

MULTI THREADING MODEL IN PARADOX GAMES: PAST, PRESENT AND FUTURE

MATHIEU ROPERT

Multi Threading <mark>Model</mark> in Paradox Games

Past, Present and Future

NV.



HIS DIVINE SHADOW - 23 Oct, 2018 @ 10:29pm

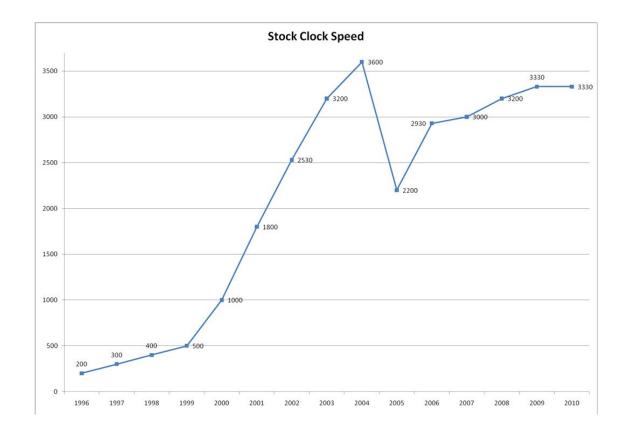
How to use multithreading in Paradox games?

Is it even possible? maybee some software trick?

How indeed?

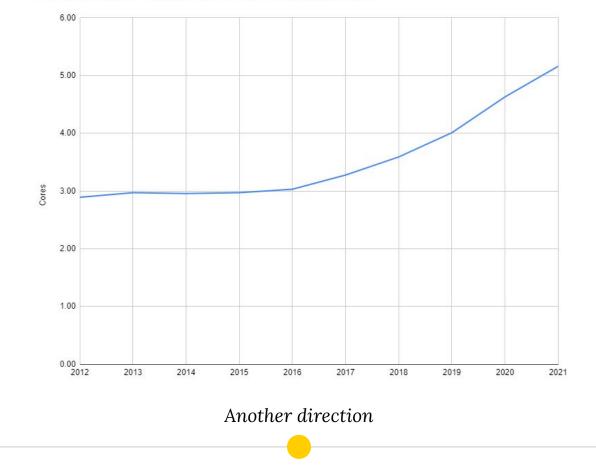


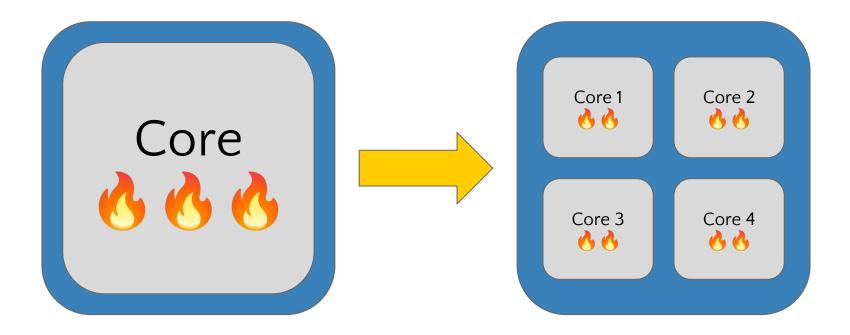
- At level 7 pick Improved Software Trick
- At level 13 pick Greater Software Trick
- Consider the Software Trickster prestige class
- Best done with Gnomes or Catfolks



Stagnation?

Average number of physical cores in Steam playerbase





New hardware, new challenges

Hello!

I am Mathieu Ropert

I'm a Tech Lead at Paradox Development Studio where I make Hearts of Iron IV.

You can reach me at:

- **MatRopert**



https://mropert.github.io



- The importance of multi-threading
- Concurrency models...
- ... in practice
- Tips & tricks



Dev Workstation

- Intel i7–7700
- 4 cores / 8 threads
- nVidia GTX 1060
- Optick profiles

Home PC

- Intel i7–10700
- 8 cores / 16 threads
- nVidia RTX 2080 Super
- vTune profiles & Demos



Conference Laptop

- Intel i5-4300U
- 2 cores / 4 threads
- Intel HD Graphics
- Google Slides

Remote Workstation

- Intel i7-12700
- 12 cores / 20 threads
- nVidia RTX 3060
- Demos (hopefully)





- Concept is quite old
- pthreads were introduced in 1995
- Most (all?) software engineering classes will cover the basics



- Desktop machines with more than one CPU only appeared late 2002
- Consumer CPUs with more than one core came up in 2005
- 2+ cores became default for Intel only in 2010

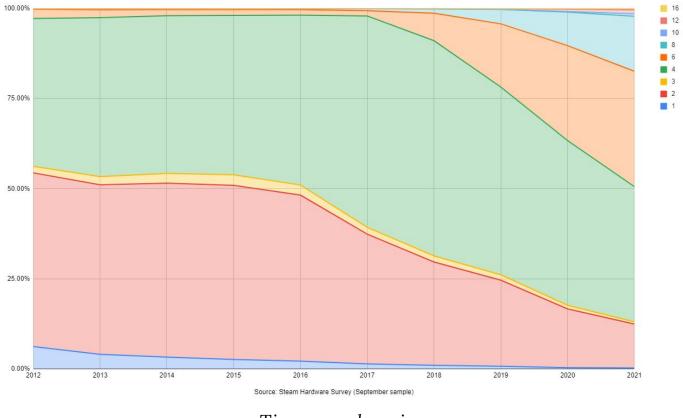


- Efficiency is limited by the number of CPUs
- Over-subscription worsens performance
- Don't spawn more work threads than you have cores

