	65	385	267	248	337	439	147	383	460	364	474	419	39	170	465	405	154	252	80
414	191	268	197	144	411	164	84	130	60	123	461	377	307	272	11	221	450	76	351	3	109	290	128	380	118	429	74	345	116	63	121	18	357	468	106	420	434	295	155
475	301	141	22	397	437	183	300	449	218	233	347	261	187	196	9	143	146	20	17	24	331	243	303	45	452	192	269	79	256	422	239	199	7	367	356	153	432	446	56
157	131	194	312	457	242	349	102	348	190	49	260	201	245	291	158	85	204	388	416	319	174	181	279	391	6	220	113	280	225	58	138	229	477	313	373	200	103	69	217
235	230	15	289	276	72	188	264	50	298	61	402	184	318	470	71	270	241	119	127	163	148	360	262	297	142	251	165	314	384	173	294	305	315	342	13	471	363	186	472
35	410	403	404	296	479	110	333	185	476	382	33	401	317	205	379	418	93	231	240	73	285	171	136	425	57	226	206	393	166	232	114	408	343	339	223	160	370	31	381
101	389	327	392	21	212	409	358	145	53	406	299	426	365	362	78	448	350	98	92	322	175	54	90	96	354	70	105	431	126	249	273	43	34	330	36	328	338	447	361
412	454	100	2	320	423	26	210	387	263	162	208	139	292	19	428	247	87	52	463	209	133	430	455	180	195	398	266	111	134	467	435	372	308	156	310	168	469	169	369
5	253	81	112	48	258	215	352	444	4	288	86	120	466	47	12	478	443	395	216	324	55	340	203	334	287	332	417	355	413	386	129	64	91	459	438	16	265	464	97
227	42	346	325	421	44	306	182	277	238	302	179	371	376	214	82	77	8	353	28	246	396	207	211	433	378	14	271	25	140	335	68	445	151	375	59	161	283	88	329
453	122	38	293	415	189	99	159	137	193	117	27	237	115	94	236	222	304	178	89	176	281	440	29	278	254	275	10	107	66	368	344	442	366	149	311	257	399	132	228

Image courtesy of Progress – Mike Furgal – Walking Through a Database Health Check

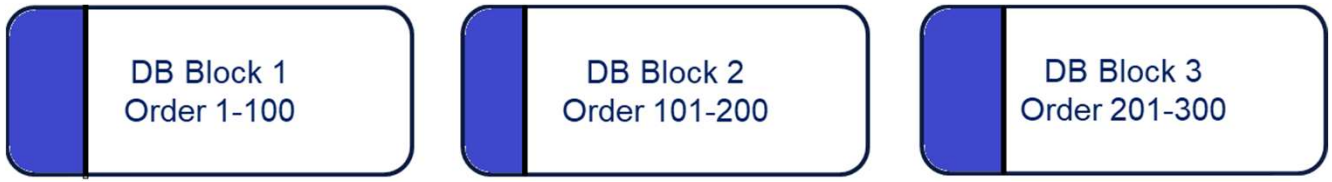
OMEGASERVE®
The last word in uptime protection!

