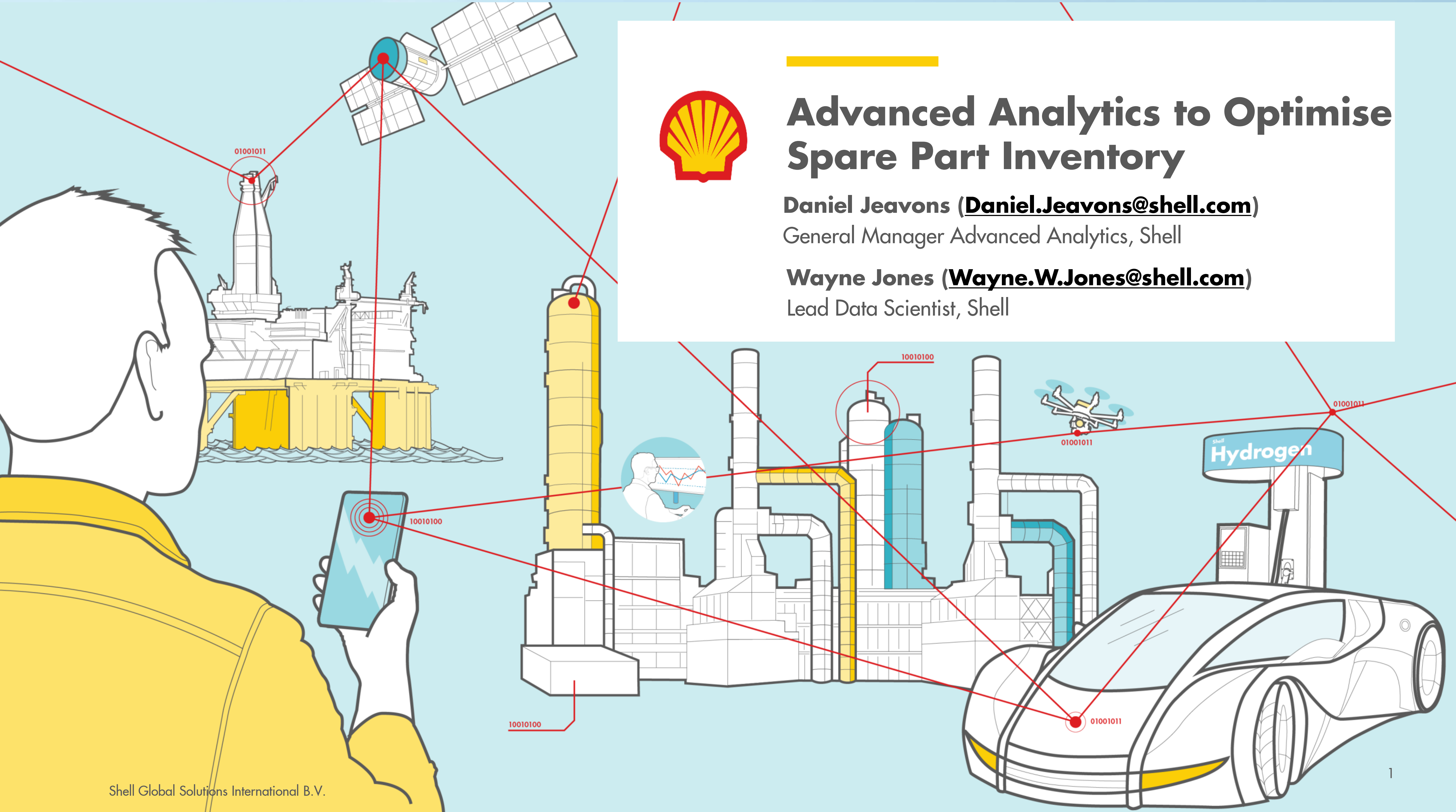


Advanced Analytics to Optimise Spare Part Inventory

Daniel Jeavons (**Daniel.Jeavons@shell.com**)
General Manager Advanced Analytics, Shell

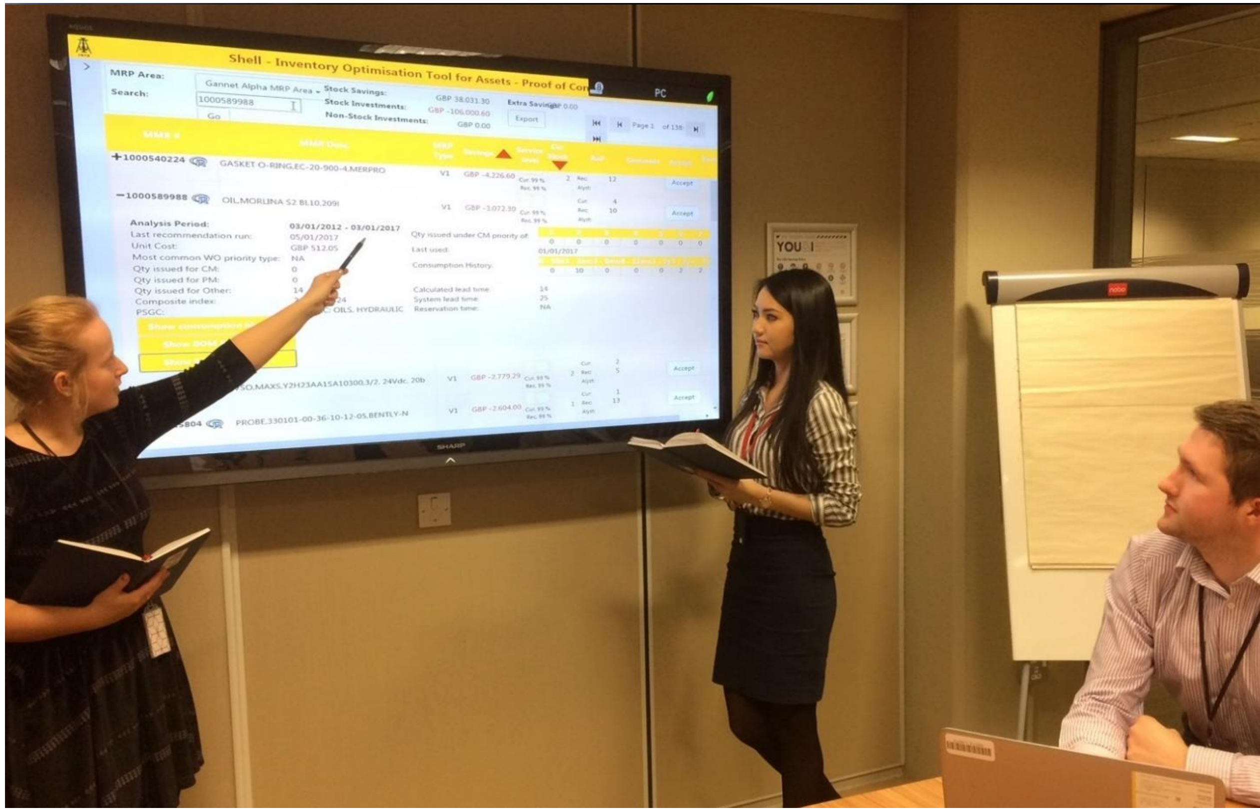
Wayne Jones (**Wayne.W.Jones@shell.com**)
Lead Data Scientist, Shell