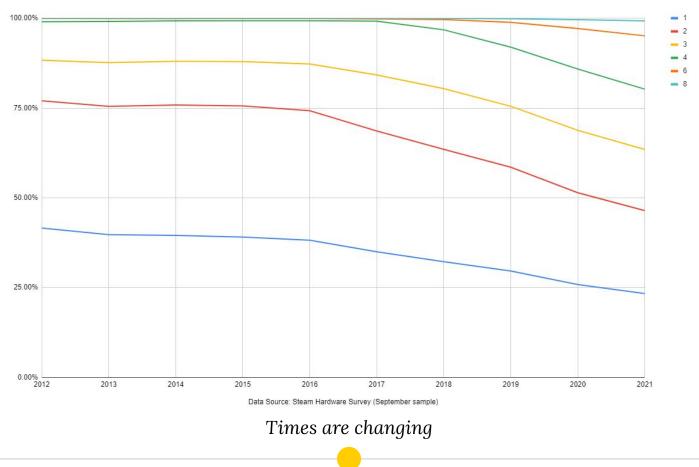


- For a long time the average desktop only had 1 or maybe 2 CPUs
- Good multithreaded code is harder to write
- Use threads for async operations, keep most of the busy work on the main thread

Steam Playerbase Physical Core Count



Times are changing



Player CPU Usage Over Time by Average Number of Work Threads



- Mono thread computation only utilizes about 25% of the userbase processing power
- Multithreading computation is not just a bonus for high-end desktops anymore
- Code needs to adapt

Paradox Games and 2 — Multithreading

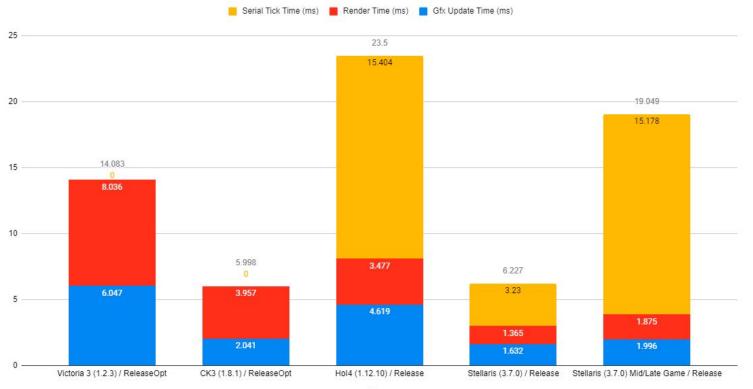
A history of historical strategy games



PDS Releases Timeline

PDS Games Frame Time (ms)

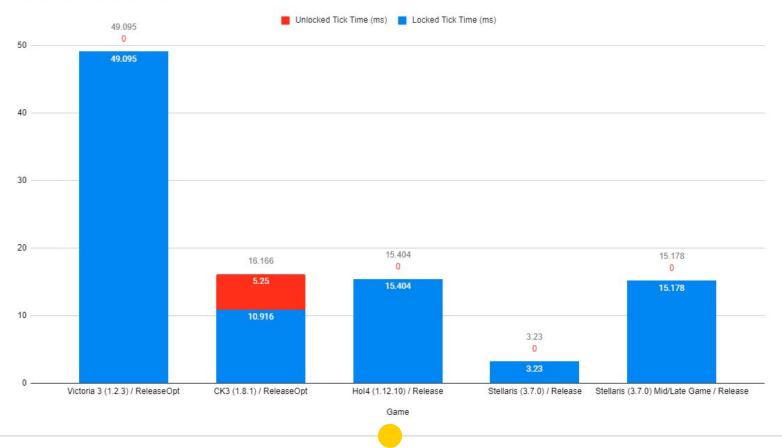
i7-12700K / 32GB RAM / RTX 3060



Game

PDS Games Tick Time (ms)

i7-12700K / 32GB RAM / RTX 3060





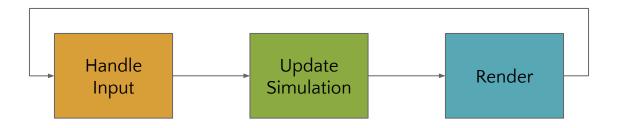
- All games use the same in-house engine, dubbed Clausewitz
- Up until Imperator (2019), games forked the engine at some point during development
- Big generational jump between Stellaris (2016) and Imperator (2019), dubbed *Jomini*



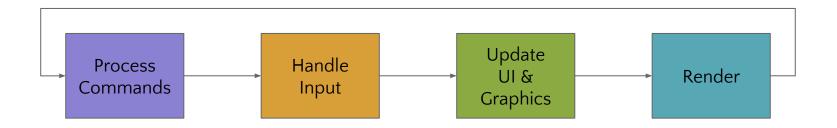
- Started with Crusader Kings II (2012)
- Multithreading done through TBB
- Mostly focused on speeding up the world simulation



Demo Time!



