

CitA

BIM GATHERING 2019



4th CitA BIM Gathering 26th September 2019, Galway, Ireland.

Delivering **better outcomes**
for Irish Construction

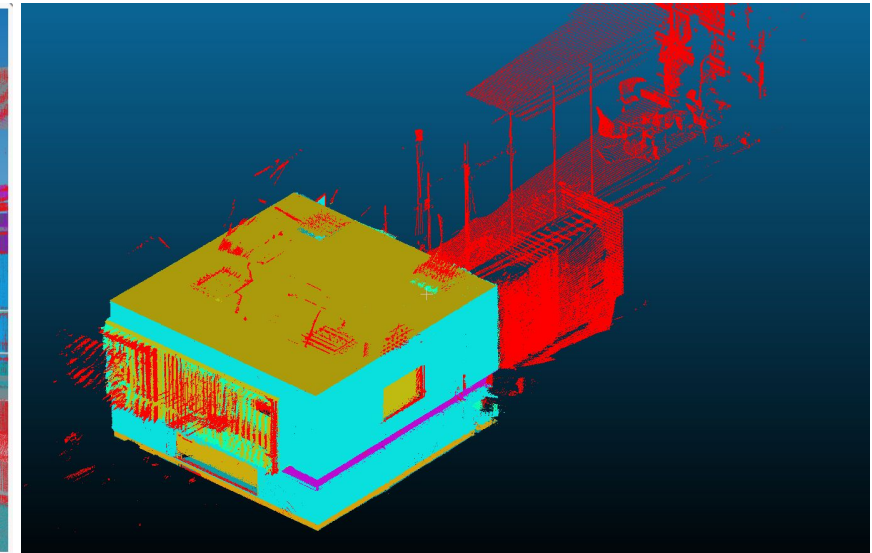
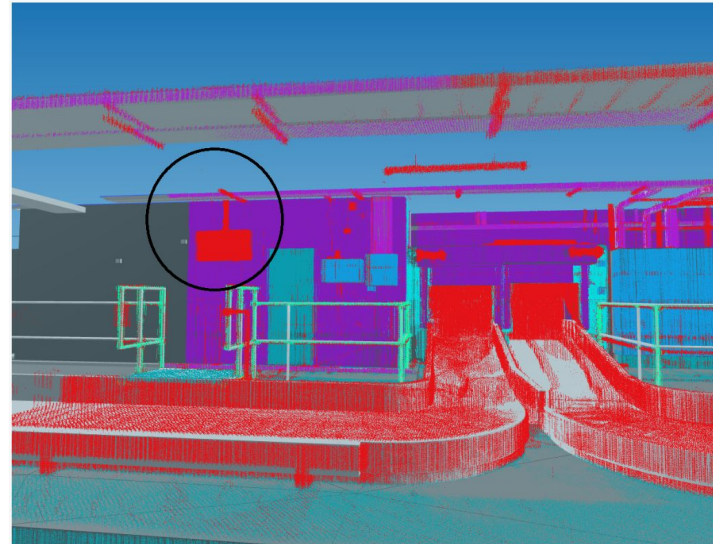


External Memory Solution For Large-Scale Point Cloud Data Processing

*Neil Hyland & Dr. Shawn O'Keeffe
BIM & Scan Ltd.*



- BIM & Scan® Ltd.
- BIM & Scan AutoCorr™
- Cloud Service
- Adaptation/Optimisation
- Testing
- Results/Conclusion