The challenge of materials management

- Inventories were managed in many different ways
- Stock levels were set on vendors' recommendation or using local optimisation schemes
- Excessive risk aversion led to rising stock levels and working capital being tied up in stores
- Excessive cost trimming led to inventories that were too small and unacceptably high risks of production deferment
- There was a need to ensure that the right parts are available in the right place at the right time

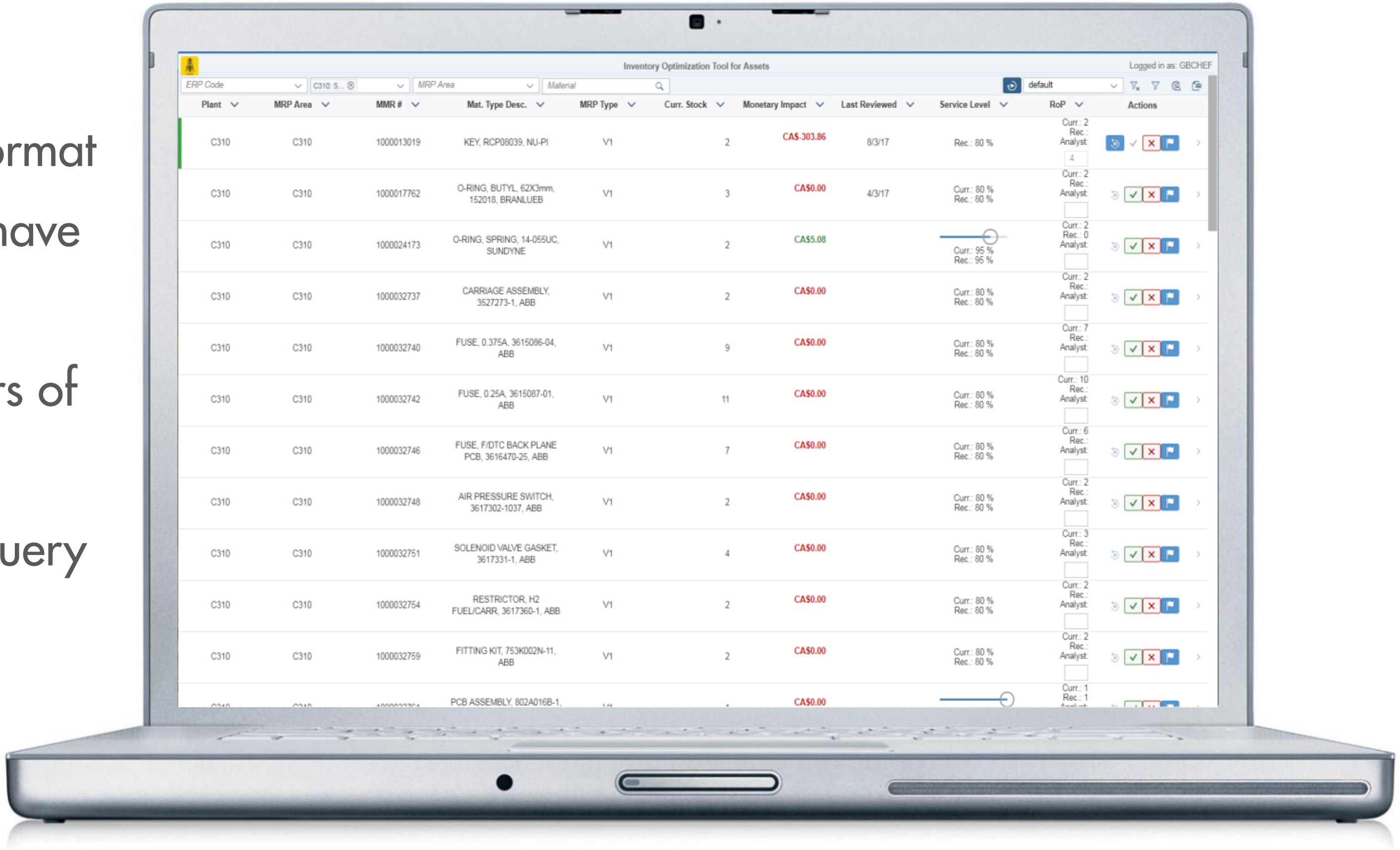
From reactive to proactive

- Co-creation through an agile team
- Combine datasets from the enterprise resource planning system to understand the problem
- Understand the input parameters to determine variability
- Assess the criticality of equipment