A Basic Game Loop



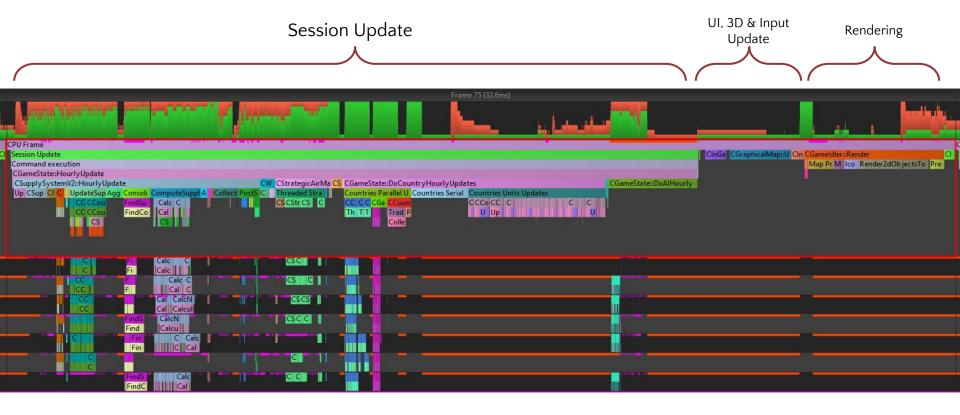
Past PDS Games Loop



- Gamestate can only be changed through command execution
- Player interactions with UI result in new commands being added to the queue
- Server queues a command to advance time by one unit at real time intervals



- Passage of time is simulated by "tick" increments.
- Depending on game, "ticks" can be an ingame hour (Hol4), day (CK2 and EU4) or fraction of day (Stellaris).
- No in-between!



Hearts of Iron IV Sample Update



- Each sub-system update is run in a sequence
- Core utilisation depends entirely on how a given system is implemented
- Most multithreading is done through parallel_for()



- Rule of a thumb: more recent systems have better threading efficiency
- Some have been retrofitted over the years to use parallelization
- UI & Graphics update / rendering are not done in a dedicated thread



- Board game heritage can still be felt in some game systems
- Unit/Combat update rely on sequence to be deterministic
- Hard to address in an existing game



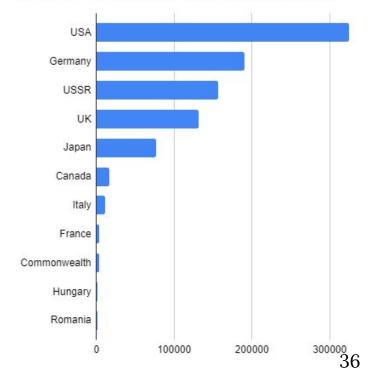


- Parallelizing updates by sub-system is limited by entity grain size
- Hol4 has:
 - 13236 provinces (tiles on map)
 - 835 states
 - 295 countries



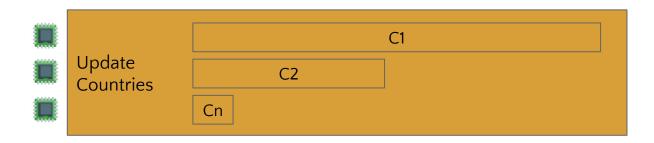
- Not all entities are created equal
- Custom scheduler can help a little
- Some optimizations turn out to be pessimizations

Aircraft Production of WW2 (approximate)





Past PDS Games Loop



Past PDS Games Loop



- Entities within a system are not equal
- System update will be as fast as the slowest entity to update, even with many cores
- Large entities in one system tend to also be big in other systems using the same breakdown



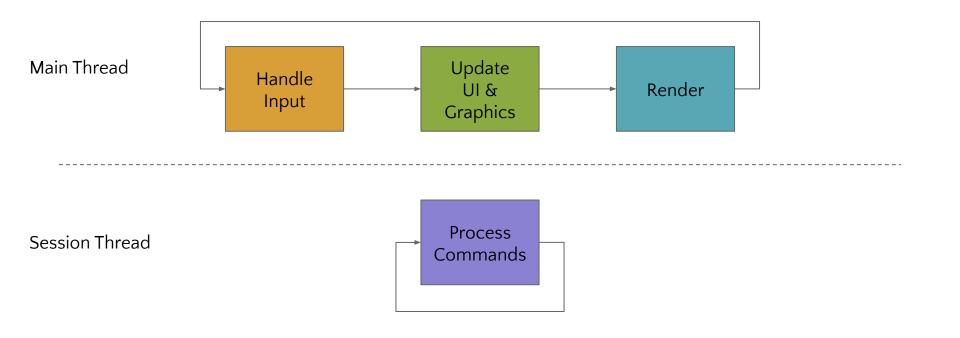
- Good enough for the time
- Some systems manage to utilize all cores
- Refitting older systems can be difficult and risky unless willing revisit game design

Multithreading in 3 Current Generation

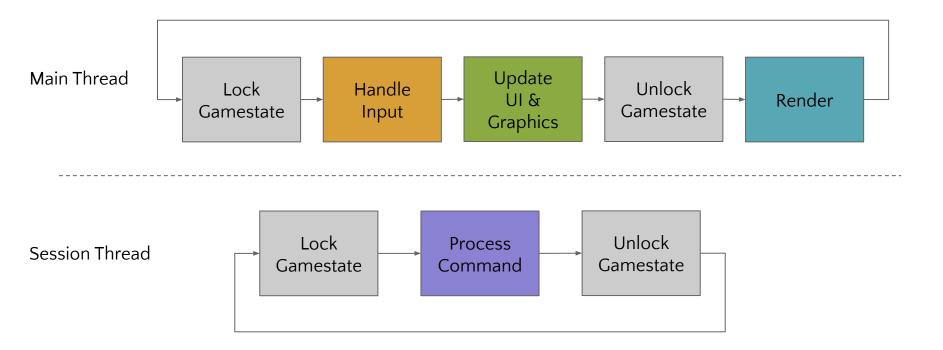
Illustrated mostly by Crusader Kings 3



- At some point both Imperator, CK3 and Victoria 3 were in development simultaneously
- Same engine, different approaches to simulation update