	A	B	C	D	E	F
1	STATUS	TYPE	NAME	GUID	SCAN	POINTS
2	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LM0	0	0
3	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LM1	0	0
4	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LM6	0	0
5	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LM7	0	0
6	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LVa	0	0
7	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LVb	0	0
8	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LVw	0	0
9	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LVx	0	0
10	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LUg	0	0
11	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LUh	0	0
12	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LUe	0	0
13	Missing	IfcMember	Curtain Wall: Curtain_Wall-Storefront:8486	2Nr59kFsLE0Qduw9nk3LUf	0	0
14	Found	IfcFlowFitting	Elbow Long Radius - Butt Welded - CS-Standard:572738	1F8KdLJEHEDA7IWJqXXXx0	2	12860
15	Found	IfcFlowFitting	Union - Slip-on - Steel - PN 25:Standard:573597	1F8KdLJEHEDA7IWJqXXcCV	2	6311
16	Found	IfcFlowFitting	Elbow Long Radius - Butt Welded - CS-Standard:572629	1F8KdLJEHEDA7IWJqXXXzN	2	5502
17	Found	IfcFlowFitting	Elbow Long Radius - Butt Welded - CS-Standard:572734	1F8KdLJEHEDA7IWJqXXXwv	4	5247



In AEC industries, how do we assure that what was built on-site, is what was designed?

What needs to be done to determine this?
What tools can/are to be used?

How do we assure stakeholders that this is an objective fact?





BIM & Scan SIGN OUT

[Admin](#) [File Manager](#) [Execution](#) [Account](#)

Run: BIM & Scan AutoCorr™ Change User

INPUT

OUTPUT

Find your files below, choose your tolerance and run BIM & Scan AutoCorr™

Model File (IFC2x3 CV 2.0)
AutoCorr_PlantRm_Test_Mode...

Point Cloud (Ordered e57)
AutoCorr_PlantRm_Test_PointCloud.e57

Tolerance
55

5mm (min) to 200mm (max)

[Watch: How to Run BIM & Scan AutoCorr™](#)