Creating user confidence

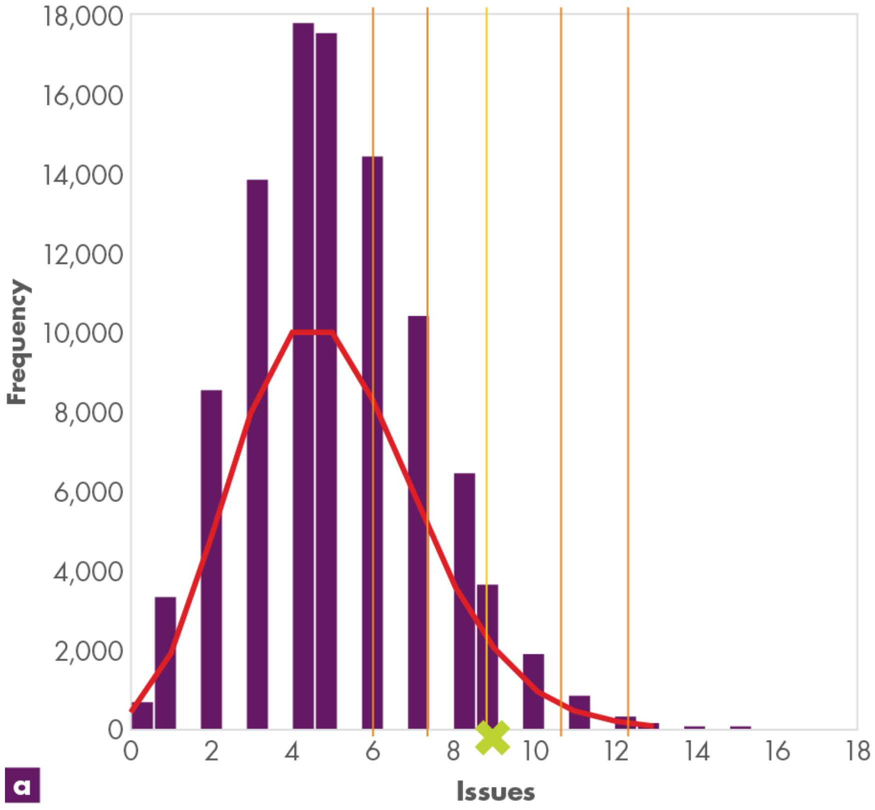
- Managing change is key
- Presenting recommendations in a user-friendly format
- Back testing to simulate what stock levels would have been for the past five years
- Building confidence and clarifying the parameters of the algorithm
- Creating a portal for users to interact with and query recommendations



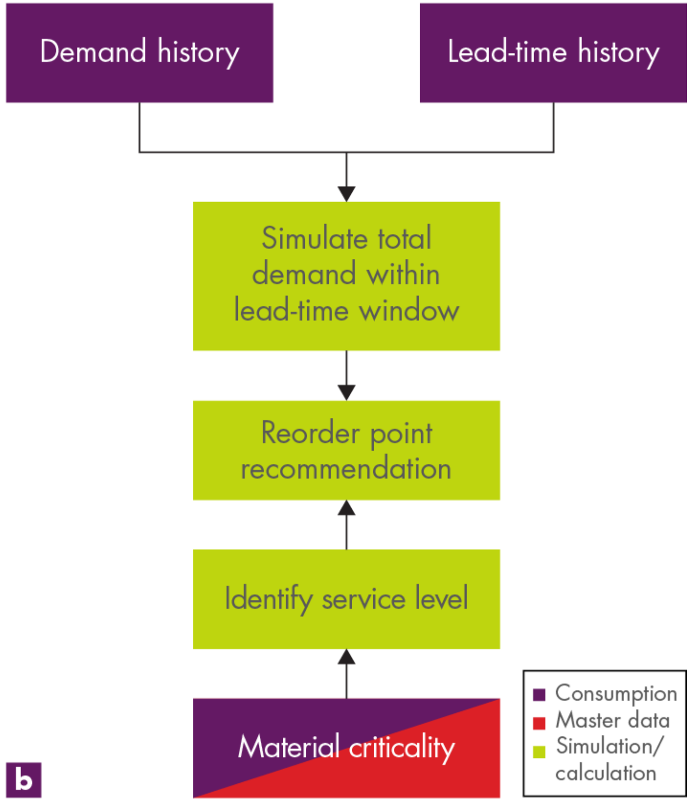
Developing the algorithm

- Criticality score to set required service level
- Minimize risk for critical items
- Simulate consumption and use the bootstrapping algorithm to meet required service level
- Innovative simulation process developed to deal with "lumpy" data

