Performance Tuning with DataStage Parallel Jobs

IBM Support Technical Exchange

Steve Wilkos July 2015



- Best practices for developing DataStage parallel jobs
- How to read the APT_DUMP_SCORE report
- Using NMON to analyze AIX and Linux performance
- Parallel job performance issues and how to resolve those issues
- References



Best Practices for Developing DataStage Parallel Jobs

Reduce processes generated at runtime

• Use a single-node configuration file (APT_CONFIG_FILE)



- Remove all partitioners and collectors (especially when using a single-node configuration file)
- Enable runtime column propagation on Copy stages with only one input and one output
- Minimize join structures (any stage with more than one input, such as Join, Lookup, Merge, Funnel)
- Minimize non-combinable stages such as Join, Aggregator, Remove Duplicates, Merge, Funnel, DB2 Enterprise, Oracle Enterprise, ODBC Enterprise, BuildOps, BufferOp
- Selectively (being careful to avoid deadlocks) disable buffering

Parallel datasets

- Use parallel datasets to land intermediate results between parallel jobs
- Parallel datasets retain data partitioning and sort order
- Datasets can only be read by other DataStage parallel jobs





Sorting

• Specify the **Don't Sort (Previously Sorted)** option for those key columns in the Sort stage if data has already been partitioned and sorted on a set of key columns

rff Sort_0 - Sort	
Stage Stage name: Sort_0 General Properties Advanced NLS Locale Image: Sorting Keys Image: Sorting Keys Image: Sort Key Node = Don't Sort (Previously Sorted) Image: Sort Key Node = Sort Image: Sort Key Node = Sort Image: Sort Key Mode = Sort Image: Sort Key Mode = Sort Image: Sort Key Mode = Sort Image: Sort Create Key Change Column = False Image: Sort Utility = DataStage Image: Stable Sort = True	 Sort Key Mode Don't Sort (Previously Sorted) Information: Type: List Don't Sort (Previously Sorted) means that input records are already sorted by this key. The Sort stage will then sort on secondary keys, if any. Don't Sort (Previously Grouped) means Available properties to add:
	OK Cancel Help

- When writing to parallel datasets sort order and partitioning are preserved.
- When reading from these datasets try to maintain this sorting, if possible, by using SAME partitioning
- Performance of individual sorts can be improved by increasing the memory usage per partition using the **Restrict Memory Usage (MB)** option of the standalone Sort stage.



Columns

- Remove unneeded columns as early as possible in the data flow
- When reading from database sources use a select list to read needed columns instead of the entire table
- When using runtime column propagation (RCP) it might be necessary to disable RCP for a
 particular stage to ensure that columns are actually removed using that stage's Output
 Mapping
- Specify a maximum length for Varchar columns
- Varchars without a maximum length (unbounded strings) can have a significant negative performance impact on a job flow

Stage	uential_File_16 - Sequ Input	ential F	ile					Columns View E	Data
JDSLi	nk14								
Gene	eral Properties Partitio	oning F	ormat Columns	Advanced					
	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description	
1	POLICY_NUMBER		BigInt		19		No		
2	POLICY_COVERAGE		VarChar	Unicode			No		
3	BIRTHDAY		Timestamp	Microsecond	26	6	Yes		
4	CALCULATE_AGE		BigInt				No		-
5	DATE_LOADED		Timestamp				No		-
6	LAST_CONTACT		Date		10		Yes		
7	FIRST_NAME		VarChar	Unicode	30		Yes		
8	LAST_NAME		VarChar	Unicode	75		Yes		
9	COUNTRY		VarChar		120		No		-
4	1		111						•
							s	Save Load.	
						ок		Cancel H	elp



- test DataStage parallel jobs with a parallel configuration file (\$APT_CONFIG_FILE) that has two or more nodes in its default pool
- more nodes does not always mean better job performance
- if you are seeing job startup time gradually increase usually that means that system resources are not available, which are critical to a job's performance
- reduce the number of transformers because transformers are individual shared object files
- a small fetch of 10-100 records could be faster if using lookup filesets rather than connecting to the database
- avoid using network drives for temporary directories
- avoid using GPFS on Linux for binaries and resource disks