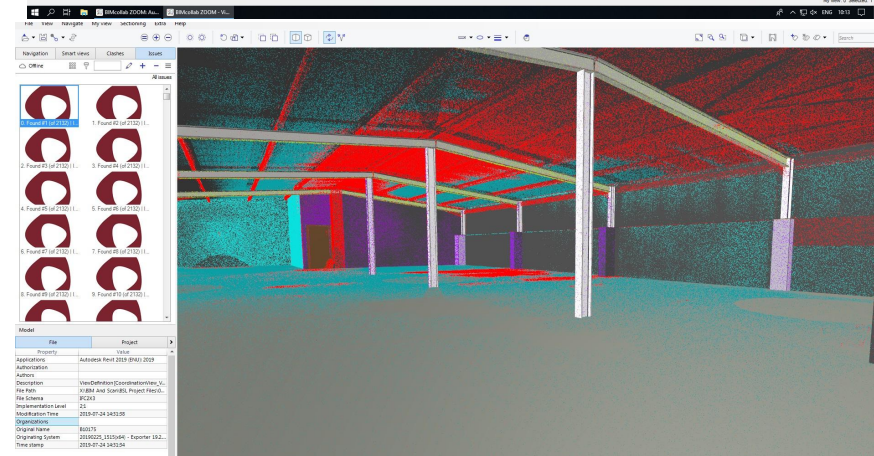
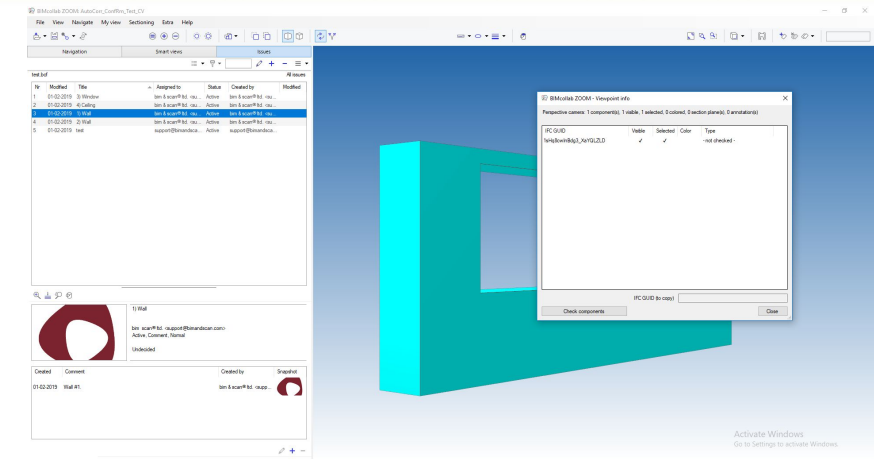
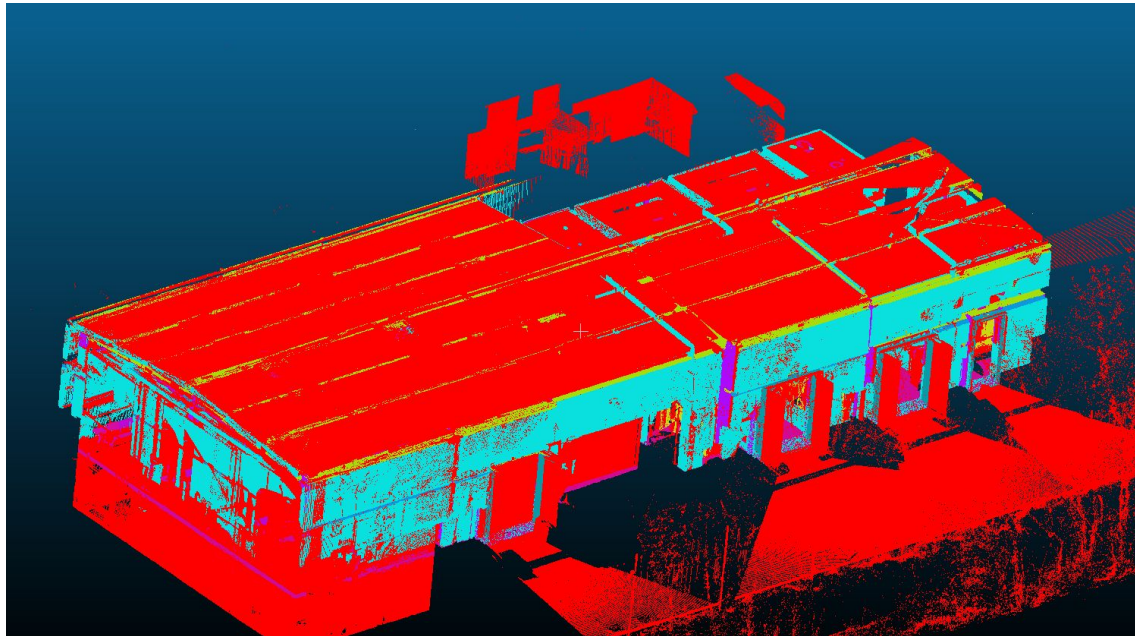
Update Input List Run

BIM & Scan AutoCorr™ is a cloud-based software service that performs analyses between models (IFC BIMs) and point clouds (E57 scans), and reports correspondences between the real-world built environment and its virtual model counterpart.

Available from our website
(bimandscan.com) for users to try.



Example results...

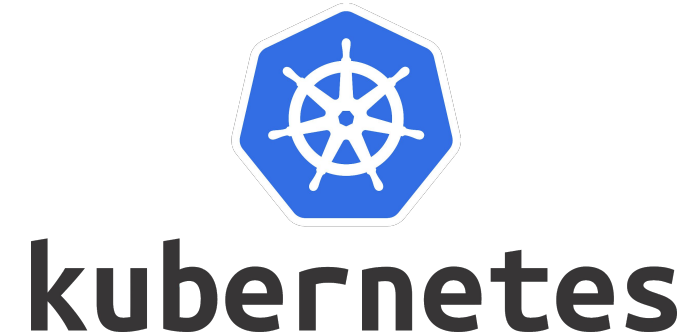
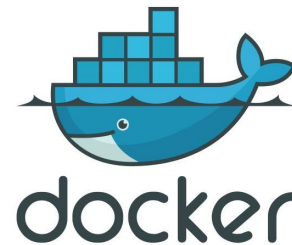




- First version: command-line executable.
 - Local desktop software tool.
 - Used by developers/those knowledgeable.
 - Hard to install/deploy elsewhere
- Second version: cloud service.
 - Available across the Internet.
 - User-friendly.
- Third (newest) version: optimised cloud service.
 - Better backend.