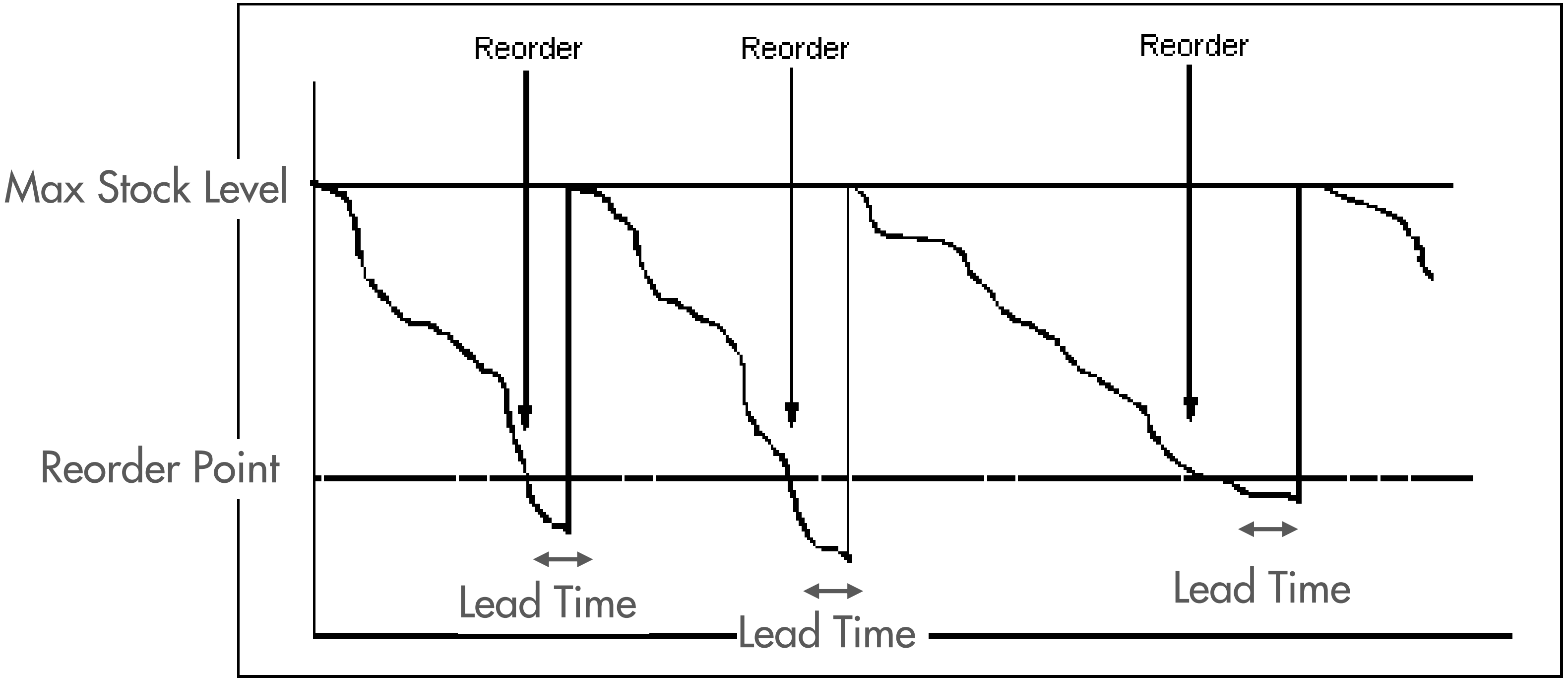
Bootstrap simulation



Inputs for reorder recommendation levels



Developing the algorithm



Pilot project: Gulf of Mexico

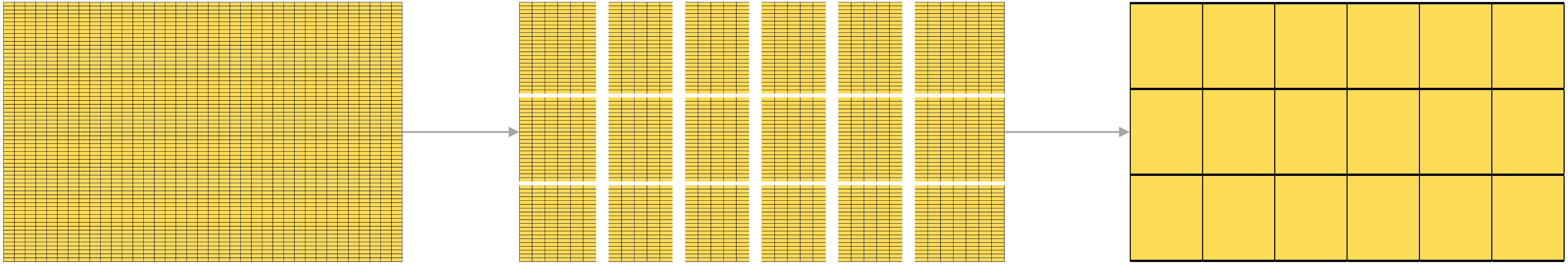
- Pilot project at Ursa field paid for itself in four weeks
- Review cost to carry
- Direct savings from cancelled purchase orders
- Millions of dollars in tracked benefits in 2016 trial
- Estimate of benefits conservative: reduced downtime not included

Scaling into a global tool

- Accessible architecture
- Running 10,000 simulations per inventory item across all materials took 48 hours
- Leveraged Spark capability to scale algorithms linearly
- Reduced simulation time from 48 hours to less than an hour
- With SAP HANA, users can get real-time answers without pre-processing