Present PDS Games Loop



Present PDS Games Loop



- Dedicated render thread guarantees at least some degree of multithreading
- Doesn't solve the biggest CPU bottleneck (gamestate update) out of the box





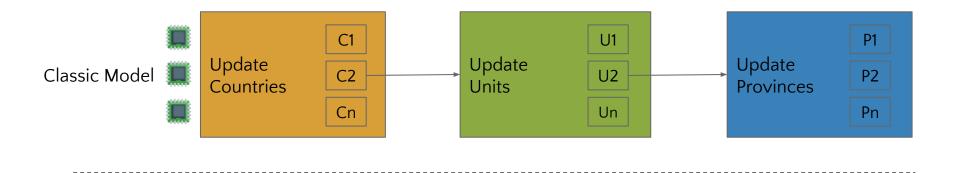
Demo Time!

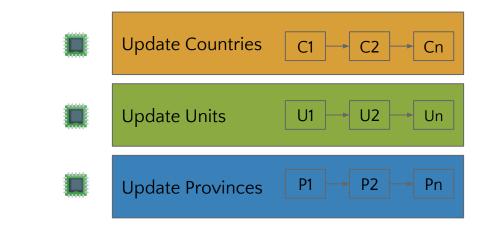
															Parallel																				
▶ CPU							Τï	W			J.											ų r									T				
Main Thread (0xa44)	CP CP CPU		CP C	C C CP	CPU C ano	CPU F	CPU Frame anony mous		CPU Fr		PU Fr C	CPU Frame anonymo					C CP		CP CPU F an `anon	C CP C	CPU a `ano`	CPU Fra `anonym	C CP (PU Fra nonym	CPU Fr C anony	CCP C	CPU F C	CP C C	CPCPC	CP CP an an	CCPU Fr anon		C C CP	CPU Fr anony
Long Task 1 (0x4a10)	ga ai C	gam o c	ai	ga o r			ai_manag	ge			ai	gam mo		ga c		game o g r		gam char	ai_	gam c	ai	ga	1	ga ai	ai_	ga co	a	ga		ga re	a	gam cha	a g	am Ja har	
Long Task 2 (0x701c)	a)_	gam ai	ai	ga on	ai	gam cha	ai_manag N		games ai g a		-	game o c		ga cha	ai		ai	ga c	ai_	gam o m	ai	ga r o		ga gu	ai_	gam ch	a	ga c		game cha	aiN	gam o	a g	ii P	gam cu
 Long Task 3 (0x6e08) 		gam cha	ai	game cult	ai	gam ai m	ai_mana NAI::Str		game cha		ai	game char		ga 🛛	a	ga a c	- [ga o	ai_	gam ch	ai	ga cha	a	ga cha	ai_	gam g	F	game chara		game o sc	ai_	gam ch	9 s	a a	gam cha
Long Task 4 (0x3fa0)		game guest	ai	game char	ai	gam cult	ai_manag NAI:	er.	games cha r	tatec theoc	ai	game ai s	1	ga o	a			ga s ai	ai_ C	gam ai gu	ai	ga cha	a	ga cha	ai_	gam cul	a	ga ch	T		ai	gam re	a g	a	gam ch
Long Task 5 (0x4af0)	ga ai_ ai	gam cul	ai	gam ai	ai	gam char	ai_manag N	er.			ai	game op re	а	garnesta characte	a	games ona w	T I	ga o	ai_	ga		ga c	a	ga o	ai_		a	ga	H	gam	a_	gam sche	9	a a	gam ai
Long Task 6 (0x4478)	ga a r	gam r o			a	gam g r	ai_manag	er	games o o	ta pr	ai	games c gue		ga	ai_ CAI	game cha s				ga rei		ga o		ga	ai_	gam ch	a C	ga o	H	game opin		ga o a		a	gam on
ж Long Task 7 (0хбе00)	ga la_		ai	games charac		gam op	ai_manag NAI::	er	game					ga o	a	game cha		ga r	ai_	gam o w	ai	ga ai	a	g o	ai_	gam o	a	iga g		game cult			a g	ame a	•
✓ Network/Session thread (0x25dc) ExecuteG E C S CDai ga C CGa ga D CAI AIU ai ai C C	x Execut S CDail S game (gam T ch	teGa Ex S C S C C C D C C Al	Execute CDaily gamest game S gu s	Came E S CD C CG D CA Al ai ai		GameComr CDailyTick CGameSta CAIManag AIUpdateT ai_manag ai_manag	cCo S teHi er::U Tasks er.cp	gamest gamest	TickC Sy	omm Ex n CD S Da CG ail CA Al ai ai	Executed CDaily S game D game D chara		CDaily Ti	ameCom Syn CDa CDai CGa Dail CAI AlU ai ai S	CDaily	eGiam Sy C C C Da C A ai a	Ex Execu S CDail game ga cha	teGame Sy CDa CD CGa Da CAI AIU ai ai	E Execu CDail gam gam char	uteGa E S C C C D C Al ai ai	Exe Exec Sy CDa CSc gam gam gue	S C C C D C	x Execu S CDa S C ga C ga D c	teGam E CDail CGa CGa AIU AIU ai_ ai_	Execu CDail game gam ai C	teGa E Sy C C C Da C Al a a	Execut CDail S game C ga S a r C	eGa E y C C C Da C	Executo CDail S game C game E char	Garne E CDail CGa CAIM AlUp a a	x Execu S CDai gam gam cult	teG E E C C g C g A c a a	x ecuteGa Dail S C ame C C a D C ha A a a	E gam