Choose optimal records per block

- RPB too high can cause fragmentation



- RPB too low can cause additional space usage



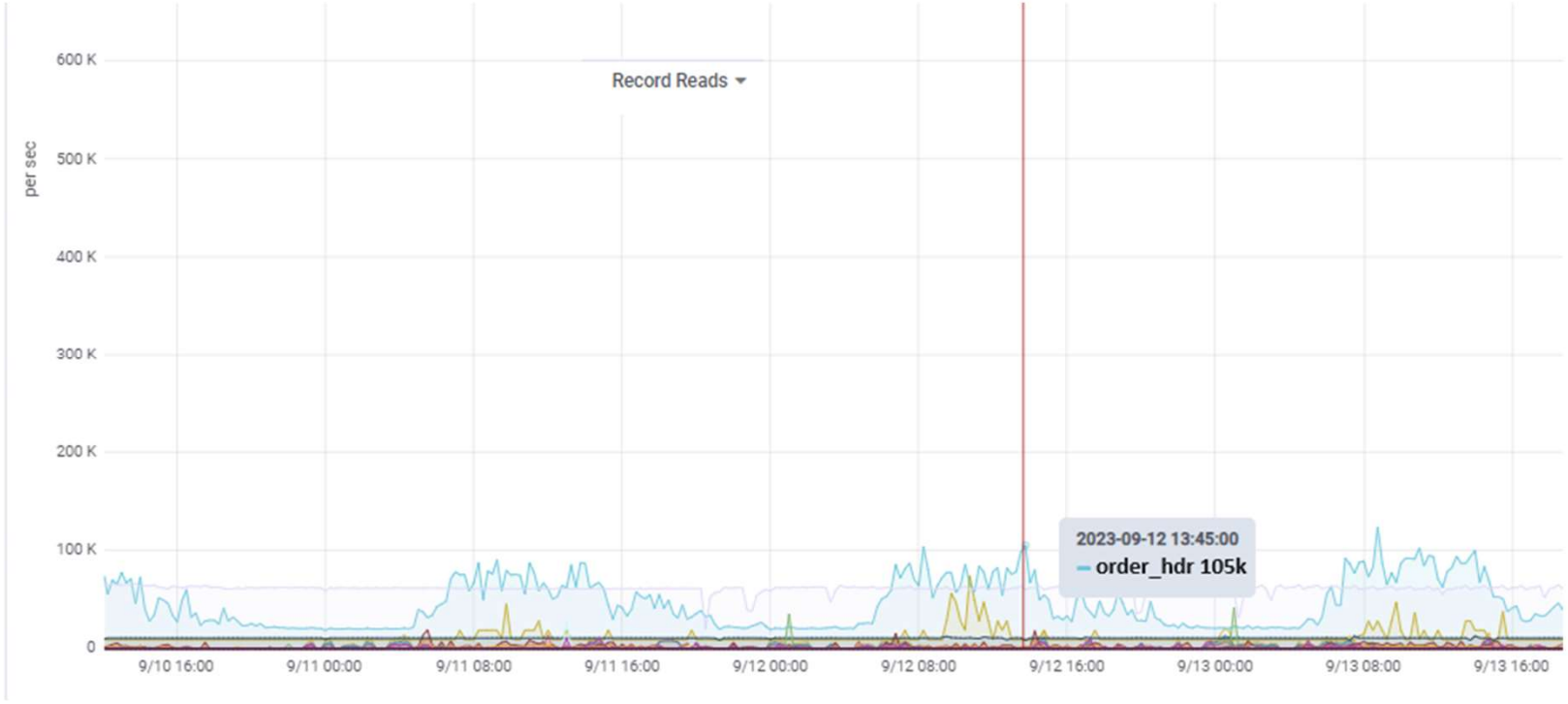
TABLES SELECTED

			-Record Size (B)--------			Fragments--------		Scatter
Table	Records	Size	Min	Max	Mean	Count	Factor	Factor
PUB.oder_track	8405364	5.9G	240	1764	747	15215285	1.8	1.0
PUB.order_hdr	9311997	10.1G	747	1598	1164	11503261	1.2	1.0
PUB.audit	1488106670	258.2G	63	10294	186	1488106675	1.0	1.0

OMEGASERVE®
The last word in uptime protection!

ORDER_HDR

- Busiest table in the database



SUMMARY OF STEPS

- Introduce INT field/index for pre-selected tables
- Introduce new areas dedicated for archiving
- Enable Table Partitioning
- Create Partition Policy
- Relocate records to new area

ENVIRONMENT DETAILS

- OpenEdge 11.7
- CentOS Linux 7
  - 12 x 2.69 GHz Xeon Platinum
  - 48GB memory
- 700GB Database
- Logistics application

V

Introduce INT field/indexes

ADD FIELD “order_hdr_arc" OF “order_hdr" AS integer
FORMAT ">>>>>9"
<…>

ADD INDEX “order_hdr_arc_idx" ON “order_hdr"
INDEX-FIELD “order_hdr_arc" ASCENDING

ADD FIELD “order_track_arc" OF “order_track" AS integer
FORMAT ">>>>>9"
<…>

ADD INDEX “order_track_arc_idx" ON “order_track_arc"
INDEX-FIELD “order_track_arc" ASCENDING

ADD FIELD "audit_arc" OF "audit" AS integer
FORMAT ">>>9"

ADD INDEX "audit_arc_idx" ON "audit"
INACTIVE
INDEX-FIELD "audit_arc" ASCENDING

V

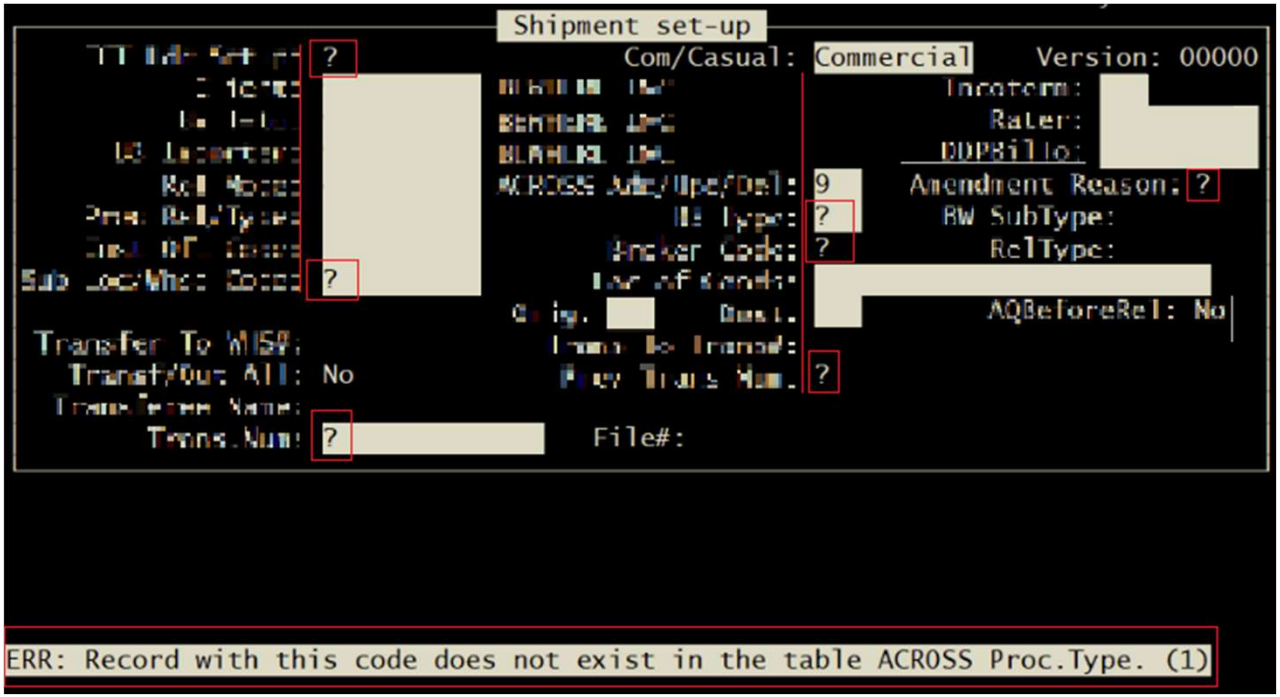
OMEGASERVE®
The last word in uptime protection!