- Deployed on Microsoft® Azure, using Docker/Kubernetes as an orchestration system.
- Available performance (RAM, CPU size, HDD capacity) comes at a premium.
- “External” memory integrated into software to help alleviate costs.
- *Did our improvements achieve what we wanted to achieve?*

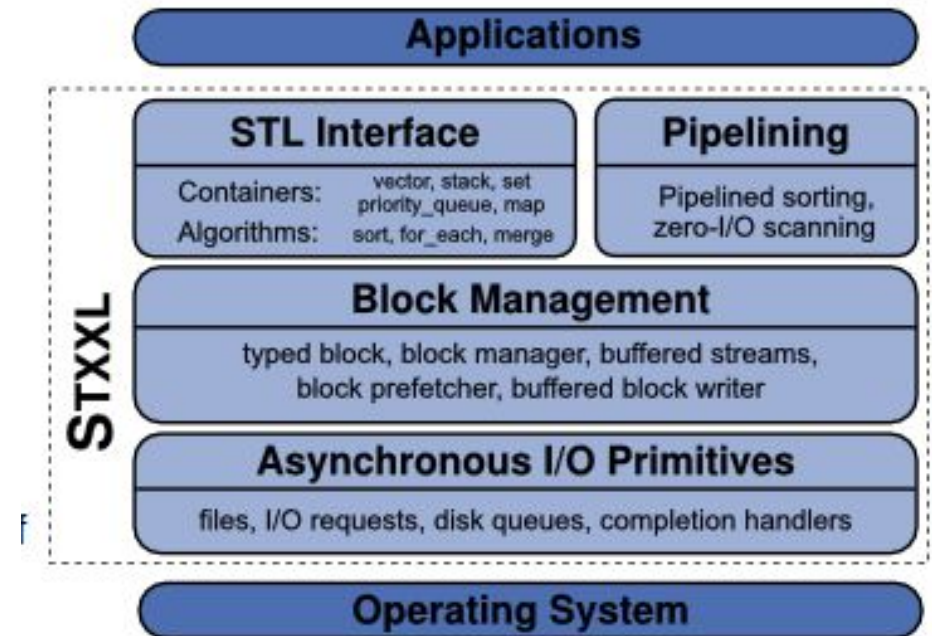




- Additional, optional layer between the software and RAM.
- (De)serialises data to-and-from hard disks instead of memory.
- Similar to OS functionality (e.g. paging, swapping).
- **Used to handle datasets larger than what can fit in RAM.**
- Applied currently to HPC, big data, distributed/cluster computing domains.
- **We saw how it could help lower our memory footprint, and therefore cost.**