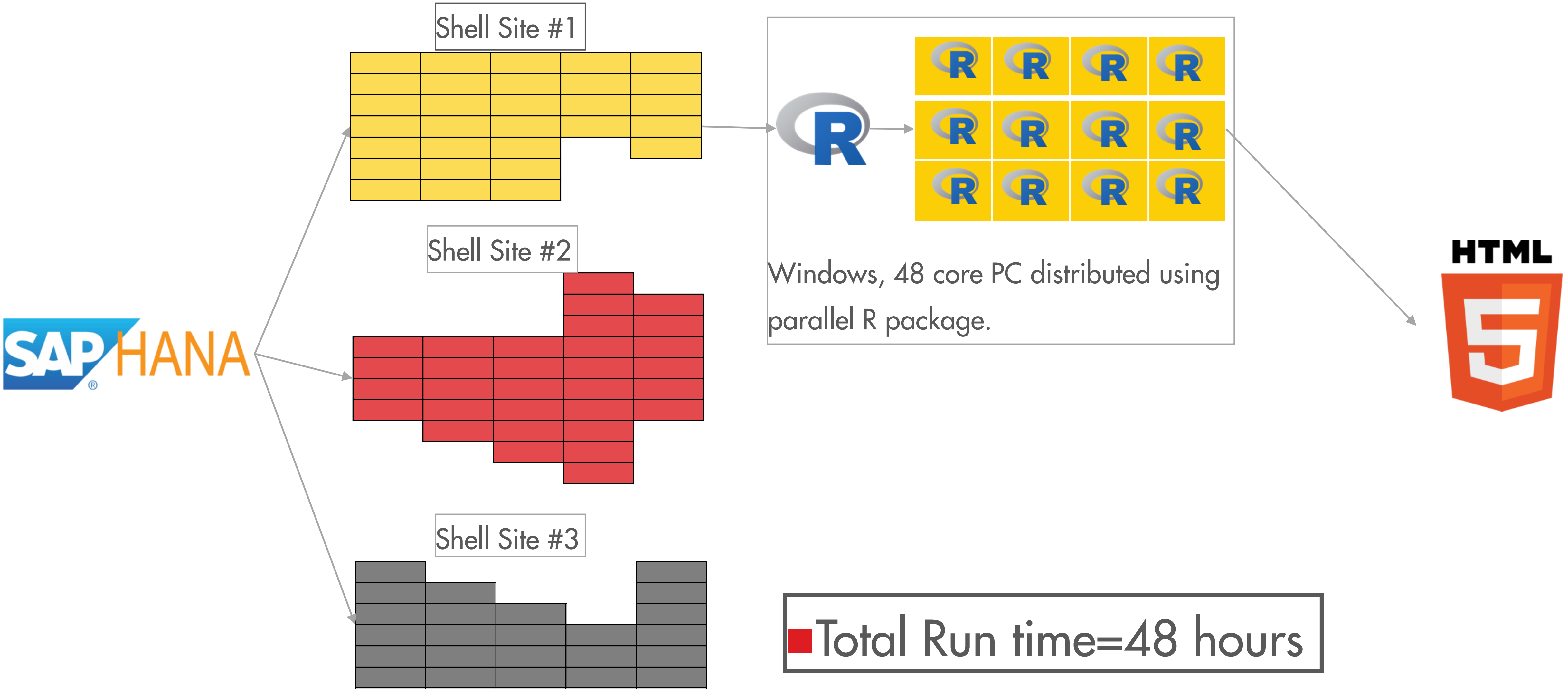
	Test 1	Test 2	Test 3	Test 4	Test 5
Total	16.21	6.45	5.17	3.58	0.44

Scale up Learning #1 – material sub-grouping.



- Embarrassingly parallel processing problem - Lots of independent small processing tasks.
- Each cell represents an individual material model– taking about 3-5 secs.
- Distribution via each individual cell, i.e. material, creates too much overhead.
- Instead the materials are grouped together into roughly equally sized groups.
- Each group job iteratively applies the model to each individual material.
- This mechanism vastly improved performance. This is true across all scale up architectures.
- This is different to traditional data “chunking” – because the data needs to be grouped according to materials.

Original Approach – iteratively applying parallelisation to each site.



Databricks

- Instead of relying on a single large machine with many cores, we used cluster computing to scale out.
- The new R API in Apache Spark was a good fit for this use case.

- For the first prototype, we tried to minimize the amount of code change as the goal was to quickly validate that the new SparkR API can handle the workload. We limited all the changes to the simulation step as following:

- For each Shell location:
  - Parallelize input data as a Spark DataFrame
  - Use SparkR::gapply() to perform parallel simulation for each of the chunks.

- With limited change to existing simulation code base, we could reduce the total simulation time to 3.97 hours on a 50 node Spark cluster on Databricks.