LOAD AS INACTIVE

- Loading ACTIVE audit index: 4hrs26min
- Loading INACTIVE audit index: <1min
- Audit index rebuild: 10min

V

OOPS! SCHEMA CHANGE REQUIRED COMPILE

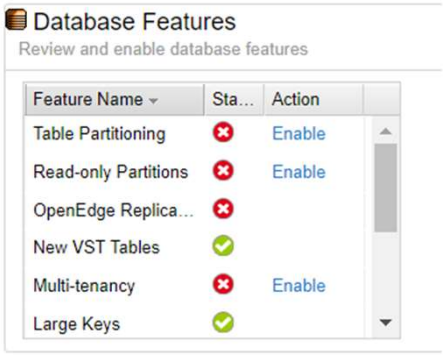


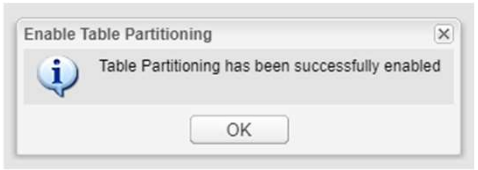
ERR: Record with this code does not exist in the table ACROSS Proc.Type. (1)

V

MEGASERVE®
The last word in uptime protection!

ENABLE TABLE PARTITIONING





OR in CHUI:
proutil <DB> -C enable table partitioning

V

ENABLE TABLE PARTITIONING

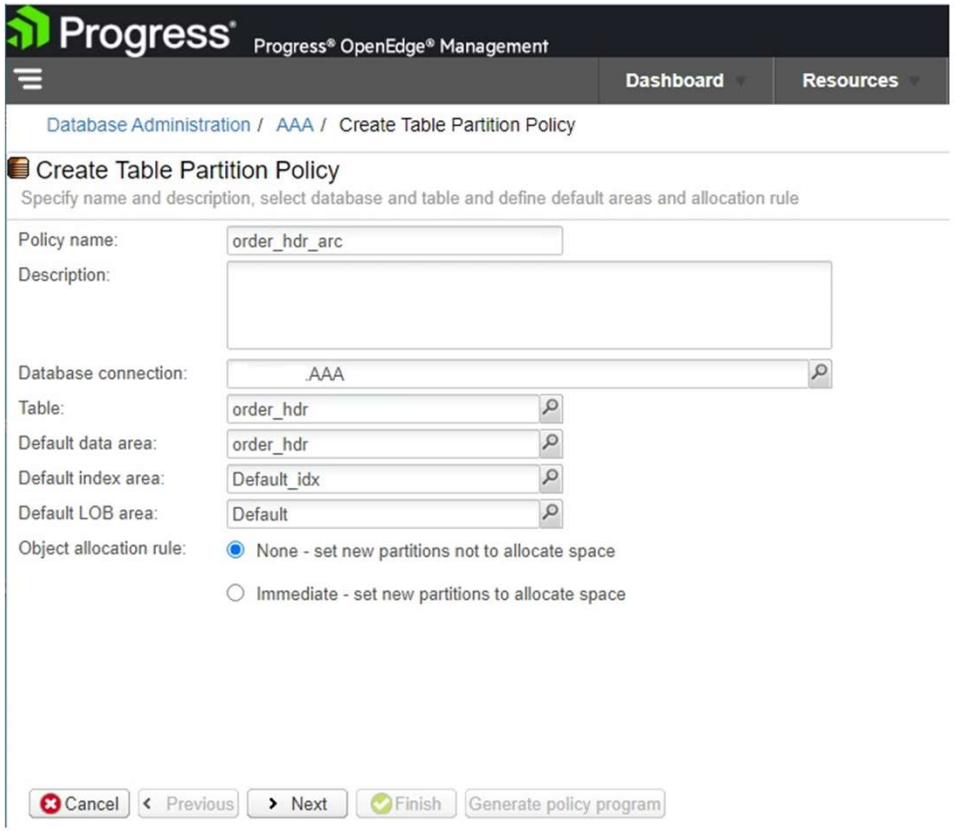
Verify:
proutil <DB> -C describe
Database Features

ID	Feature	Active	Details
5	Large Files	Yes	
9	64 Bit DH KEYS	Yes	
10	Large Keys	Yes	
11	64 Bit Sequences	Yes	
21	Table Partitioning	Yes	
23	New VST Tables	Yes	