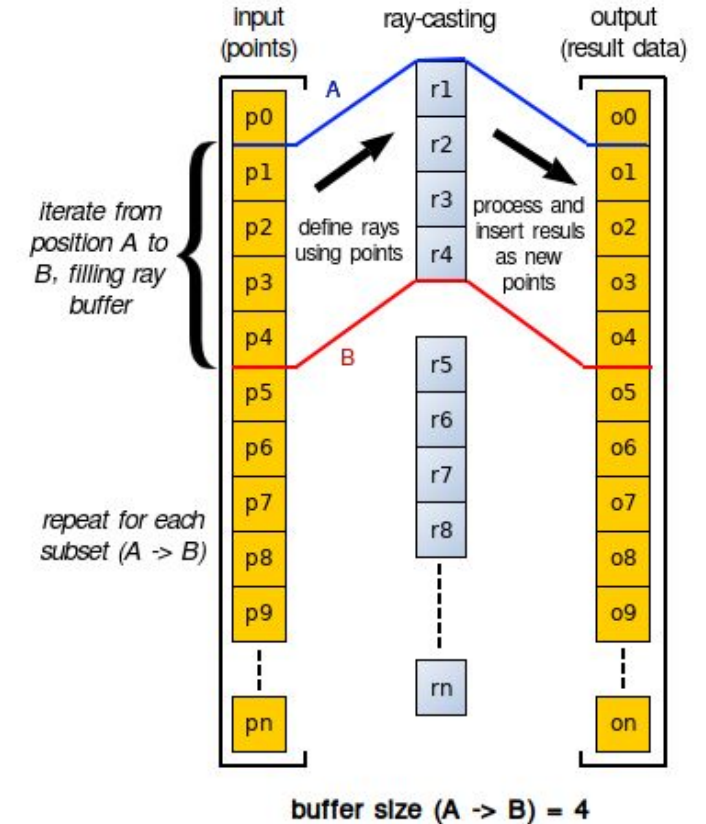


- STXXL (“Standard Template Library for Extra-Large Datasets” | <http://stxxl.org>)
- An existing stable/mature C++ library.
- Provides data structures built on external memory.
- Attempts to maintain performance similar to in-memory computation (best case scenario).
- Sequential streaming of data useful for converting our algorithm.





- Samples point cloud/scan input to reduce size but keep same structure.
- Performs bulk ray-casting using point data and processes results into new dataset.
- Incurs high memory overhead when operating on medium-to-large inputs.
- **Ray-casting best candidate to use with external memory.**





Based on hardware/infrastructure used...

<i>Software</i>	<i>Execution RAM</i>	<i>Cloud VM RAM</i>	<i>Execution Capacity</i>
In-Memory	8Gb	32Gb	~3 (*)
External Memory	2Gb	32Gb	~15 (*)