Anatomy of the SparkR gapply function

Cmd 10

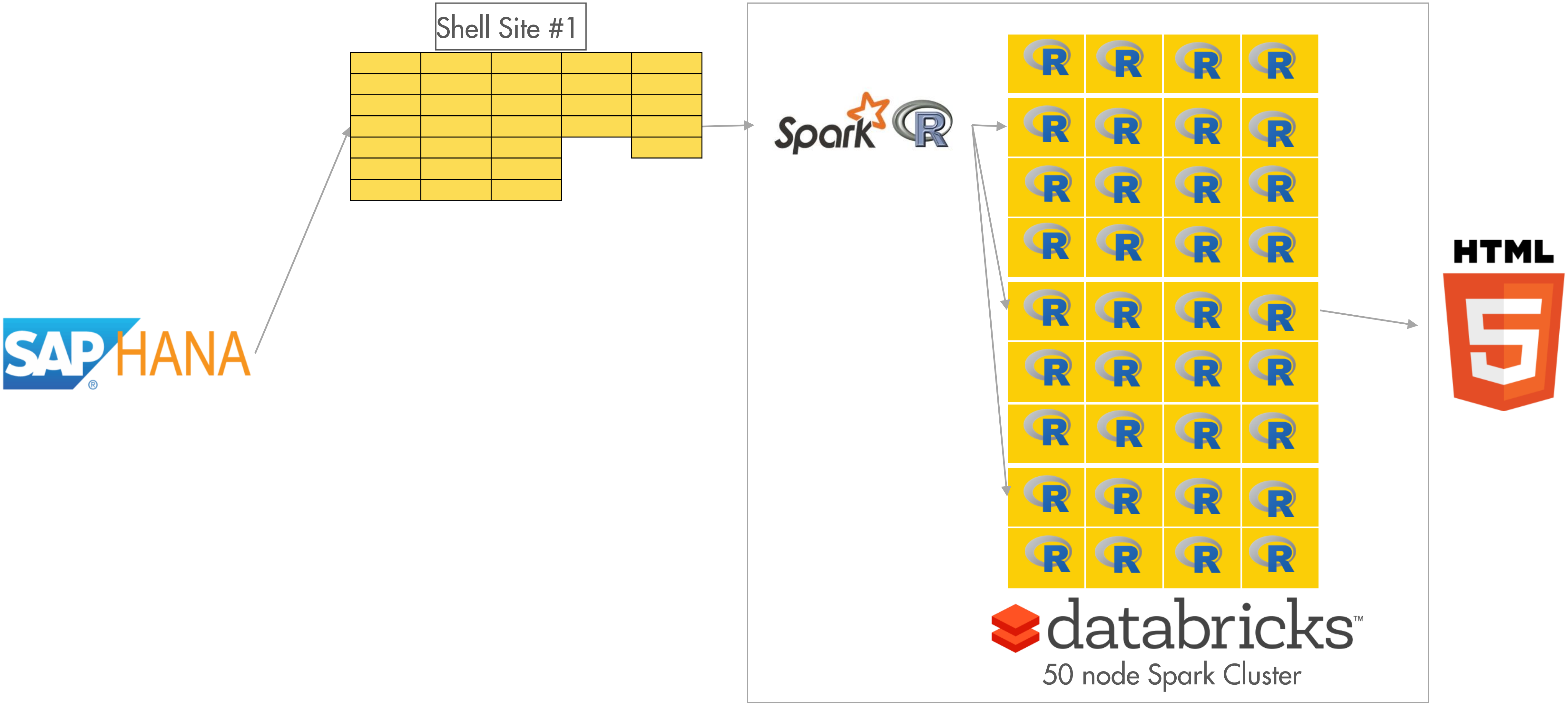
Calculate Bootstrap Service Levels per key_cols_boot

```
recomendation=SparkR::gapply(boot_in,key_cols_sim,
                             function(key,x){
                               library('dplyr')
                               sim=gev_sim(key,x[x$Table=='GR',],prior_var,sigma_prop,N_iter,N_burn,N_boot,key_cols_sim) ## simulate lead times for the combination of "key_cols_sim"
                               sim$Table='ST' # add table indicator to sim times table
                               sim=merge(sim,unique(x[,key_cols_boot]),by=key_cols_sim,all=T) # join sim times to the unique values of key_cols_boot present in x

                               x=x[x$Table=='MC',] # filter x for only rows coming from MaterialConsumption
                               x=dplyr::bind_rows(x,sim)

                               y=by(x,as.factor(x$MRP_Area_ID),bootstrap_service_level,key=key,service_levels=CT_CNFG_SL_TO,key_cols_boot=key_cols_boot,simplify=F)  ## calculate the RoP for each MRP area present in x
                               y=as.data.frame(do.call('rbind',y),stringsAsFactors=F) # convery output to data frame
                               rownames(y)=NULL
                               y
                             },RoP_schema)
```

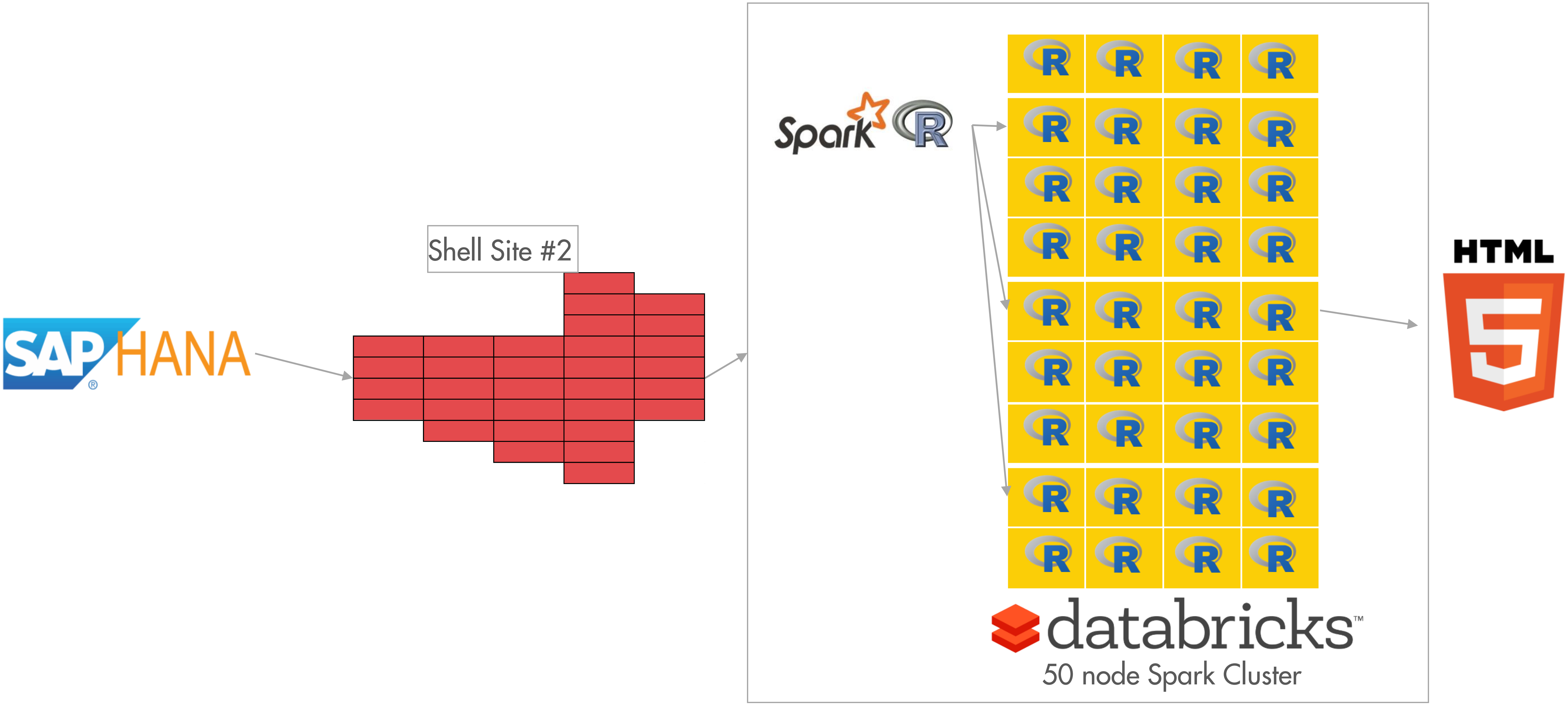
Command took 0.19 seconds -- by Wayne.W.Jones@shell.com at 10/23/2017, 8:31:55 AM on DATABRICKS_SUMMIT (clone) (clone)