V

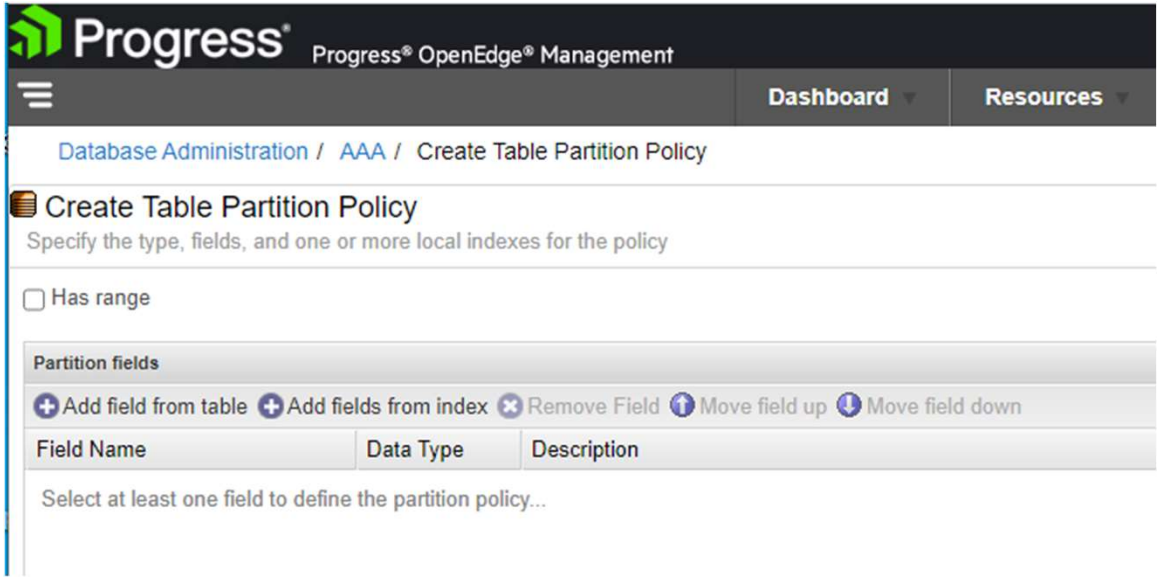
OMEGASERVE®
The last word in uptime protection!