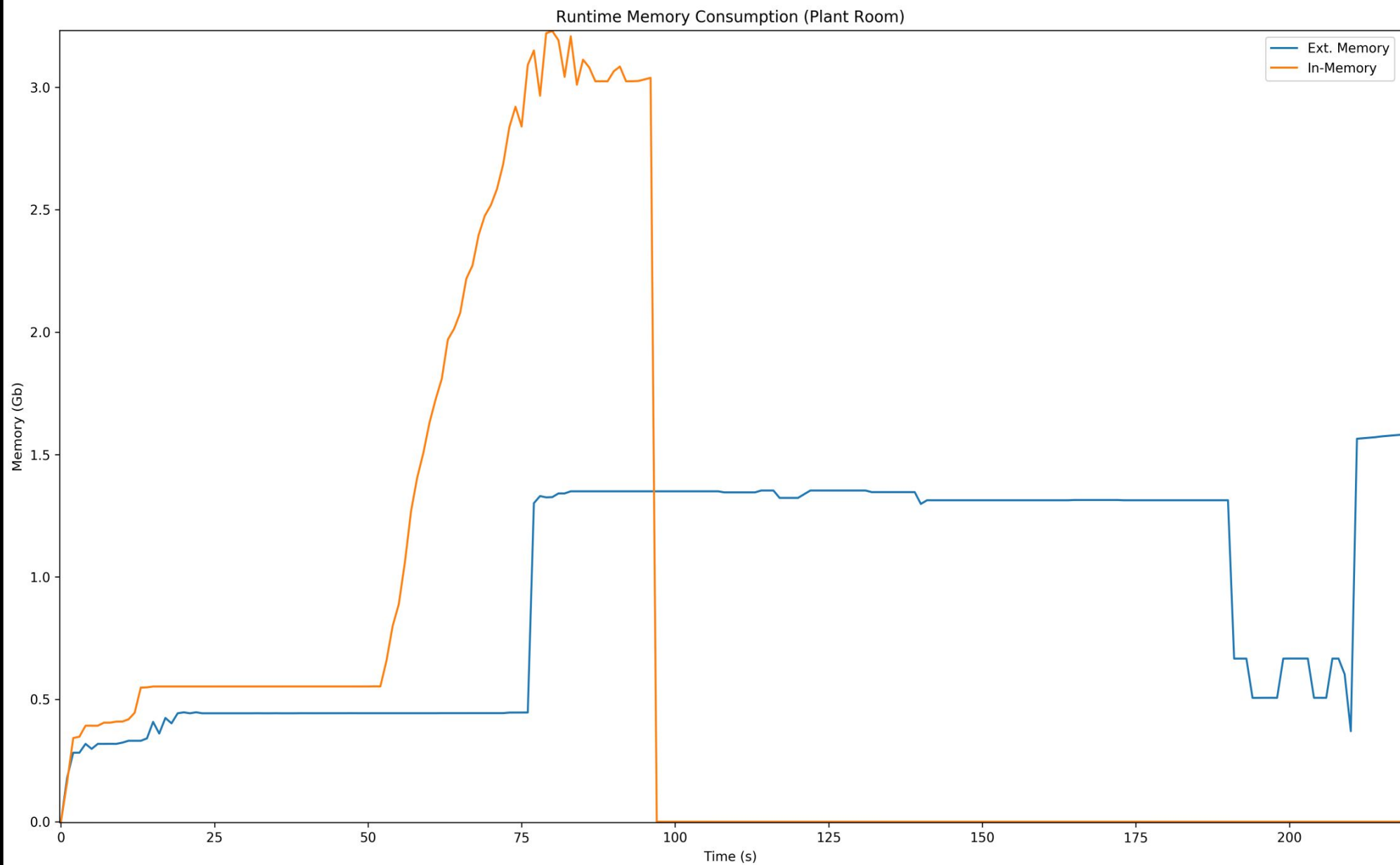
* theoretical upper bound after subtracting overhead for OS, management software, etc...



- Software deployment.
 - Local: command-line Docker container, unlimited RAM.
 - Cloud: standard user-submitted execution, fixed amount of RAM.
- Measured time taken, memory/RAM consumed.
 - Local: per-second snapshots of performance characteristics of container.
 - Cloud: timestamps of execution start/finish.