Crusader Kings 3 Profile



- Parallelize updates by system instead of by entity
- Split updates between bits that needs read and write access to gamestate
- Do the heavy lifting with only the read lock if possible





CK3 Model

Past vs Present PDS Games Loop



- Read-lock part of update can only modify "private" data in the gamestate
- Write-lock part of an update can change any data in the gamestate
- Try to keep most of the update in the first part



- Entities within a system are guaranteed to be updated in a deterministic sequential order
- Read-lock part of a system update can be parallelized with other systems updates
 And Input update, Graphics update and rendering



- Experience has shown this model is easy to teach to newcomers
- Explain the constraints of the 2 update steps
- Newly added system immediately benefit from multithreading performance



- Architecture-wise, the CK3 model is the one with the most potential
- Imperator used an update model fairly similar to the previous generation
- Victoria 3 tick is a series of tasks, but it can't run them in parallel 😢



- Better threading efficiency even when combined with old school update patterns
 Dedicated session thread
- CK3 is really fast
- Model is easy enough to teach, but not enforced by the engine API

Thoughts for the 4 – Future

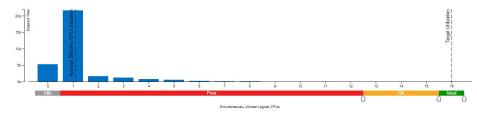
Where I look anxiously look at my NDA

⊙ Effective CPU Utilization[©]: 6.8% (1.086 out of 16 logical CPUs) ►

Effective CPU Utilization Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.



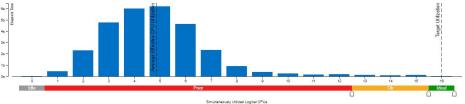


Seffective CPU Utilization[®]: 30.4% (4.871 out of 16 logical CPUs) ►



This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.





Generational Gap



- CK3 model has proven to be quite more effective than the others
- No immediate limitation to solve
- Focus on making it more accessible as a design pattern



- Not formalized / enforced by the update API
- More of a best-practice to teach each time
- Implementation split between game and engine



- Try the same model in next title
- Move the base update model to the engine
- Rename/refactor interface to emphasize the read-lock vs write-lock update steps



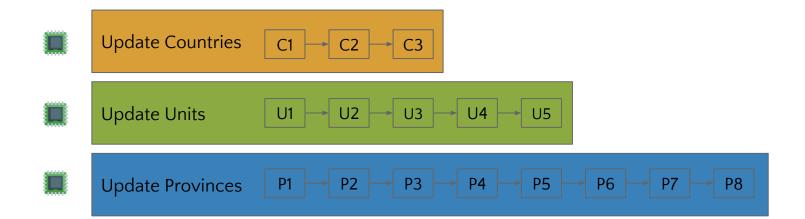
- Look at potential limitations or future hindrances
- Current average CPU utilisation on CK3 is around 5 out of 16 cores
- Can we do better?