CREATE PARTITION POLICY



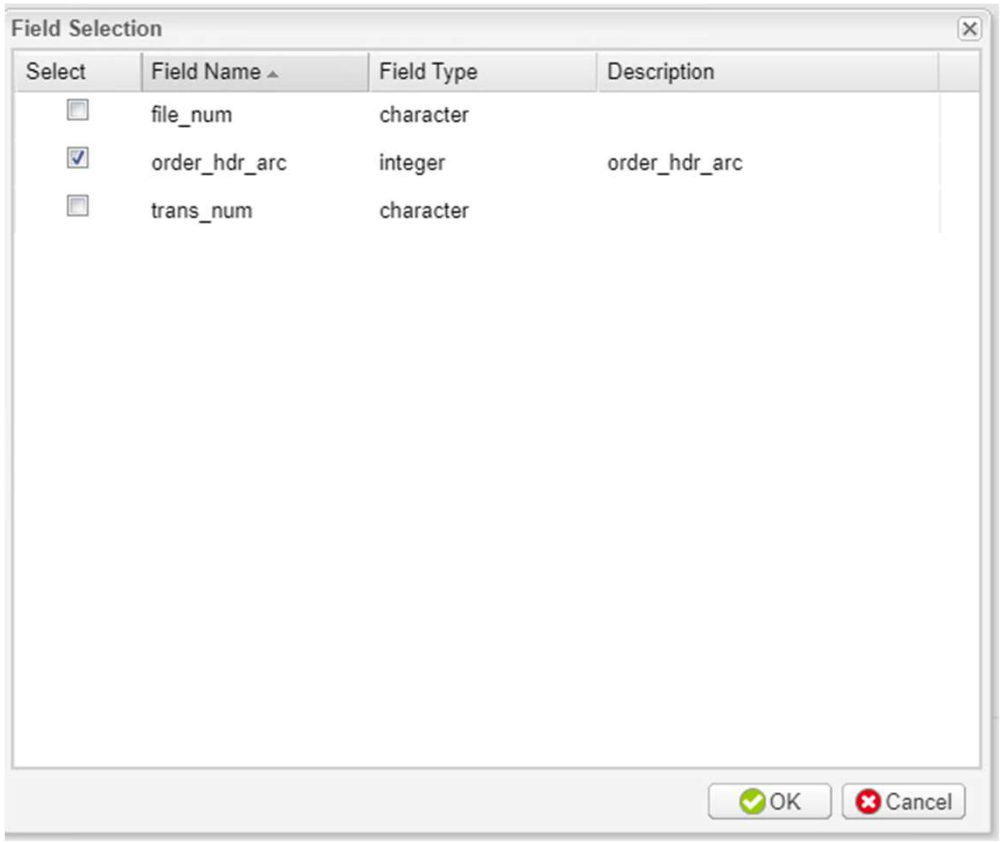
V

CREATE PARTITION POLICY



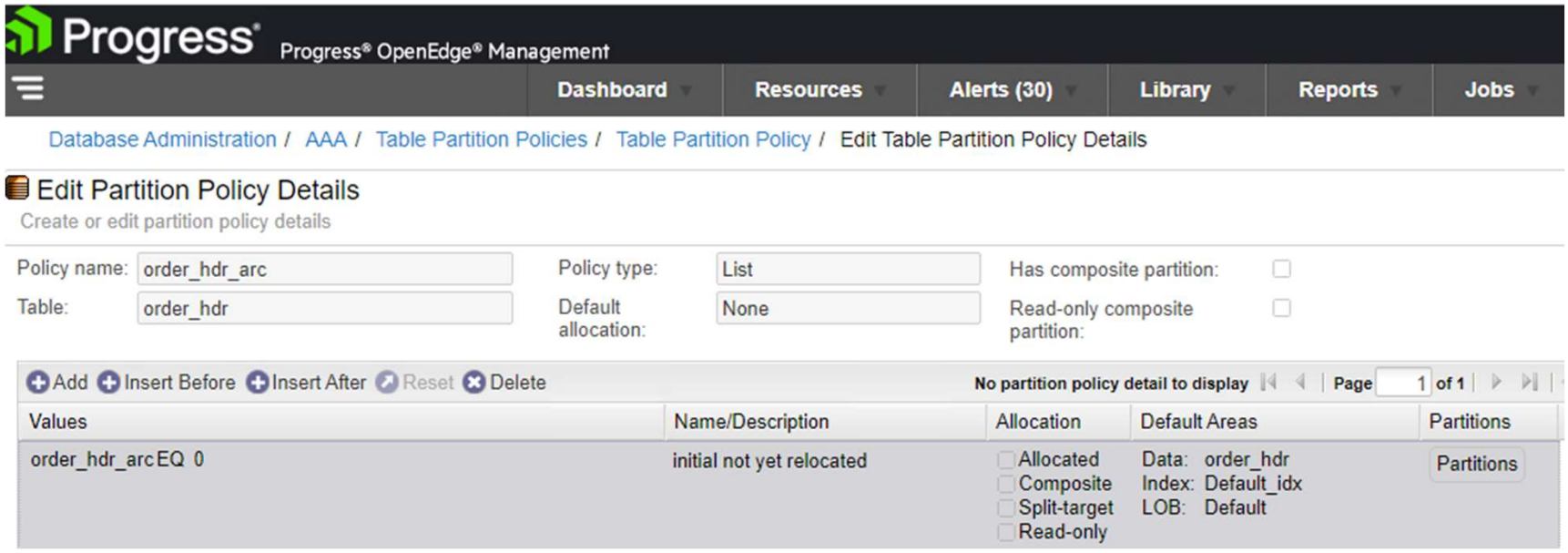
V

CREATE PARTITION POLICY



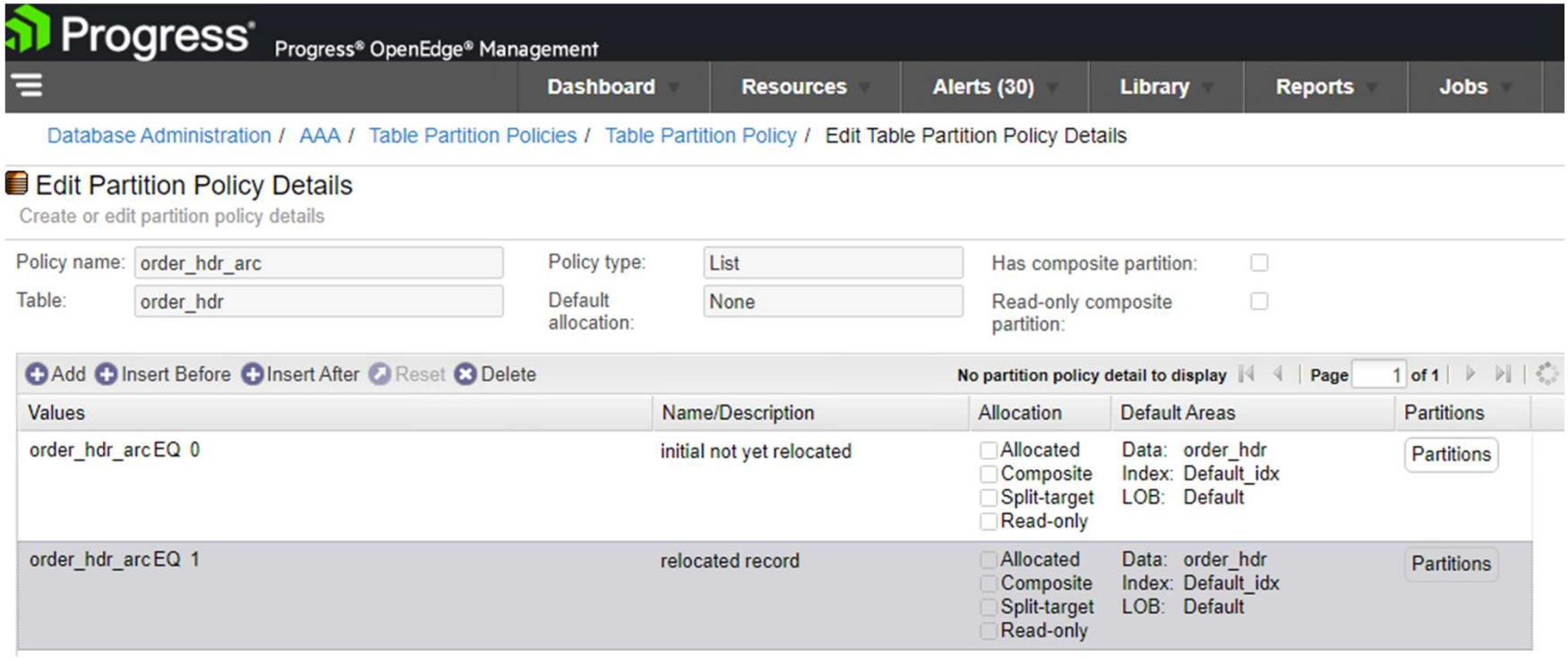
V

CREATE PARTITION POLICY DETAILS



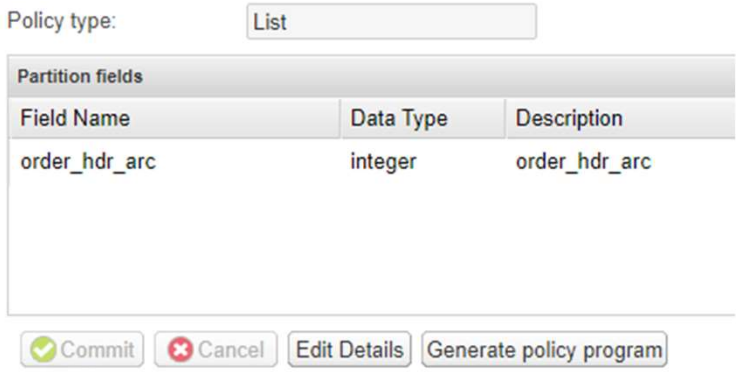
V

CREATE PARTITION POLICY DETAILS



V

CREATE PARTITION POLICY DETAILS




```
/* Instantiate a new Partition Policy */
partition Policy = service:New Partition Policy("order_hdr_arc").
assign
            partition Policy:Table = service:GetTable("order_hdr")
            partition Policy:HasRange = no
            partition Policy:Default DataArea =
service:GetArea("order_hdr")
            partition Policy:Default IndexArea =
service:GetArea("Default_idx")
            partition Policy:Default LibEria =
service:GetArea("Default")
            partition Policy:Default Allocation = "None"
            .
/* Add order_hdr_arc to the fields collection */
partition Policy:Fields:Add(partition
Policy:Table:Fields:Find("order_hdr_arc")).
/* Add order_hdr_arc_idx to the local index collection */
partition Policy:Indexes:Add(partition
Policy:Table:Indexes:Find("order_hdr_arc_idx")).

/* Create (commit) the partition Policy in the service */
service:Create Partition Policy(partition Policy).

catch e as Error:
            errorHandler = new DataAdminErrorHandler().
            errorHandler:Error(e).
end catch.
```

V

OMEGASERVE®
The last word in uptime protection!

START RELOCATION

```
----------------------------Error----------------------------
| **Unable to understand after – “TRIGGER” . (247)         |
|              ** Invalid statement. (254)                  |
| ** order_hdr_w.p Could not understand line 1. (198)       |
|      ** Unable to understand after – “END”. (247)         |
| ** order_hdr_w.p Could not understand line 4. (198)       |
|                                                           |
-------------------------------------------------------------

                                <OK>
```

```
disable triggers for load of order_hdr.
for each order_hdr:
order_hdr_arc = 1.
end.
```

*Can be throttled with pauses

V

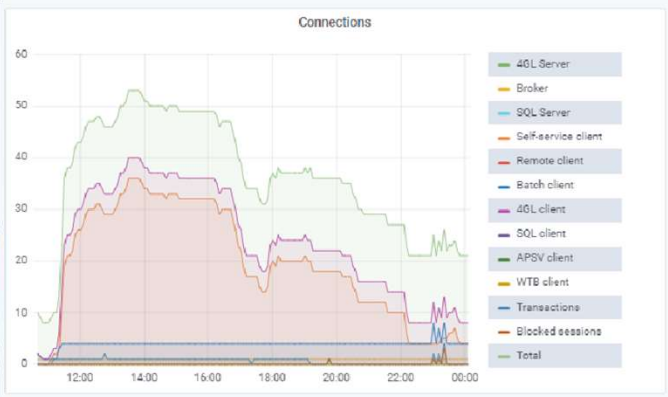
START RELOCATION

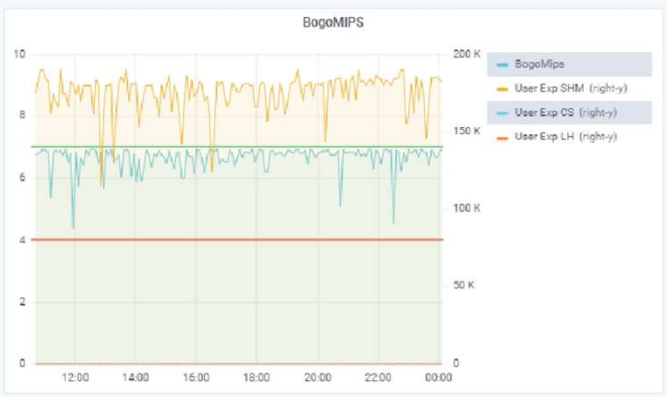
```
def var n as integer no-undo.
find first order_hdr .
mainloop:
   do while (true) transaction:
      do n = 1 to 10000:
         if not available (order_hdr) then leave mainloop.
         order_hdr.order_hdr_arc = 1.
         find next order_hdr.
         pause 10.
       end.
   end.
```

V

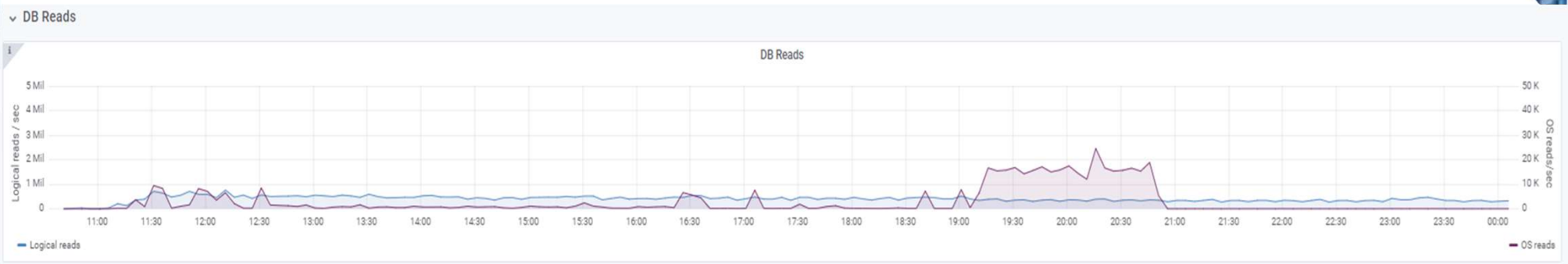
OMEGASERVE®
The last word in uptime protection!