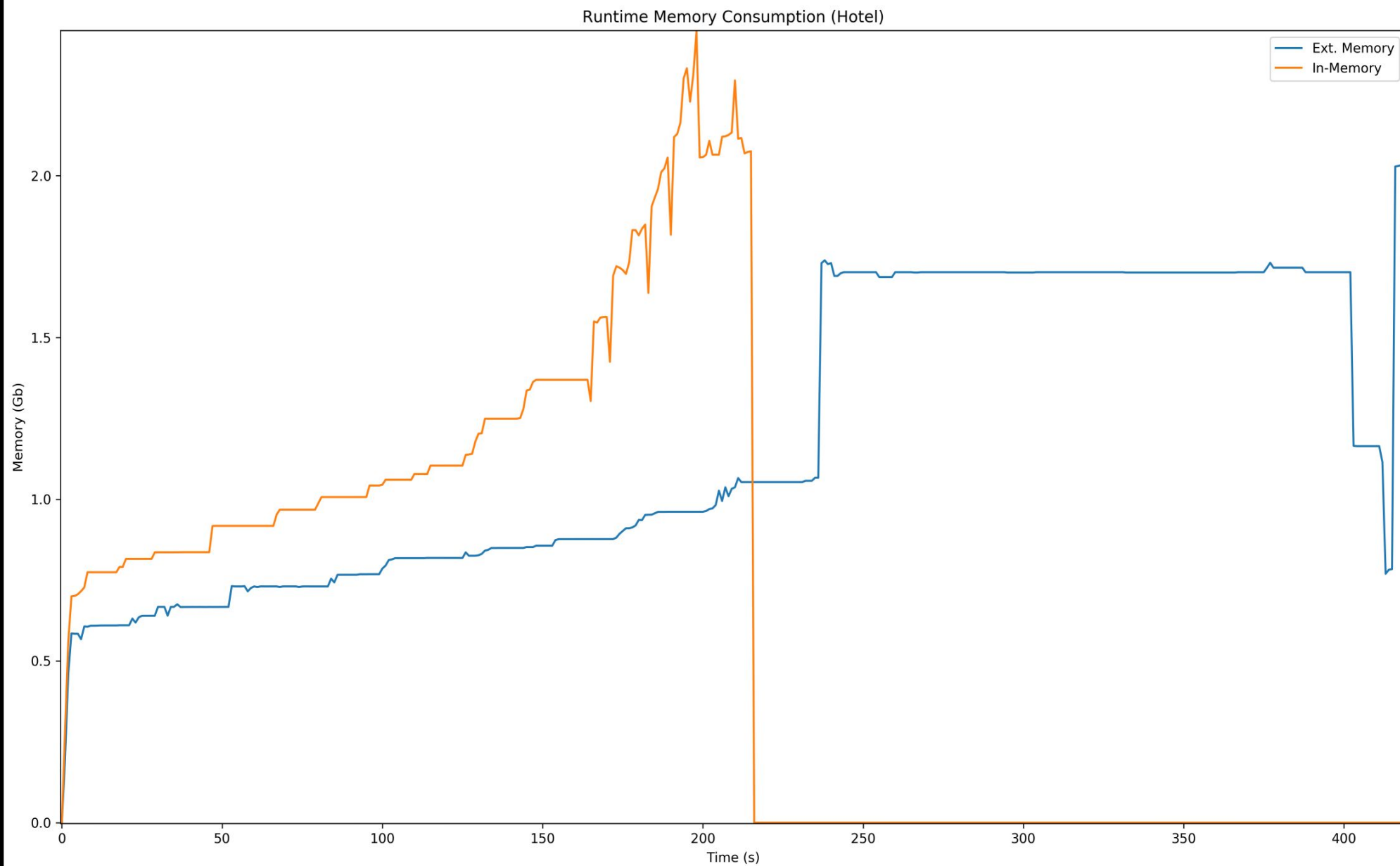
“Plant Room” dataset.