How to read the APT_DUMP_SCORE report

The APT_DUMP_SCORE report

- The APT_DUMP_SCORE report records activity within the Information Server parallel engine
- Set the environment variable APT_DUMP_SCORE=TRUE
- In DataStage Administrator client click Parallel -> Reporting
- When a DataStage job is executed the data flow information in the compile job is combined with the APT_CONFIG_FILE configuration file to produce a detailed execution plan named the score
- The score is useful in analyzing job performance



The APT_DUMP_SCORE report

• The dump score contains two sections -- the data sets (DS) and the operators (OP).

Data sets - The data sets that are listed in the score are the same type of data sets that you create with the Data Set stage.

In this report they are temporary memory and/or disk storage during the job's run

Operators - Operators are individual parallel engine stages that you might see on the user interface



APT_CONFIG_FILE file

- The APT_CONFIG_FILE file specifies the nature and amount of parallelism along with the specific resource that are used to run a job
- Default location is /IBM/InformationServer/Server/Configurations

```
{
node "node1"
{
    fastname "development"
    pools ""
    resource disk "/u1/IS113/IBM/InformationServer/Server/Datasets" {pools ""}
    resource scratchdisk "/u1/IS85/IBM/InformationServer/Server/Scratch" {pools ""}
}
node "node2"
{
    fastname "development"
    pools ""
    resource disk "/u1/IS113/IBM/InformationServer/Server/Datasets" {pools ""}
    resource disk "/u1/IS113/IBM/InformationServer/Server/Datasets" {pools ""}
}
```

Sample DataStage parallel job



APT_DUMP_SCORE report dataset section

```
Message: main program: This step has 8 datasets:
ds0: {op0[1p] (sequential IN)
   eAny<>eCollectAny
   op1[2p] (parallel Checksum 1)}
ds1: {op1[2p] (parallel Checksum 1)
   eOther(APT HashPartitioner { key={ value=SK }
})#>eCollectAnv
   op3[2p] (parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(0) in Join 2)}
ds2: {op2[1p] (sequential IN DS)
   eOther(APT HashPartitioner { key={ value=SK }
})<>eCollectAny
   op4[2p] (parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(1) in Join 2)}
ds3: {op3[2p] (parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(0) in Join 2)
   [pp] eSame=>eCollectAny
   op5[2p] (parallel APT JoinSubOperatorNC in Join 2)}
ds4: {op4[2p] (parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(1) in Join 2)
   [pp] eSame=>eCollectAny
   op5[2p] (parallel APT JoinSubOperatorNC in Join 2)}
ds5: {op5[2p] (parallel APT JoinSubOperatorNC in Join 2)
   eAny=>eCollectAny
   op6[2p] (parallel APT TransformOperatorImpIV0S3 FullOuter Join Sample Transformer 3 in
Transformer 3)
ds6: {op6[2p] (parallel APT TransformOperatorImpIV0S3 FullOuter Join Sample Transformer 3 in
Transformer 3)
   >>eCollectAnv
   op7[1p] (sequential APT RealFileExportOperator1 in OUT DS)}
ds7: {op6[2p] (parallel APT TransformOperatorImpIV0S3 FullOuter Join Sample Transformer 3 in
Transformer 3)
   >>eCollectAny
   op8[1p] (sequential APT RealFileExportOperator1 in OUT)}
```

Data sets structure:

```
ds0: {op0[1p] (sequential IN)
eAny<>eCollectAny
op1[2p] (parallel Checksum_1)}
ds1: {op1[2p] (parallel Checksum_1)
eOther(APT_HashPartitioner { key={ value=SK }
})#>eCollectAny
op3[2p] (parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(0) in
Join_2)}
```

- The name of the dataset is ds0. Within the curly brackets there are three stages:
 - -- the source of the data set is operator 0, sequential IN
 - -- the activity of the data set is operator 1, parallel Checksum_1