Cmd 11

Scale up with 50 node Databricks Spark Cluster.

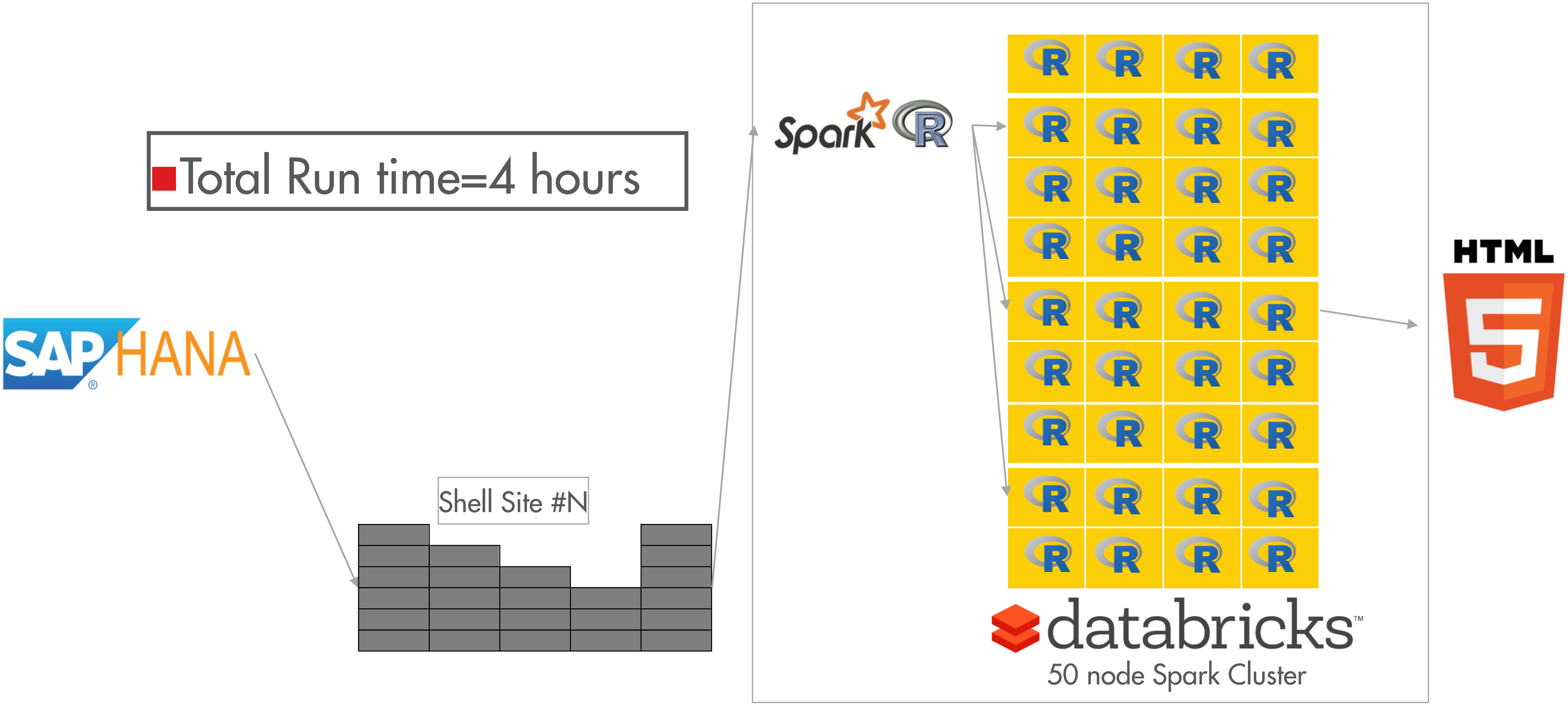


Scale up with 50 node Databricks Spark Cluster.