Ini Cha M M CA AlUpdateTasksInPara ai_manager.cpp:1001 STaskItem:	_)	37 (17.0n	Frame 13	
anonymous-namespace':UpdateRunners Execut SynchronizedCo CDaily Tick ComeStateHiddenData Wi CEve gamestatecommands CAI Manager:UpdatePostEx Lii Cha M CA AIUpdateTasksInPara ai_manager.cpp:1001 Characters CAIWarCoordinato Characters CPreExecut ai_mana C N C STaskItem:					L. Stan
anonymous-namespace'::UpdateRunners Execut					AN AND
anonymous-namespace'::UpdateRunners Execut	_				PIL Frame
SynchronizedCo CDaily TickCommand:Inter Sync CDaily TickCommand: Obaily TickCom CGameStateHildenDataWr CEvel gamestatecommands: Daily TickCom CAlManager::UpdatePosttxi E STaskItem::Vail STaskItem::Vail Daily TickCom CAlManager::UpdatePosttxi E STaskItem::Vail STaskItem::Vail Ini Cha M M CA AlUpdateTasksinPara E STaskItem::Vail STaskItem::Vail STaskItem:: STaskItem:: STaskItem:: Gamestatecommands Characters PreExecut ai_man gamestatecommands Characters PreExecut Gamestatecommands Characters PreExecut ai_man gamestatecommands Characters PreExecut Gamestatecommands Characters PreExecut ai_man gamestate gamestate Gamestate Gamestate Gamestate NAI game Game Game Game Game Game NAI game Game Game Game Game Game Game Ai_man game game Game Game Game Game Game Game Game <td< td=""><td></td><td></td><td></td><td>espace'::UpdateRunners</td><td></td></td<>				espace'::UpdateRunners	
CDaily TickCom CGameStateHiddenDataWr CEve gamestatecommands. Daily Tick CAlManager:UpdatePostEx E STaskItem::Wai Ini Cha M M CA AlUpdateTaskIPA E STaskItem::Wai Ini Cha M M CA AlUpdateTaskIPA E STaskItem::Wai STaskItem:: STaskItem:: gamestatecommands. CAIWarCoordinato characters PreExecut gamestatecommands. CN characters PreExecut gamestatecommands. Imi Cha III (Initiation in the stress of the stress) modifiers P gamestatecommands. Imi Cha III (Initiation in the stress) gamestate gamestate Imi Cha IIII (Initiation in the stress) gamestate gamestate Imi Cha IIII (Initiation in the stress) gamestate gamestate Imi IIII (Initiation in the stress) gamestate gamestate Imi IIIII (Initiation in the stress) gamestate gamestate Imi IIIII (Initiation in the stress) gamestate gamestate Imi IIIII (Initiation in the stress) gamestate gamestate Imi IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	id	ExecuteGameCommand			
DailyTick CAIManager::UpdatePostEx E STaskItem::Wai Ini Cha M M CA AlUpdateTasksInPara ai_manager.cpp:1001 gamestatecommands CA STaskItem:: gamestatecommands CAWarCoordinato characters PreExecut ai_man gamestatecommands. C N gamestatecommands. ai_man gamestatecommands. ai_man gamestatecommands. MAI game ai_man game ai_man game ai_man game ai_ma game ai_ma game ai_ma game ai_ma game	Synchroniz	CDailyTickCommand:	ter Syn	CDailyTickCommand::Int	SynchronizedCo
Ini Cha M M CA AlUpdateTasksInPara ai_manager.cpp:1001 STaskItem:: ai_manager.cpp:1001 gamestatecommands CAlWarCoordinato characters PreExecut ai_man Gamestatecommands. C N characters 2PreExecut ai_man gamestatecommands. C N characters 2PreExecut characters	CDailyTick	gamestatecommands.		CGameStateHiddenDataV	CDailyTickCom
ai_manager.cpp:1001 STaskItem:: ai_manager.cpp:1001 GAIWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecut ai_man gamestatec N gamestatec N gamestatec ai_man gamestatec N gamestatec N gamestatec N gamestatec N gamestatec N gamestatec ai_man gamestatec N gamestatec N gamestatec ai_man gamestatec N gamestatec ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai_man ai ai_m	DailyTick	STaskItem::Wai			
STaskitem:: ai_manager.cpp:1001 CAIWarCoordinato CAIWARCOORD	I Ch MM				Ini Cha M M
ai_manager.cpp:1001 gamestatecommands CAWarCoordinato characters PreExecut ai_man gamestatecommands C N characters PreExecut ai_man gamestatecommands characters PreExecut ai_man gamestatec N modifiers P ai_man gamestate N ai PreExecut ai_man gamestate NAI: onactions mapgraph game NAI armi ai_ma game NAI armi ai_ma game NAI armi ai_ma game NAI armi ai_ma game ai_ma game					
CAlWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecut ai_man gamestatec N modifiers P ai_man gamestate ai_man gamestate NAI: onactions mapgraph game CMapGrap relat ai_ma game NAI armii ai_ma game NAI armii				STaskItem::	
CAlWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecut ai_man gamestatec N modifiers P ai_man gamestate ai_man gamestate NAI: onactions mapgraph game CMapGrap relat ai_ma game NAI armii ai_ma game NAI armii					
CAlWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecut ai_man gamestatec N modifiers P ai_man gamestate ai_man gamestate NAI: onactions mapgraph game CMapGrap relat ai_ma game NAI armii ai_ma game NAI armii					
CAlWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecu ai_man gamestatec N modifiers P ai_ma gamestate ai_ma gamestate NAI: onactions mapgraph game CMapGrap relat ciai_ma game ai_ma game NAI armii ai_ma game NAI armii					
CAlWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecut ai_man gamestatec N modifiers P ai_man gamestate ai_man gamestate NAI: onactions mapgraph game CMapGrap relat ai_ma game NAI armii ai_ma game NAI armii					
CAlWarCoordinato characters PreExecut ai_man gamestatecommands. C N characters2 PreExecut ai_man gamestatec N modifiers P ai_man gamestate ai_man gamestate NAI: onactions mapgraph game CMapGrap relat ai_ma game NAI armii ai_ma game NAI armii					
ai_man gamestatecommands. C N characters2 PreExecu ai_man gamestatec N gamestatec ai_man gamestate ai_man gamestat ai_man gamestat ai_man game CMagGraph game CMagGraph game ai_man game NAI armi ai_ma game NAI armi					
C N characters2 PreExecu ai_man gamestatec ai_man gamestate ai_man gamestate ai_man gamestat ai_man gamestat CMapGrap relat ai_ma game NAI armi ai_ma game NAI armi ai_ma game			to		
ai_man gamestatec N modifiers P ai_mai gamestate ai Pretixee ai Pretixee ai MAI: cMapGrap relat ai_mai game NAI armi ai_mai game NAI armi ai_mai game vassal ai_man gamesta					S
N modifiers P ai_maa gamestate N ai PreExec ai_maa gamestat NAI:: onactions mapgraph game CMapGrap relat NAI armi ai_maa game NAI armi ai_maa game NAI armi ai_maa game yassal yassal ai_maa gamesta		characters2 PreExecu			
ai_ma ai_ma ai_maa gamestat ai_maa gamestat cMapGrap cCMapGrap ai_ma ai_ma ai_ma game vassal ai_man gamesta game cMapGrap cMapGra					
N ai PreExec ai_mana gamestat NAI:: onactions mapgraphi game CMapGrap relat ai_ game NAI armi ai_ma game vassal ai_man ai_man gamesta		and the second se		and the second	
ai_mana gamestat NAI: onactions mapgraph game CMapGrap relat ai_ma game NAI armi ai_ma game vassal ai_man gamesta					
NAI:: onactions mapgraph game CMapGrap relat ai_ game NAI armi ai_ma game vassal ai_man gamesta				A REAL PROPERTY OF A READ REAL PROPERTY OF A REAL P	
mapgraph CMapGrap ai NAI ai ai ai ai ai ai ai ai ai ai ai ai ai					
CMapGrap relat aigame NAI armi ai_ma game vassal ai_man gamesta		Balander De De Cancilla			
ai_ma game NAI armi ai_ma game Vassal ai_man gamesta					
NAI armi ai_mai game vassai ai_man gamesta					
ai_ma vassal ai_man gamesta					
ai_man gamesta					
ai_man gamesta				ai_ma	
		guests P			
ai_man game		game		al_man	