Data Set symbols

- The symbols between the originating partitioning method and the target read method translates to the parallelism of the partitioning.
- Here is the list of the symbols and their definition:
 - -> Sequential to Sequential
 <> Sequential to Parallel
 => Parallel to Parallel (*SAME*)
 #> Parallel to Parallel (*NOT SAME*)
 >> Parallel to Sequential
 > No Source or Target
- eOther is the originating or input method
- eCollectAny is the target read method

```
ds0: {op0[1p] (sequential IN)
eAny<>eCollectAny
op1[2p] (parallel Checksum_1)}
ds1: {op1[2p] (parallel Checksum_1)
eOther(APT_HashPartitioner { key={ value=SK }
})#>eCollectAny
op3[2p] (parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(0) in Join_2)}
```

APT_DUMP_SCORE report operator section

```
It has 9 operators:
 op0[1p] {(sequential IN)
   on nodes (
    node1[op0,p0]
   )}
 op1[2p] {(parallel Checksum 1)
   on nodes (
    node1[op1.p0]
    node2[op1,p1]
   )}
 op2[1p] {(sequential IN_DS)
   on nodes (
    node1[op2,p0]
   )}
 op3[2p] {(parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(0) in Join 2)
   on nodes (
    node1[op3,p0]
    node2[op3,p1]
   )}
 op4[2p] {(parallel inserted tsort operator {key={value=SK, subArgs={asc}}}(1) in Join 2)
   on nodes (
    node1[op4,p0]
    node2[op4,p1]
   )}
 op5[2p] {(parallel APT JoinSubOperatorNC in Join 2)
   on nodes (
    node1[op5,p0]
    node2[op5,p1]
   )}
 op6[2p] {(parallel APT TransformOperatorImpIV0S3 FullOuter Join Sample Transformer 3 in Transformer 3)
   on nodes (
    node1[op6,p0]
    node2[op6,p1]
   )}
 op7[1p] {(sequential APT RealFileExportOperator1 in OUT DS)
   on nodes (
    node2[op7,p0]
   )}
 op8[1p] {(sequential APT RealFileExportOperator1 in OUT)
   on nodes (
    node2[op8,p0]
17 )}
```

[[mi]]

Operators structure

```
op0[1p] {(sequential IN)
on nodes (
node1[op0,p0]
)}
op1[2p] {(parallel Checksum_1)
on nodes (
node1[op1,p0]
node2[op1,p1]
)}
```

- The two operators are op0 and op1
- op0[1p] indicates that one partition is provided to the operator by the engine
- op1[2p] indicates that two partitions are provided to the operator by the engine
- (sequential IN) indicates execution mode sequential and stage name IN
- (parallel Checksum_1) indicates execution mode parallel and stage name Checksum_1
- op0 runs on node 1
- op1 runs are node 1 and node 2

Buffer operators

• Buffer operators are used when the downstream operator is at risk of getting overloaded with data while it is processing

op6[4p] {(parallel buffer(0)) on nodes (node1[op6,p0] node2[op6,p1]

• Buffer operators are an attempt to produce a buffer zone where two things happen:

1) The buffer operator communicates with the upstream operator to manage the sending of data

2) The buffer operator holds on to the data until the downstream operator is ready for the next block of data

- If your parallel job is running slowly look for the number of buffer operators in the report
- Buffer operators prevent race conditions between operators
- Best practices job design can reduce the amount of buffering that occurs



The APT_DUMP_SCORE report

To improve parallel job performance:

- If your parallel job is running slowly look for the number of buffer operators in the report
- Examine the report for sort operators because sorting can be detrimental to parallel job performance
- Join stages require data to be sorted so the engine inserts a tsort operator
- Data sets use memory which can impact job performance
- Ensure that your intended design is correctly implemented by the parallel engine



Using NMON to analyze AIX and Linux performance

nmon

You can download the nmon analyzer here: (latest version is 4.6 nmon_analyser_v46.zip)

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/ /Power%20Systems/page/nmon_analyser