Size: ~4.5Gb



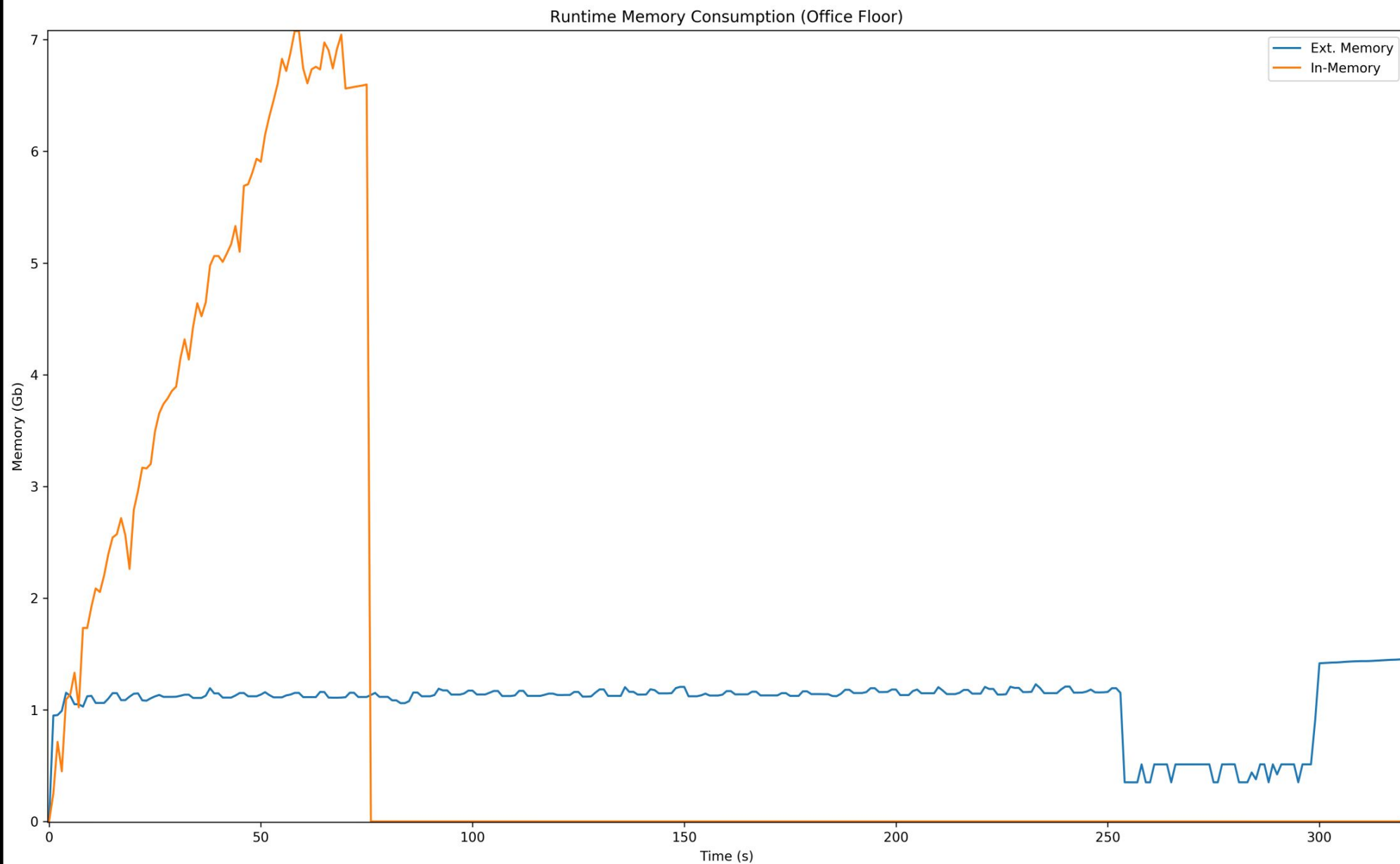
“Hotel” dataset.

Size: ~7.6Gb



“Office Floor” dataset.

Size: ~6.1Gb





- Approx. double time taken for external memory version.
- 50-75% slower on average.
- However, huge reduction in memory overhead.

<i>Perf. Timing</i>	<i>LOCAL</i>	<i>CLOUD</i>
Plant Room	44.5%	41.1%
Hotel	51.6%	44.6%
Office Floor	23.8%	45.7%

Execution time of in-memory version as a percentage of external memory performance.



Throughput (considering parallel execution):

$$T = C / D$$

(capacity divided by duration)

e.g. Plant Room dataset...

$$T_i = 3 / 392s \quad \Rightarrow \quad 0.0077 \text{ (exec. per sec.)} \quad \Rightarrow \quad 0.462 \text{ (exec. per min.)}$$

$$T_e = 15 / 954s \quad \Rightarrow \quad 0.0157 \text{ (exec. per sec.)} \quad \Rightarrow \quad 0.942 \text{ (exec. per min.)}$$



Despite longer execution times, overall performance of the cloud system is maintained thanks to parallel throughput increase.

Increased throughput also makes the most out of our current cloud infrastructure. More executions in a single VM means lower cost per execution.

Scaling costs is controlled in a similar way.



Thank You | Questions?