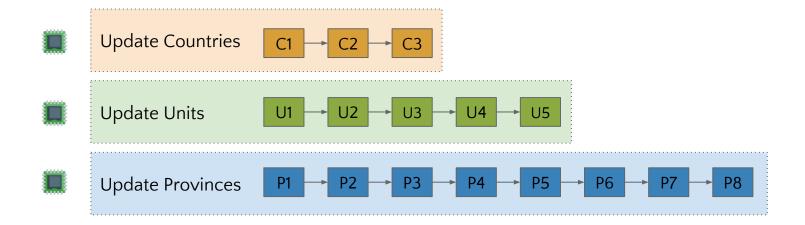
CK3 on 16 cores



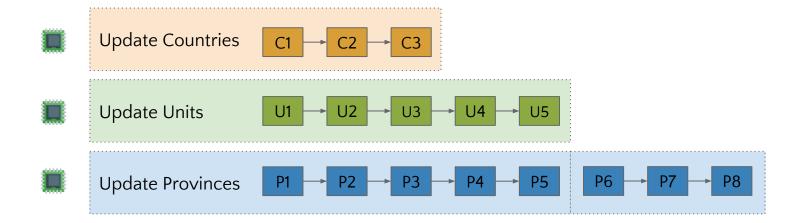
- Make entities in a system only rely on others' public data
- Have no order of execution requirement
- Each entity (or batch of entities) becomes a task you can schedule



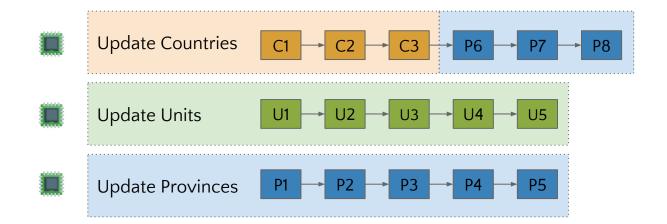
Present PDS Games Loop



Potential Future PDS Games Loop



Potential Future PDS Games Loop



Potential Future PDS Games Loop



- Define sub-system update requirements
 - Read-only or read-write gamestate access
 - Entity in-order requirements
- Allow the update scheduler to break down sub-system updates into smaller chunks when the right requirements are filled



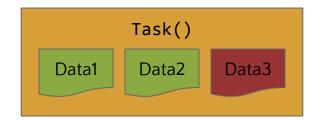
- Double-down on what CK3 started
- Make the model more explicit in the API
- Offer a way to break down systems into smaller chunks automatically if possible



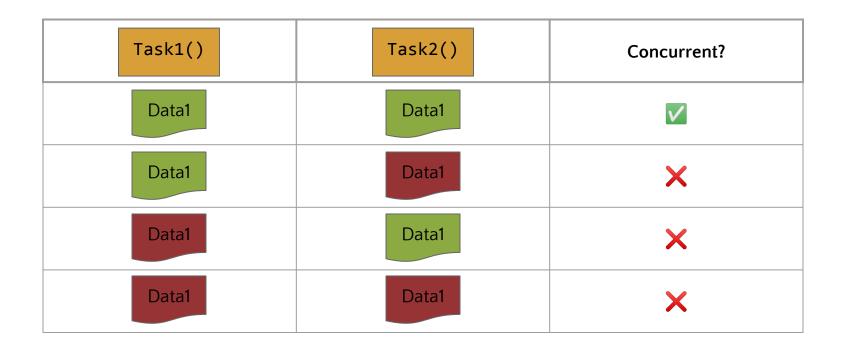
Let's take a step back



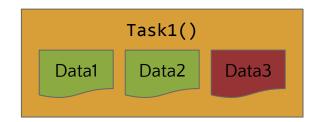
What's in a task?

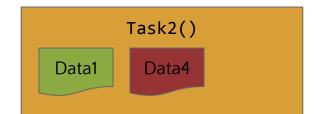


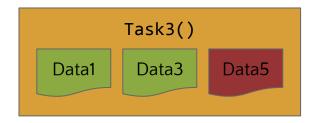
What's in a task?