- nmon creates graphs to aid in analyzing performance issues
- nmon works with Microsoft Excel 2003 and later on AIX and Linux
- nmon is a free tool that is not supported by IBM



click Analyze nmon data

х		5. 6.	Ŧ						nmon analys	er v46.xlsm - Excel								?	_	5 ×
F	ILE	HOME	INSERT	PAGE L	AYOUT FORMUL	LAS DATA	REVIEW	VIEW												Sign in
	b 3	🖌 Cut		iol	× 10 × 0 •	=	»	Wrap Toxt	Conoral	. 8			F		∑ AutoSur	n • A	n in	0	0	
	L 🛙	🖹 Copy 👻	A	Idi	A A	= =	of the	wiap iext	General		ŧ s				🕹 Fill 👻	Z				
Pa	ste	🖉 Format Pain	nter B	<u>Ι</u> <u>υ</u> .	- <u>A</u> - <u>A</u> -		€ → →	Merge & Cente	r • \$ • % •	€.0 .00 Cond	tional Format as	Cell	Insert Delet	e Format	📌 Clear 🕶	Sort &	Find &	Share W	/ebEx	
	- -	liphoard		c.	iont 5		Alianmont		5 Numbe	r 5	stude *	Styles *	Colle	Ť		Filter *	Select *	Mobe	,	~
							Alighthetic		- Numbe		Styles		Cells			cuting		WEDD		
		А			В			D	F	F	G	н	1		К		М		J	(🔺
1	V4.6	6	3	30-Jun-15	0		-				0									
2																				
3																				
4									Anal	/ze nmon data										
5										20 1111011 data										
6																				
2																				
9	Out	but																		_
10	GR/	APHS			ALL	CHA	RTS	Specify ou	tput option for a	enerated graphs	(see user quide)								
11	INT	ERVALS			1	999	999	FIRST and	LAST time inter	als to process	Ĵ	,								
12	TIM	ES						FIRST and	LAST time value	s to process (no	te - overrides IN	ITERVAL	S settings) -	see Use	r Guide for	details				
13	ME	RGE	1		NO	NOT	FOP	Specify YE	ES to merge the i	nput files and TO	P if you want to	o merge th	ne TOP and	UARG se	ections					
14	PIV	OT			NO			Specify YE	S to generate a	Pivot chart using	the parameters	s on the S	ettings shee	et						
15	SCA	ATTER			NO			Indicates if	Scatter charts s	hall be included.	Applies to TOF	P CPU% I	by PID char	tonly. I	his option is	to avoid Ex	cel 2013	crash.	£-11.16 -	. In m
10	BIG				YES			Set to NO t	on your data has	> 1048576 rows not ESS analysis	s or large lines () Set this i		improve peri	ormance bi	u the Ana	liyzer will	iaii ii a	larç
18	FILE	5 FLIST			TEO			Name of fil	e containing a lis	t of nmon files to	be processed	(blank = c	(noleit							
19								Nume of hi	e containing a lis	it of fillion files to	be processed		nulog)				3			
20																	-			
21																				
22																		_		
23	-																			
24	-							2							8		2		-	
20																				
20																			-	
28																				
29																				
30																				
31																				
32																				
	• •	Analy	/ser	Settings	Release Notes	+					•									Þ
REA	ωDY														E		.		+	100%
				Y																