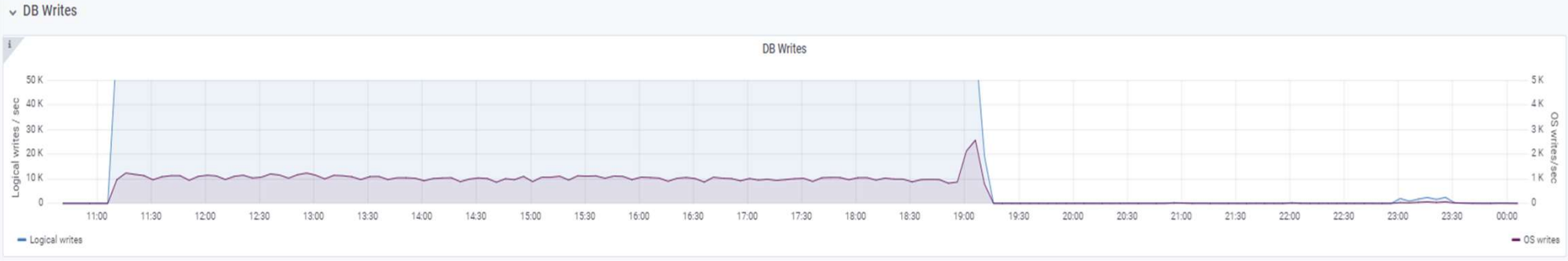
MONITORING IMPACT





MONITORING IMPACT

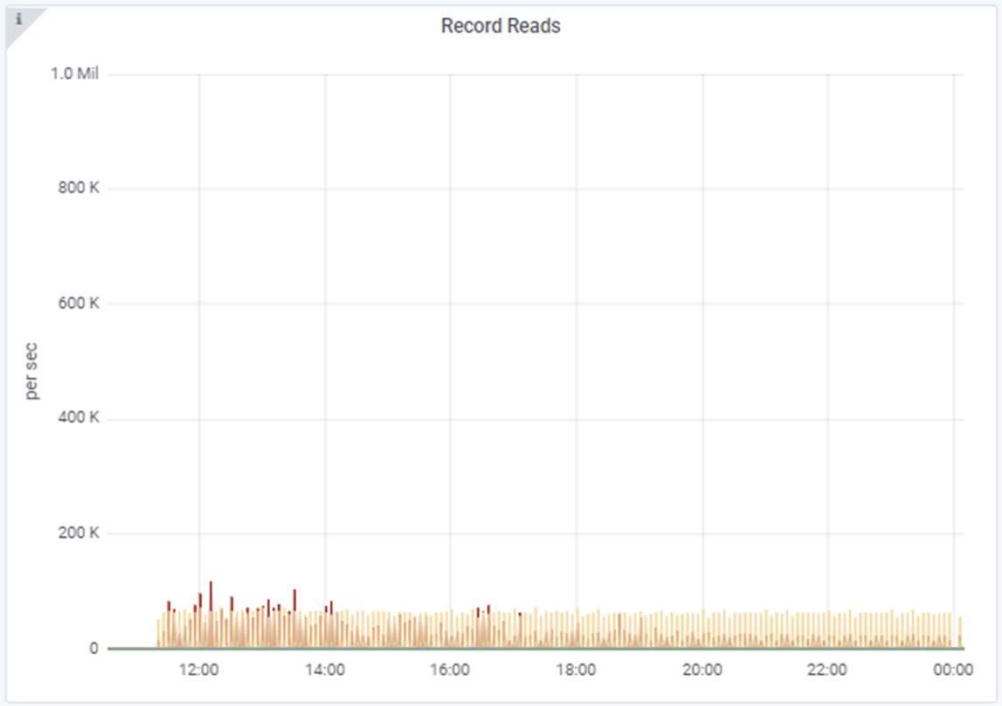




OMEGASERVE®
The last word in uptime protection!