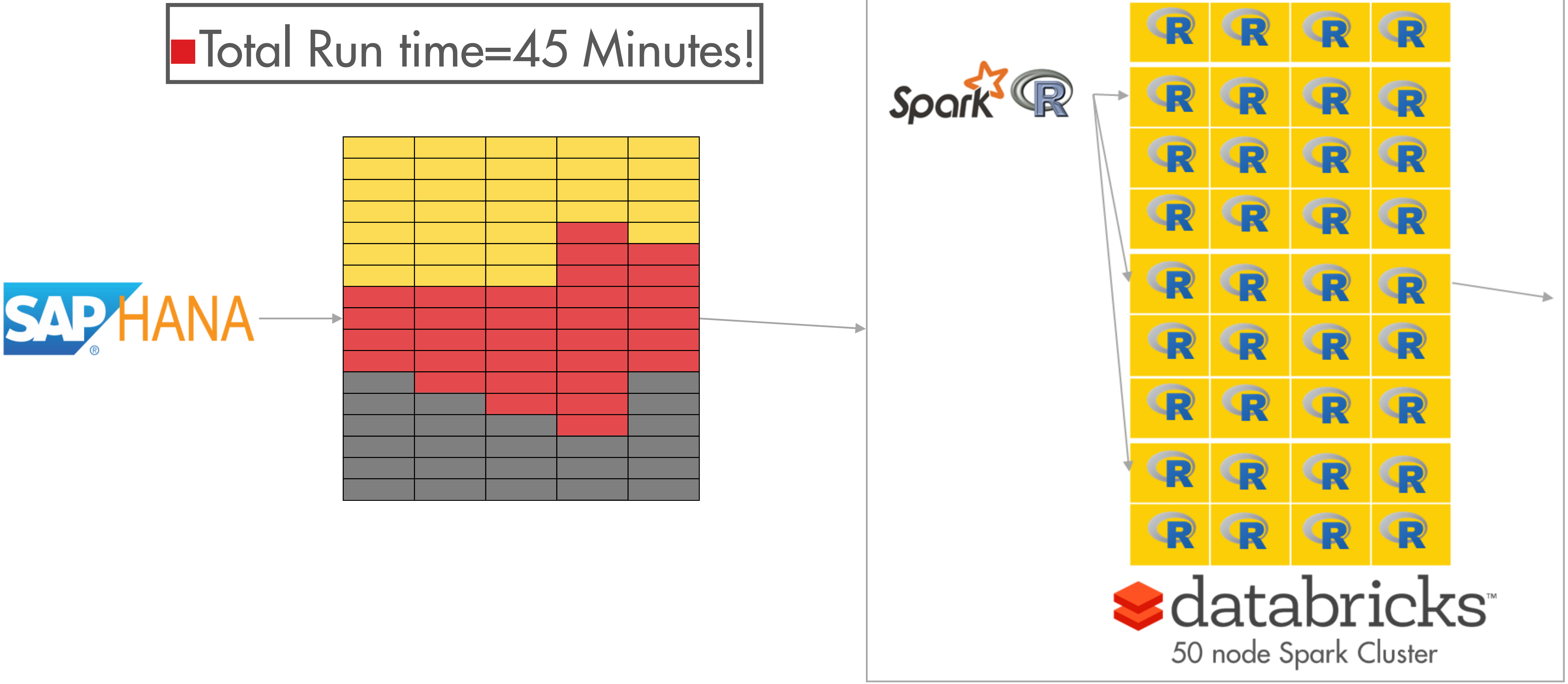


Scale up with 50 node Databricks Spark Cluster.

- Total Run time=4 hours



Pipelining approach – reducing over

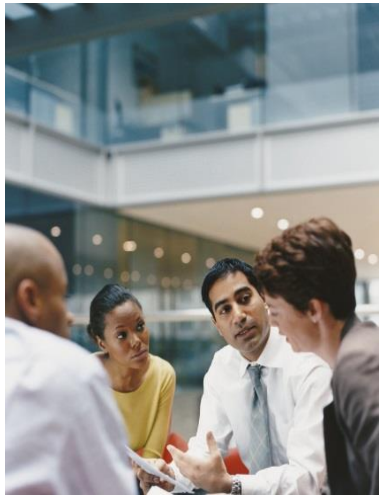


How this changes inventory management

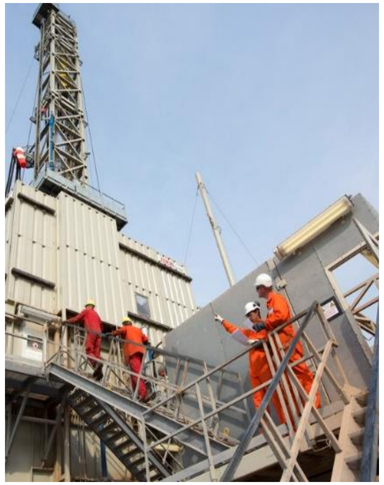
	Materials requirements planning	Dormant stock	Stock to bill of materials	Proactive inventory	Lead-time updates
Then	Materials requirements planning run without analysis	Dormant stock reports are time consuming to create	Stock to bill of materials linkage must be done on line-by-line basis using four different reports	No review of inventory levels	Historic lead times difficult to calculate and therefore rarely updated
Now	All replenishment orders scheduled via materials requirements planning are reviewed to ensure need and quantities	Dormant stock reports created easily with export from optimisation tool; all necessary data in one screen	All key data points in one screen and materials requiring review are easily pulled from optimisation tool export	Inventory levels reviewed on a value basis	Historic lead times provided in tool using sophisticated method
Key successes	Reduced stock replenishment spend	Dormant stock reduction amounts to millions of dollars	At Auger and Mars 100% stock to bill of materials linkage All other assets improved linkage by more than 10%	Inventory balance reduced by several million dollars 2016 (Gulf of Mexico wide)	More than 3,500 material master record lead times updated to provide more accurate estimated times of arrival for materials

Current status

Growing Shell-wide analytics capability



Potential value partners – independent use of algorithm



Global project in IT delivery phase – replication across Upstream and Downstream assets

Any Questions?

Q&A