🗱 🔒 🕤 e e = aix129snmonJulv7.nmon.xlsx - Excel CHART TOOLS ? 🗹 🗕 🗗 🗙 HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DESIGN FORMAT Sign in fx Disk total KB/s aix1ainf29s A1 • : × 1 × В С D Е G Н Ν U 🔺 A F 0 P Q Μ R S 15:01:23 Last 17:14:56 1 Samples 802 First 2 System Summary aix1ainf29s 7/7/2015 3 4 -PhysicalCPU -IO/sec 5 6 3.5 160 7 8 140 3.0 9 10 120 11 2.5 12 13 100 14 2.0 #CPUs Disk xfer 15 80 16 1.5 17 18 60 19 1.0 20 40 21 22 0.5 20 23 24 25 0.0 15:01 15:03 15:05 15:07 17:03 17:05 17:07 17:09 17:11 5:09 m 19 5:23 ŝ 49 5 6:05 6:09 :49 6:53 6:55 L6:59 7:01 7:13 26 6:01 :03 6:07 6:51 L6:57 ŝ 6.0 27 28 **Disk tps statistics** VP_CPU: User% Wait% Idle% CPU% PhysCPU Virtual CP Other LPA Pool CPUs Entitled Weight Sys% 0.0 0.6 29 2.2 192.0 Avg disk tps during an interval: 3 Avg 1.7 2.4 6.3 10.0 40.0 11.0 2.5 30 Max disk tps during an interval: 143 Max 18.0 8.1 0.0 12.4 32.9 3.3 10.0 40.0 11.0 2.5 192.0 31 17:03:36 0.0 5.2 5.3 5.3 Max disk tps interval time: Max:Avg 8.2 4.8 1.0 32 0 Total number of Mbytes read: 33 Total number of Mbytes written: 129 34 Read/Write Ratio: 0.0 SYS_SUMM AAA DISK_SUMM ... (+) BBBB BBBC BBBD BBBF BBBL BBBN BBBP BBBV 4 . 4 • READY Ħ . . . 100% +

The excel file that is created

CPU utilization



Analyze CPU% by command

٨	1		ш		T		1		V		T.				M				N			0		D		0		D		c		т		TT.		V		14/	
me	Size	3	⊓ ResTex	t B	1 LesDat	a) CharlO	9	K KRAM	F	L Paging	,	Com	mand	IVI			1	NI Ma	l class	Inte	U rvalC	PWS	P et	Us	er	Δ	к rg		2				0	_	V		VV	
ine	1	1272	(0510)	0	CODU	0	charlo	0		0	aBing	0	com	mana						ciuss	inte	i vuic	1 110		0.0		- 1	.9											
)																			
													CP	U%	by	co	mm	nan	d a	ix1	Lair	nf2	9s	7/7	7/2	015	5												
																		Avg.	W	Avg.	■M	ax.																	
	-																																						
	-																																						
	- ⁹⁰																																						
	30																																						
	-																																						
	70 -																																						
	-																																						
	50																																						
	+0																																						
	- 30																																						
	20																																						
	.0																																						
	0 +			<u> </u>		1		-				-								-	1	_												_			-		-
		lient aso	tbin	amf	atch	ener	vagt mcd	fmd	agnt	agsr	smo	yptd	etty	dl/d	init init	n64	java	biod	nom	14cb	etm	proc	aix	osh	lego	p2d	mob .	dpgc	racr	mcd	ockd	cmd	vreg	sshd	on de	1slp	tbin	mom	
		adc	audi	0	dcw	liste	ipro Ib2f	db2	disc	discu	Р	xcr	00	= TC		bmc		×	Daei	-	⊆	اد	stop		ġ	0	ran	2 2	Ŭ	-	pc.ld	ť	s	-, - E	SVSE	tier	gent	s_n	2
					õ.	CIII	cin			-		ucp_		Bil					ack			ssyn	non								-		sl				lma	opa	
	_											en		+					Callb			h															t,	-	
	-														<u>.</u>				dicC																				
	-																		erio																				
	-																		lpxP																				
	-																		2																				
	SCPU)4 9	CPU	ALL	TOP	I	JARG	77	77	CPU	ALL	C	PU01	C	2002	CF	2003	C	PU04		Ð			4															_
						_					_					1					U											F	н (г	តា 🔳	п				
																																H	H L	9 C					