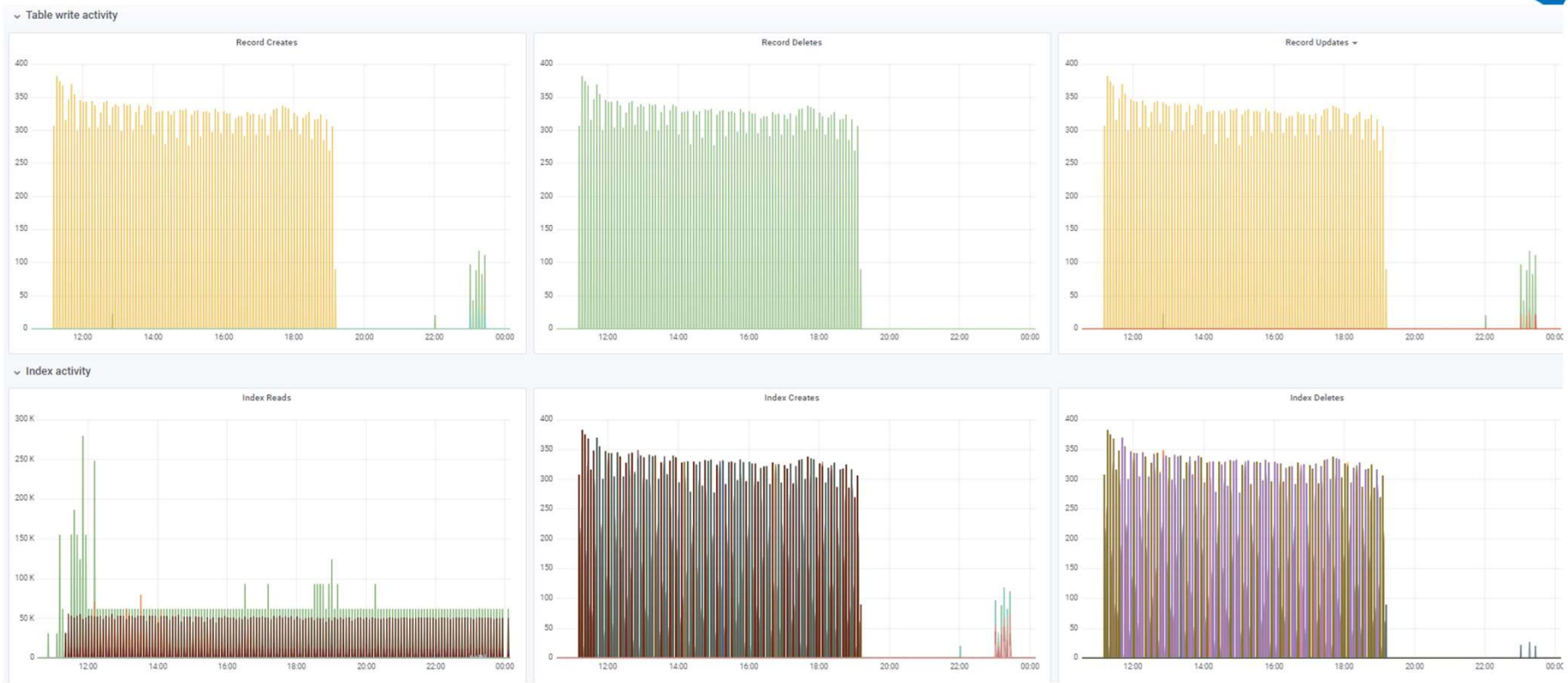
MONITORING IMPACT

Order_hdr reads (red) eclipsed by other table reads (orange)



MONITORING IMPACT

Records moved at a steady average 325 records/second



OTHER OBSERVATIONS

- No BI impact
- Significant AI writes: 231GB vs 199MB typical daily of AI files
- Completion time approximately 8h

```
PARTITION STATUS
----------------------
Table                                                    Rows
PUB.order_hdr
  initial:0                                                0
  order_hdr_archived:0                                     0
  order_hdr_archived:1                                     9313844
```

OMEGASERVE®
The last word in uptime protection!

POST RELOCATION RESULTS

Before:

Table	Records	-Record Size (B)- Size	Min	Max	Mean	---Fragments--- Count	Scatter Factor	Factor
PUB.order_hdr	9311997	10.1G	747	1598	1164	11503261	1.2	1.0

After:

Table	Records	-Record Size (B)- Size	Min	Max	Mean	---Fragments--- Count	Scatter Factor	Factor
PUB.cs_hdr.Initial(Composite)	0	0.0B	0	0	0	0	0.0	0.0
PUB.order_hdr.order_hdr_archived	9313844	10.1G	747	1598	1164	9313844	1.0	1.0

MEGASERVE®
The last word in uptime protection!

POST RELOCATION PERFORMANCE RESULTS

- Significant read improvement (x2 faster) on order_hdr:

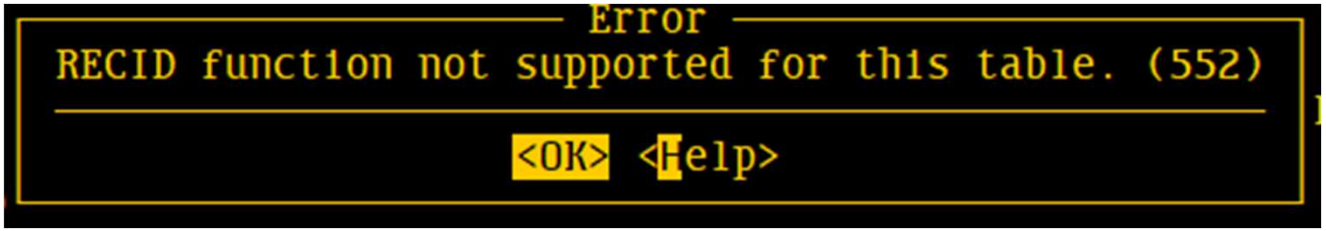
During work-hours (130 users) BEFORE relocation	80 sec
db online (but no users, access disabled)	50 sec
db online during relocation (30 users)	65 sec
During work-hours (130 users) AFTER relocation	40 sec

V

OMEGASERVE®
The last word in uptime protection!

MONDAY MORNING GOTCHA

- Some code still used RECIDs



V

TABLE PARTITION ROLLBACK

- No “quick” roll-back procedure
- Option A: Full dump/load (6hrs downtime)
- Option B: Dump table (2-3hrs downtime)
  - Shut down and back up DB
  - Dump order_hdr
  - Change initial policy to Split-Target
  - Truncate partitions
  - Drop table
  - Reload DF without partition properties
  - Reload order_hdr
  - Rebuild indexes for order_hdr
  - Re-enable AI and replication
  - Restart

OMEGASERVE®
The last word in uptime protection!

OE 12.3+

Use new REORG feature on-line

```
proutil AAA -C TABLEREORG order_hdr

<…>

Total records processed: 9522348. (20060)

Table reorganization operation reorg completed
successfully. (20043)
```

TOTAL TIME: 20 hours!

CONCLUSION

- Online reorg with table partitioning can work
  - BUT…
- Not a one-size fits all solution
- Environment should be reviewed to determine the best solution
- Test, test and TEST

OMEGASERVE®
The last word in uptime protection!

QUESTIONS?

PLEASE STOP BY AND VISIT OUR BOOTH!