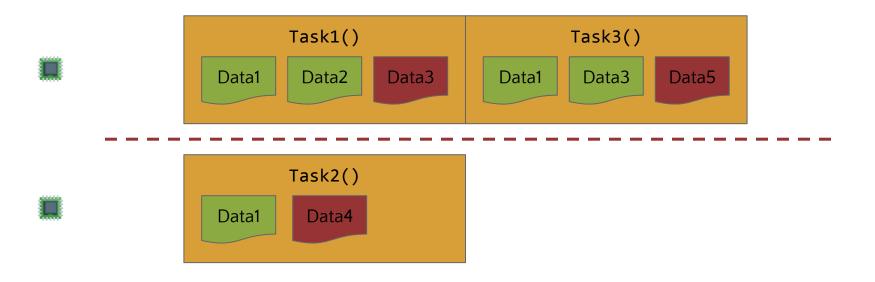
Concurrent Data Access

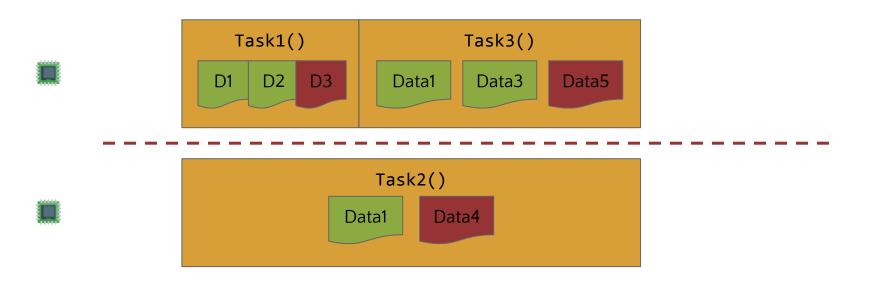






What's in a task?

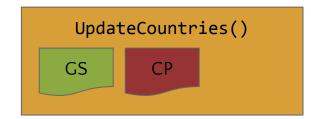


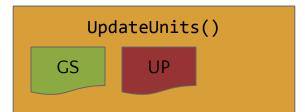


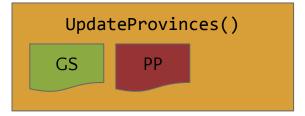
Scheduling



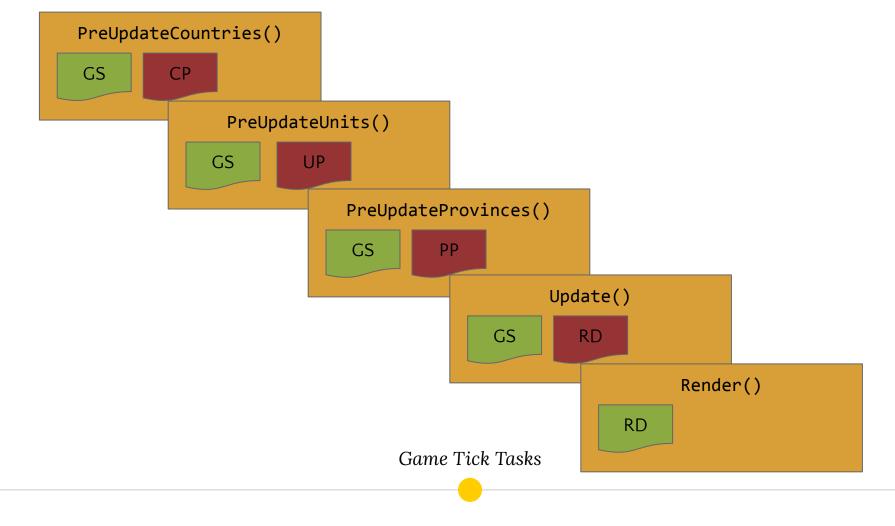
Game Tick PreUpdate Task

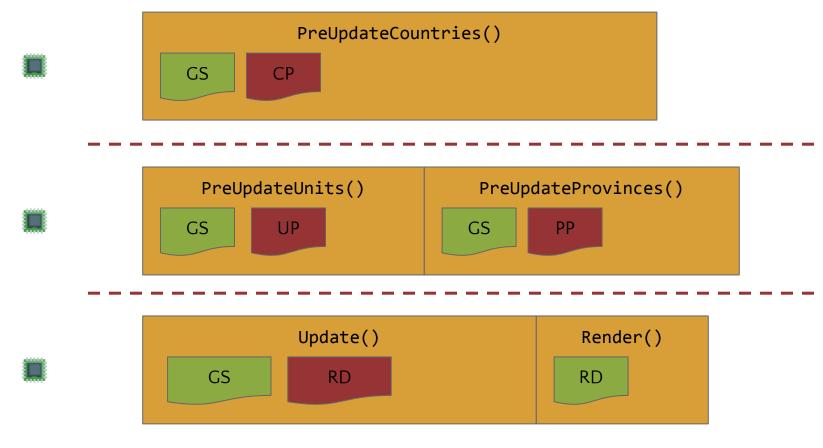






Game Tick PreUpdate Tasks





Game Tick Tasks Scheduling



- Game Tick is a series of tasks
- Tasks have inputs (read) and outputs (writes)
- 2 tasks can be scheduled at the same time if they reads and writes don't conflict



- CK3 Pre Update Tasks read all gamestate and write to a private stash
- CK3 Update Tasks read from one private stash and write to all gamestate
- Update Task reads all gamestate



- Better define update tasks read and writes (which subsections of the gamestate)
- Make scheduler consider both task logical dependencies and r/w data access
- Fit update/render task in that model?



- CK3 model is really simple to reason about and teach to new programmers
- Adding task dependencies might run against parallelism if there are too many
- Explicit data usage declaration make design iteration more expensive



The threads of past, present and future



- Using modern CPUs efficiently require good core utilization
- Adding parallel_for to existing code only gets you so far
- Adopting a model that enforces thread-friendly constraints is key

Furthermore



Furthermore, I think your build should be destroyed





Any questions ?

You can reach me at

- A mro@puchiko.net
- ♥ @MatRopert
- @mropert
- https://mropert.github.io