Memory usage

	▼ : []	XV	f _x																												
A	B N Drococc ⁹	C	-0/ Cu	D	E	D:	F	(5 v	Н		Ι		J	K		L	М	_	Ν		0	P		Q		R		S	T	
nory	N Process70	rscaci	1e70 Sy	stern70	riee%	PI	meu%	User	/0	N	/lem	۱or۱	/ Use	e aix	(1air	nf29s	7/7	/201	15												
													Sustand	0/ =1	Drococci	и <u>–</u> гс	•														
00												. 1	System	70	FIUCESS	70 <mark> </mark>	Laciie70														
90 -																															
80 -																															
70 -																															
60 -																															
50 -																															
40 -																															
30 -																															
20 -																															
10 -																															
0																															
:01	03	110	116	:19	25	:28	32 34 36	38	:41	:45	50	24	00 228	:01	:05	11	116	22	23	31	33	38	:40	:45	:49	1 22 4	292	:00	:04	00: 11	
15	51 5	មមដ	មមម	51 51 54 51 51 52 51 52 52 52 51 52 52 52 51 52 52 52 52 52 51 52 52 52 52 52 52 52 52 52 52 52 52 525	1 2 2	5 5	ងងង	5 5	2 2	5 5 F	151	1 11 H	1 1 1	16 16	16 16	16 16	16 16 16	16	101	16 16	16	16	16	16	16 16	161	16	11	111	111	Ę
	IOADAPT	JFSF	ILE	JFSINO	DE I	MEM	MEM	NEW	ME	MUSE	NE	T N	IETPAC	KET	NETS	IZE I	(+)	E .	-												
										_															Ⅲ	₿			 I		+

Network I/O



Disk total KB/s



How the nmon report can help you improve job performance

• % CPU is high

Acton: identify which processes are consuming large amounts of CPU

• Disk % Busy is high

Action: cache more data in-memory

• Network I/O is high

Action: increase network bandwidth

• Physical memory usage is high

Action: identify which processes are using memory



Parallel job performance issues and how to resolve those issues

Recording performance data at design time

- 1. Open a job in the Designer client

- Click Edit > Job Properties
 Click the Execution page
 Select the Record job performance data check box

Jobs\RowGenToPeek - Job Properties	
General Parameters Job control Dependencies NLS Execution Defaults	
Compile in Trace Mode Trace Options Force Sequential Mode Limits Per Partition Number of Records Per Link: 100 Advanced runtime options:	
 Include project setting ✓ Record job performance data 	
	OK Cancel <u>H</u> elp



Recording performance data at run time

- 1. Open a job in the Designer client, or select a job in the display area of the Director client
- 2. Click the Run button on the toolbar to open the Job Run Options window
- 3. Click the General page
- 4. Select the Record job performance data check box

RowGenToPeek - Job Run Options	
Limits General	
 Generate <u>operational metadata</u> Disable project-level message handling Disable compiled-in job-level message handling Record job performance <u>data</u> 	
Workload management	
Project default (MediumPriorityJobs)	
Run <u>V</u> alidate Cancel	<u>H</u> elp



Viewing performance data

- Open the Performance Analysis window by using one of the following methods:
 - In the Designer client, click File > Performance Analysis
 - In the Director client, click Job > Analyze Performance
 - In either client, click the Performance Analysis toolbar button

9 	I C → PicturoLong Paint
	* [Parallel - Test]
	IBM® InfoSphere® DataStage® and QualityStage Designer
	File Edit View Repository Diagram Debug Import Export Tools Window Help
	Parallel
÷	□ □ 1500 × 920px □ Size: 125.3KB 200% □ □ •



Viewing performance data



Environment variables

• APT_CONFIG_FILE

File specifies the nature and amount of parallelism along with the specific resources that are used to run a job

• APT_SCORE_DUMP

Report records activity within the Information Server parallel engine

• APT_PM_PLAYER_TIMING

Identify which stages in a job are consuming large amounts of CPU time and how much data the stage is processing

• APT_NO_SORT_INSERTION

Prevents the automatic insertion of sort components in your job to optimize the performance of the operators in your data flow

• APT_BUFFER_MAXIMUM_MEMORY

Sets the default value of Maximum memory buffer size.



Parallel job performance issue:

- On RedHat Enterprise Linux 6.1 operating system parallel jobs run with degradated performance
- On RedHat Linux 6.1 system there is a 10 90% performance degradation depending on the hardware configuration and DataStage node configuration during parallel job runs. An increase in the number of CPU cores and DataStage node counts used on the system leads to more performance degradation
- This is caused by parallel jobs using transformer stages and use of the CurrentTimestampMS() function in the transformer stage on the Redhat Linux 6.1 system
- On the computer that has the Redhat Enterprise Linux 6.1 operating system installed, set the environment variable as follows:

TZ=:/etc/localtime

before running parallel jobs. The CPU time improves

www.ibm.com/support/docview.wss?uid=swg21598208



Parallel job performance issue:

- DataStage parallel job has slow performance sending data to remote compute nodes
- When a DataStage job is setup with nodes running on multiple physical machines the transmission of data for large VarChar columns to remote nodes may be very slow compared to sending the same data to local nodes
- DataStage can transfer data faster to local nodes versus remote nodes
- When sending VarChar data to remote nodes if a length for the data has been declared, then space is allocated for the maximum size and the full bound size is transmitted to the remote node.
- However, if you have unbound data size (length of VarChar is undefined), then the size of data sent to the remote system is the actual variable size of each record
- The difference will not usually be that large unless the max size of VarChar column is much larger than the average size.
- www.ibm.com/support/docview.wss?uid=swg21667407



Parallel job performance issue:

- After upgrading to RedHat Linux 6 parallel jobs are using more memory
- The increased memory usage is caused by the Linux kernel 'Huge Page' feature
- This feature has been in Linux for years but normally was not enabled
- Starting with Red Hat Enterprise Linux 6 the Huge Page feature is enabled by default ("Transparent Huge Pages")
- Users will experience 30% or more increase in memory usage during parallel job execution
- Resolution is to turn off Huge Pages
- Documented in technote:

http://www.ibm.com/support/docview.wss?uid=swg21664196



Best practices for developing DataStage parallel jobs

Developing DataStage and QualityStage parallel jobs http://www.ibm.com/support/knowledgecenter/SSZJPZ 11.3.0/com.ibm.swg.im.iis.ds.nav.doc /topics/dshold_developing_parallel_ds_and_gs_jobs.html

Designing parallel jobs

http://www.ibm.com/support/knowledgecenter/SSZJPZ 11.3.0/com.ibm.swg.im.iis.ds.parjob.d ev.doc/topics/g deeref Parallel Jobs General Information.html?lang=en

InfoSphere DataStage Parallel Framework Standard Practices http://www.redbooks.ibm.com/redbooks/pdfs/sg247830.pdf



How to read the APT_DUMP_SCORE report

InfoSphere DataStage parallel jobs: Understanding the content of the APT_DUMP_SCORE report

http://www.ibm.com/support/docview.wss?uid=swg21595704

Using NMON to analyze AIX and Linux performance

nmon performance: A free tool to analyze AIX and Linux performance <u>http://www.ibm.com/developerworks/aix/library/au-analyze_aix/</u>



Common parallel job performance issues and how to resolve those issues

Running parallel jobs in debug mode <u>http://www.ibm.com/support/knowledgecenter/SSZJPZ 11.3.0/com.ibm.swg.im.iis.ds.parjob.d</u> <u>ev.doc/topics/debuggingparalleljobsusingthedebugger.html?lang=en</u>

Debugging parallel jobs with the debugging stages http://www.ibm.com/support/knowledgecenter/SSZJPZ 11.3.0/com.ibm.swg.im.iis.ds.parjob.d ev.doc/topics/debuggingyoudesigns.html?lang=en

How to turn on Tracing for DataStage Parallel Job http://www.ibm.com/support/docview.wss?uid=swg21441558

InfoSphere DataStage: Parallel Job Performance Issue on Redhat Linux 6.1 System http://www.ibm.com/support/docview.wss?uid=swg21598208

DataStage parallel job job has slow performance sending data to remote compute nodes http://www.ibm.com/support/docview.wss?uid=swg21667407



Thank-you!

